An Octave-based Multi-Resolution CQT Architecture for Diffusion-based Audio Generation

Maurício do V. M. da Costa
MTDML, IMM
University of Osnabrück
Osnabrück, Germany
madovalemade@uni-osnabrueck.de

Eloi Moliner
Acoustics Lab, DICE
Aalto University
Espoo, Finland
eloi.moliner@aalto.fi

***Abstract*—This paper introduces MR-CQTdiff, a novel neural-network architecture for diffusion-based audio generation that leverages a multi-resolution Constant-Q Transform (CQT). The proposed architecture employs an efficient, invertible CQT framework that adjusts the time-frequency resolution on an octave-by-octave basis. This design addresses the issue of low temporal resolution at lower frequencies, enabling more flexible and expressive audio generation. We conduct an evaluation using the Fréchet Audio Distance (FAD) metric across various architectures and two datasets. Experimental results demonstrate that MR-CQTdiff achieves state-of-the-art audio quality, outperforming competing architectures.**

***Index Terms*—diffusion models, generative models, Constant-Q Transform**

I. Introduction

Time-Frequency Representations (TFRs) are fundamental tools in a wide range of audio processing applications, often employed in state-of-the-art deep learning systems for audio analysis [1], [2], synthesis [3]–[6] , and enhancement [7], [8]. TFRs can be computed with varying resolutions. For example, generating a spectrogram using the short-time Fourier transform (STFT) involves selecting an analysis window whose length determines the trade-off between time and frequency resolution. The longer the analysis window, the higher the frequency resolution—at the cost of lower time resolution—and vice versa. Therefore, selecting an appropriate resolution for the task and signal characteristics is essential.

Spectrograms based on the STFT use linear time and frequency grids. For audio with strong harmonic content, such as music, harmonic components of musical notes are linearly spaced in frequency, and their spacing varies across pitches. An alternative TFR, the Constant-Q Transform (CQT), adopts a logarithmic frequency scale, yielding pitch-invariant harmonic spacing. This makes the CQT particularly suitable for Convolutional Neural Networks (CNNs), as it aligns well with the translation-equivariant nature of convolutional kernels when processing harmonic signals. However, the CQT exhibits low time resolution at lower frequencies, which limits its ability to capture transient events. This leads to excessive temporal smearing of percussive sounds and low-pitch notes, resulting in degraded feature quality for learning tasks.

Both authors contributed equally to this work. It was funded by Volkswagen Foundation (Volkswagen Stiftung) Germany, under Grant no. 96 881.

This paper focuses on diffusion-based generative models [9]–[11], a class of deep generative methods that have demonstrated strong performance across various tasks, including speech synthesis [12], [13], audio restoration [14], and automatic music generation [15], [16]. Although diffusion models are architecture-agnostic, both architectural design and the choice of data representation significantly affect performance. Representations that emphasize task-relevant structure introduce inductive biases that facilitate training and improve sample quality [17].

Early approaches to diffusion-based audio generation operated directly on the waveform domain, targeting applications such as speech [18], sound effects [19], and drum synthesis [20]. However, modeling raw waveforms remains challenging due to their high dimensionality and limited structure, making pattern learning inefficient and poorly scalable. As an alternative, several works proposed operating in the mel-spectrogram domain, which provides a lower-dimensional and perceptually meaningful representation of audio [12], [21], [22]. While this reduces the complexity of modeling, it introduces an additional burden: the need for a separately trained neural vocoder to reconstruct audio, which can become a bottleneck in quality and flexibility.

More recently, following their success in image generation [23], latent diffusion models (LDMs) have emerged as a dominant approach for audio generation [15], [16], [24], [25]. These models decompose the generative process into two stages: first, an autoencoder is trained to compress audio into a lower-dimensional latent space; then, a diffusion model is trained to model the distribution of these latent representations. This strategy concentrates domain-specific design efforts on the autoencoder, allowing the diffusion model to adopt general-purpose, highly scalable architectures, such as transformers [26], due to the reduced structure and dimensionality of the latent space. While this approach enables efficient training and the modeling of complex audio distributions, it also presents notable challenges. The quality of the final output is inherently limited by the reconstruction error of the autoencoder, and the separation of training into two independent stages introduces additional complexity [27]. Moreover, operating in a latent domain often complicates conditional generation tasks, such as solving inverse problems, where the relationship between

observations and latent variables is indirect or unknown [28].

An alternative paradigm seeks to overcome the curse of dimensionality by leveraging invertible time–frequency representations. These representations reveal inherent structure in audio signals, exhibit sparsity, and provide a potentially more tractable domain for modeling complex data distributions. One line of work explores the use of the STFT, either by defining the diffusion process directly in the STFT domain [7], [29], or by exploiting the differentiability of the inverse STFT to operate in the waveform domain while processing features in the time–frequency domain [8], [30]. The latter approach retains the simplicity of waveform-based modeling, while benefiting from the inductive biases introduced by time–frequency representations. Building on this idea, Moliner et al. [8], [17] proposed using the CQT for music restoration tasks. Their method relies on an invertible and differentiable implementation of the CQT, combined with a U-Net architecture that exploits temporal redundancies in the transform for efficient and scalable processing [17].

In this work, we introduce the multi-resolution CQTdiff (MR-CQTdiff), a modification to the CQTdiff+ model proposed in [17]. Our architecture leverages a CQT filter bank, i.e. multiple parallel CQTs with different resolutions covering complementary frequency ranges, aiming to balance time-frequency resolution by progressively decreasing the ratio between time and frequency resolutions on an octave basis across the audible spectrum. In doing so, it avoids excessively low time resolution at low frequencies while maintaining a relatively high frequency resolution at high frequencies, which a single CQT cannot achieve. While still supporting a U-Net-based architecture, the CQT filter bank better captures transient and low-frequency content, enabling higher-quality generation of musical material overall.

The remainder of the paper is organized as follows: Section II introduces the diffusion model framework and training strategy; Section III details the MR-CQTdiff architecture; Section IV presents the experiments conducted and results achieved; and Section V concludes with a discussion of findings and future research directions.

II. Diffusion Models for Audio Generation

Diffusion models are a class of generative models that learn to synthesize data by reversing a gradual noising process. In the forward process, data samples $\boldsymbol{x}_0$ are progressively corrupted by adding Gaussian noise, resulting in a sequence of increasingly noisy versions of the data $\boldsymbol{x}_\tau$. This transformation defines a diffusion process parameterized by a continuous time variable τ [10].[1] During training, a neural network is optimized to approximate the score function $s_\theta(\boldsymbol{x}, \tau) \approx \nabla_{\boldsymbol{x}} \log p_\tau(\boldsymbol{x}_\tau)$, which defines a vector field that indicates the direction towards regions of higher probability at time τ. Once trained, the score model can be used to guide a reverse diffusion process, which transforms samples drawn from of Gaussian distribution $\boldsymbol{x}_T \sim \mathcal{N}(\mathbf{0}, \sigma_{\max}^2 \boldsymbol{I})$ into samples from the training data distribution $\boldsymbol{x}_0 \sim p_{\text{data}}$.

The score model $s_\theta(\mathbf{x}_\tau, \tau)$ is typically trained using the denoising score matching objective [31]:

$$\mathbb{E}_{\boldsymbol{x}_0, \boldsymbol{\varepsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})} \left[\lambda(\tau) \left\| s_{\boldsymbol{\theta}}(\boldsymbol{x}_0 + \sigma(\tau)\boldsymbol{\varepsilon}, \tau) - \frac{\boldsymbol{x}_0 - \boldsymbol{x}_\tau}{\sigma^2(\tau)} \right\|_2^2 \right], \tag{1}$$

where $\lambda(\tau)$ is a time-dependent weighting parameter, and $\sigma(\tau)$ defines the noise level at time τ.

Following [8], [17], we adopt several design choices proposed by Karras et al. [11], such as defining $\sigma(\tau) = \tau$. To ensure stability and scale consistency throughout training, Karras et al. [11] also propose the following parameterization of the score model:

$$s_{\boldsymbol{\theta}}(\boldsymbol{x}_\tau, \tau) = \frac{(c_{\text{skip}}(\tau) - 1)x_\tau + c_{\text{out}}(\tau) F_{\boldsymbol{\theta}}(c_{\text{in}}(\tau)\boldsymbol{x}_\tau, \tau)}{\sigma^2(\tau)}, \tag{2}$$

where $F_{\boldsymbol{\theta}}$ is the core neural network, and the weighting parameters c_{skip}, c_{out} and c_{in} are chosen to maintain close-to-unit variance in the input and output of $F_{\boldsymbol{\theta}}$, , which is known to improve neural network training stability.

The contributions of this paper focus on the architectural design of the core network $F_{\boldsymbol{\theta}}$. While our experiments adopt the parameterization introduced by Karras et al. [11], we believe that architectural choices are largely orthogonal to the specific diffusion parameterization. Therefore, our findings should generalize to alternative formulations, such as DDPM [9] or Flow Matching [32].

A. Inference

Also following [11], we employ the following ordinary differential equation (ODE) to traverse the generative (reverse) process:

$$d\boldsymbol{x} = -\tau \, s_\theta(\boldsymbol{x}_\tau, \tau) \, d\tau, \tag{3}$$

where $d\tau$ is an infinitesimal negative time step.

At inference time, the continuous time variable τ is discretized into a sequence of T steps using a noise schedule defined as

$$\tau_i = \left(\sigma_{\max}^{1/\rho} + \frac{i}{T-1} \left(\sigma_{\min}^{1/\rho} - \sigma_{\max}^{1/\rho} \right) \right)^\rho, \quad i = 0, \ldots, T-1, \tag{4}$$

where $\sigma_{\max}$ and $\sigma_{\min}$ denote the maximum and minimum noise levels, respectively, and ρ controls the nonlinearity of the spacing between time steps. Unless otherwise stated, we use $\sigma_{\max} = 8$, $\sigma_{\min} = 10^{-5}$, $\rho = 10$, and $T = 51$ steps.

To numerically integrate the reverse ODE, we use the second-order Heun's method (also known as improved Euler), as proposed in [11]. This solver provides a good trade-off between sample quality and computational efficiency.

III. Multi-Resolution CQT Architecture

Our architecture builds upon the CQT-Diff+ algorithm, proposed in [17], which operates in the time-frequency domain using a CQT. The transform implementation is a differentiable version of the CQT introduced by Velasco et al. [33] and

[1] Note that the time variable τ is used to describe the diffusion process, which is independent of the time domain t of the audio signal.

Holighaus et al. [34], which is computationally efficient by leveraging FFT-based band-pass filters. This implementation is also invertible, enabling perfect reconstruction up to numerical error.

Formally, the system can be described as

$$F_{\boldsymbol{\theta}} = \mathrm{IC}Q\mathrm{T} \circ U_{\boldsymbol{\theta}} \circ \mathrm{C}Q\mathrm{T}, \tag{5}$$

where $\circ$ denotes the function composition operation, U_θ represents the neural network with trainable weights θ, and CQT and ICQT are the constant-Q-transform operator and its inverse, respectively. Note that the ICQT must be *differentiable* to allow backpropagation, but there is no need to make it *trainable*.

The center frequencies f_k of the K filters g_k are logarithmically distributed within the frequency range of interest and can be calculated by

$$f_k = f_{\min} 2^{\frac{k-1}{b}}, \text{ for } k = 1, 2, 3, \ldots, K, \tag{6}$$

where b denotes the number of bins per octave and $f_{\min}$ is the lowest center frequency. The maximum frequency can be arbitrarily chosen and is typically set close to the Nyquist limit $f_k = f_\mathrm{s}/2$. In our implementation, different CQTs cover complementary frequency ranges, thus requiring different minimum and maximum frequencies.

Given the strong presence of harmonic content in musical signals, the constant-Q transform (CQT) provides a key advantage: its logarithmic frequency scale promotes pitch-equivariant symmetry. This property makes the CQT particularly well-suited for convolutional architectures, outperforming general STFT-based spectrograms in harmonic contexts. Due to the transform's design (where filters become progressively narrower at lower frequencies), the corresponding impulse responses in the time domain must be proportionally longer, as dictated by the uncertainty principle.[2] An additional consequence is that a CQT with a regular time-frequency grid inherently exhibits redundancy in the time domain, which increases toward lower frequencies.

An effective strategy to reduce excessive redundancy, used in CQT-Diff+, is to adopt octave-based regular grids, where the number of time frames is halved with each lower octave. This approach integrates naturally with the U-Net architecture, which represents data at multiple resolutions through progressive downsampling. In CQT-Diff+, the first U-Net level processes the highest octave (with the finest time resolution), and each subsequent level receives a concatenation of the next lower octave with a downsampled version of the higher octaves. This ensures that all inputs are aligned in resolution. Further implementation details are provided in [17].

Despite its efficiency, this system inherits a fundamental limitation of the CQT: poor time resolution at low frequencies. This manifests as increasing temporal smearing toward the lower end of the spectrum, which compromises the representation of transient information, such as fast pitch fluctuations. One way to improve time resolution at low frequencies is to reduce the number of bins per octave (b), resulting in shorter filter impulse responses and broader frequency bandwidths. However, this comes at the cost of frequency resolution at higher frequencies, potentially impairing the network's ability to distinguish fine harmonic structures in that range.

[2]The uncertainty principle, or Heisenberg's principle, states that the product of a signal's time support and the frequency support of its transform is lower bounded.

A. Architecture Design

The MR-CQTdiff architecture follows a U-Net structure, as illustrated in Figure 1, with concatenative skip connections linking encoder and decoder layers at corresponding resolutions. Anti-aliasing filters are applied during both downsampling and upsampling stages. Each resolution level (including the bottleneck) contains a residual block, referred to as "Res. Block", which serves as the core computational unit. In the encoder, the input is divided into octave-specific segments and processed independently using "In. Blocks". These features are concatenated along the frequency axis with the corresponding U-Net latents and augmented with residual connections from resized input features to maintain information flow.

The decoder mirrors this dual-path structure: a main path containing "Res. Blocks" with progressive upsampling, and an auxiliary "outer" path that enhances gradient flow. At each decoder level, lower-octave features are discarded from the main path and passed through "Out. Blocks" to the outer path. These features are eventually routed to the ICQT module. This two-path setup ensures both efficient feature reuse and improved training stability.

Our solution addresses the aforementioned time-resolution problem by computing multiple CQTs that cover complementary frequency ranges. The proposed architecture follows the overall structure of CQT-Diff+, replacing certain time-domain downsampling operations with frequency-domain ones when transitioning between resolutions. The main diagram in Figure 1 illustrates a simplified example of the architecture for $N = 3$ octaves. In this case, two CQTs are used: CQT-1 is computed with b bins per octave, while CQT-2 uses $b/2$ bins per octave, trading frequency resolution for improved time resolution at the lowest octave (Oct. N–3). Since Oct. N (the highest) and Oct. N–1 share the same number of frequency bins b, a $2\times$ downsampling in time is applied to match their dimensions for concatenation. At the transition to the next U-Net level—the crossover point between the different CQTs—the time dimensions of the features are already aligned, so frequency downsampling is applied at that point.

All building blocks in the architecture are conditioned on a noise-level embedding (σ-emb), constructed using Random Fourier Features [35] followed by a three-layer MLP. Conditioning is applied via feature-wise linear modulation without shifts. The "In. Block" expands the input (real and imaginary parts) from two channels to the desired latent size using a 1×1 convolution, followed by Group Normalization (shift-free), Gaussian-error-linear-unit ("GELU") activation, and a linear layer. The "Out. Block" mirrors this structure, applying 1×1 convolution at the end to reduce the latent size back to two channels. Each "Res. Block" contains shift-free Group

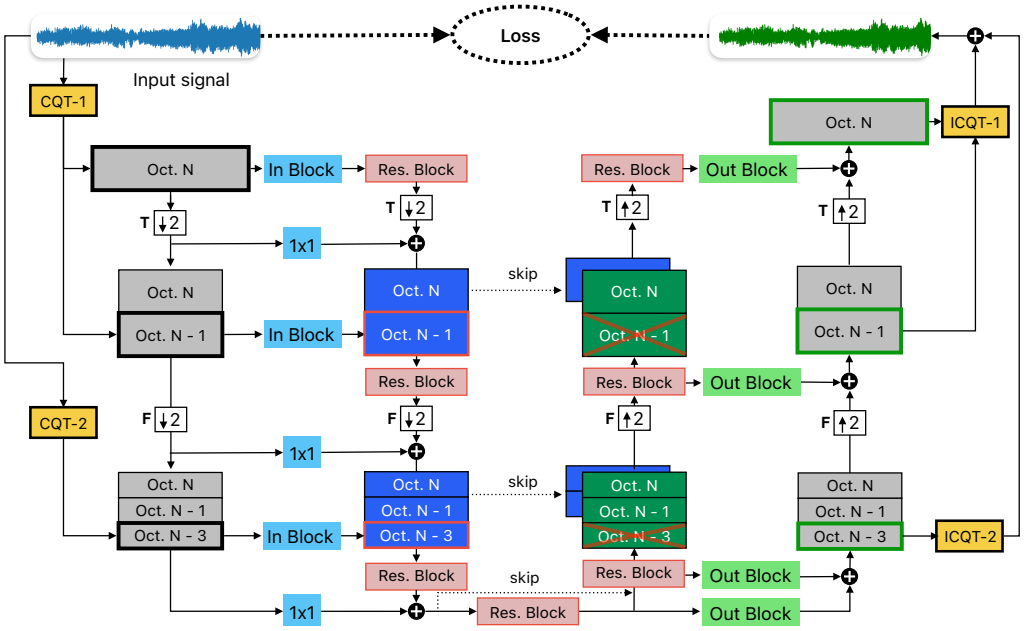

Fig. 1. Main structure of the MR-C*Q*Tdiff architecture, illustrating a simplified example with three hypothetical octaves processed by two C*Q*Ts: CQT-1 with resolution b bins per octave, and CQT-2 with $b/2$ bins per octave.

Normalization, GELU activation, and convolutions in time and frequency, with exponentially increasing dilation along the frequency axis to achieve pitch-equivariant receptive fields.

B. Hyperparameter Specification

The system operates with audio signals sampled at 44.1 kHz, which are transformed by three C*Q*Ts with resolutions $b = \{8, 16, 32\}$ bins/oct., respectively covering the frequency ranges $43.1 - 344.5$ Hz, $344.5 - 5512.5$ Hz, and $5512.5 - 22050$ Hz, which together span $N_{\text{oct}} = 9$ octaves. As a means of comparison, in the original study [17], the CQTdiff+ algorithm used a C*Q*T with $b = 64$, which yields $8\times$ lower time resolution at the first octaves. Table I details the frequency range covered by each octave, its number of bins per octave, the level at the U-Net at which that octave is fed, and the resampling type (time- or frequency-wise).

The U-Net depth corresponds to the number of octaves, with feature sizes increasing from 32 in the shallowest layers to 256 at the bottleneck. Each "Res. Block" contains between two and five stacked dilated convolutions, with fewer dilations in shallower layers due to the smaller number of frequency bins and reduced need for a large receptive field. Importantly, these architectural modifications only marginally affect the overall model size, which remains around 40 million parameters.[3]

TABLE I
HYPERPARAMETERS OF THE MR-C*Q*T STRUCTURE.

Octave	Frequency Range (Hz)	b	U-Net Level	Resampling
9	11025 – 22050	32	1	Time
8	5512.5 – 11025	32	2	Freq.
7	2756.3 – 5512.5	16	3	Time
6	1378.1 – 2756.3	16	4	Time
5	689.1 – 1378.1	16	5	Time
4	344.5 – 689.1	16	6	Freq.
3	172.3 – 344.5	8	7	Time
2	86.1 – 172.3	8	8	Time
1	43.1 – 86.1	8	9	–

IV. Evaluation

A. Training Datasets

We trained all models on two datasets: FMA-Large [36], a diverse collection of 106,574 30-second music tracks across various genres, and OpenSinger [37], a dataset of professionally recorded solo vocal performances. These datasets present distinct challenges: FMA-Large covers a broad range of musical styles and production qualities, while OpenSinger focuses on clean vocal recordings across different singers and pitches, making them complementary benchmarks for evaluating generative performance.[4]

B. Baselines

We trained four models in the waveform domain using 6-second audio segments and the same diffusion parameterization, formally described in Section II. The only difference between models lies in the architectural choices:

[3]Our implementation is available online on https://github.com/eloimoliner/MR-CQTdiff

[4]We opted not to run experiments with the MAESTRO dataset due to the presence of poor-quality, noisy recordings, especially in early recordings. Preliminary tests resulted in noisy samples across all models, significantly influencing in the quality assessment of the generated samples.

- *UNet-1D*: A 1-dimensional U-Net composed of temporal convolutions, similar to architectures used in waveform-domain diffusion [18]–[20].
- *NCSN++*: A 2-dimensional U-Net operating on STFT representations, originally introduced in [10] and later adapted for speech enhancement in [7]. This architecture has also been applied to speech and singing voice modeling [30], following the same differentiable composed design. It uses 2D convolutions over time and frequency axes.
- *CQTdiff+*: The baseline model introduced in [8], which uses a differentiable and invertible C*Q*T representation combined with a U-Net architecture. Our proposed model is built upon this baseline.
- *MR-CQTdiff* (ours): The proposed model, which extends CQTdiff+ by introducing a multi-resolution C*Q*T filter bank.

All these architectures are configured to share a similar parameter count of around 40 million parameters.

To provide additional comparison, we also evaluated a latent diffusion model (*LDM*) using the publicly available autoencoder from Stable Audio Open [16]. This model employs a Transformer-based architecture similar to that used in the original work, and has approximately 67 million parameters. To ensure a fairer comparison, we adapted our diffusion parameterization to the latent domain, applying the same training framework as used for the waveform-domain models. Necessary adjustments, such as modifying the noise schedule, training loss weighting, and preconditioning, were made to account for the properties of the latent space.

All models were trained for 500,000 iterations using the Adam optimizer with a learning rate of 1×10^{-4} and a batch size of 4, on a single NVIDIA A100 GPU. Model checkpoints were saved and evaluated every 100,000 iterations. During training, we maintained an exponential moving average (EMA) of the model weights with a decay rate of 0.9999, which was used for inference. Total training times (500,000 iterations) varied depending on the model: approximately 46 hours for *UNet-1D*, 80 hours for the *NCSN++* model, 125 hours for *CQTDiff+*, 120 hours for *MRCQT-Diff*, and 25 hours for *LDM*.

C. Experiments: Unconditional Generation

We evaluated the performance of the five models on unconditional audio generation using two datasets: *OpenSinger* and *FMA*. Generation was performed using the same sampler and sampling parameters described in Section II-A, with the exception of the *LDM*, for which we used $\sigma_{\max} = 100$ and $\sigma_{\min} = 10^{-4}$. For each model, we unconditionally generated 512 audio samples of 6 seconds every 100,000 training iterations.

To assess generation quality, we used the *Fréchet Audio Distance (FAD)*, computed from CLAP embeddings [38] using the official FADtk implementation [39]. FAD quantifies the distance between the distributions of embeddings extracted from real and generated audio. In order to analyze the contribution of each test sample to the FAD, and thus potentially detect outliers, we compute individual FAD scores for each song, as proposed in [39]. CLAP embeddings were selected due to their strong correlation with perceptual audio and musical quality [39].

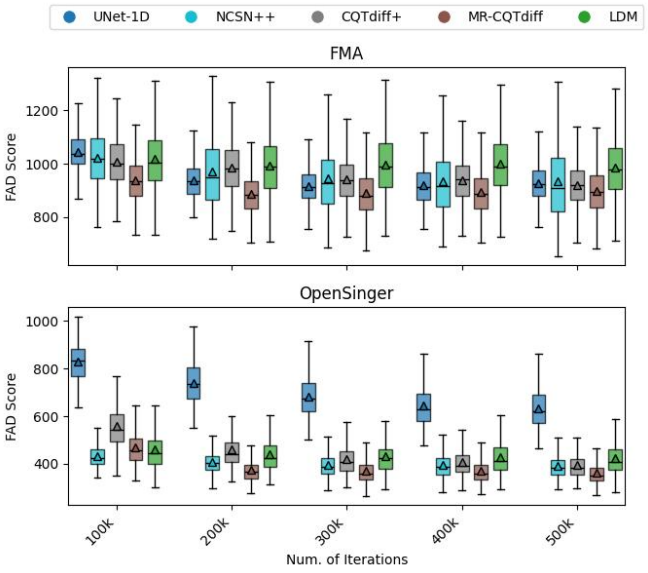

Fig. 2. Boxplot of per-example FAD scores computed with CLAP embeddings, across different models and training iterations, for the *FMA* and the *OpenSinger* datasets. Lower values indicate closer alignment between generated and real audio distributions.

As the reference set for computing FAD, we used the held-out test partitions of *FMA* and *OpenSinger*. The *FMA* test set comprises approximately 17 hours of music, while the *OpenSinger* test set contains around 2.5 hours of audio. Since the reference sets are entirely disjoint from the training data, this evaluation setup allows us to measure the models' generalization ability.

D. Results

Figure 2 illustrates the distributions of FAD scores for all models at various training iterations, separately for the *FMA* (top) and *OpenSinger* (bottom) datasets. For the *OpenSinger* dataset, the proposed architecture clearly outperforms the others, achieving near-stable performance by around 200,000 iterations. This suggests that the model effectively captures transient details, which are especially important for singing voice, where rapid pitch variations and prominent non-harmonic sounds (e.g., consonants) occur frequently.

A similar trend is observed in the results obtained using the *FMA* dataset, although the score distributions are wider, reflecting the greater diversity and complexity of general music data. Despite this challenge, the proposed MR-CQTdiff consistently attains the lowest median FAD scores and stabilizes its performance by 200,000 iterations, whereas other models require more iterations to converge. In this setting, the latent diffusion model (LDM) performs the worst, possibly due to artifacts introduced by the autoencoder reconstruction.

Qualitative audio examples are available on the companion website[5]. Although samples generated from the *OpenSinger* model, especially MR-CQTdiff, exhibit high quality, examples from the *FMA* dataset often sound less pleasing across all models. This outcome is expected given the higher variability and complexity of the *FMA* dataset and the unconditional generation task. We hypothesize that more perceptually satisfying results could be achieved through guided or conditional generation approaches.

V. Conclusions and Future Work

In this work, we introduced the MR-CQTdiff: a novel architecture for diffusion-based audio generation that leverages a multi-resolution constant-Q transform. The proposed architecture aims to overcome the time-frequency resolution trade-offs inherent to standard CQT-based representations.

We evaluated the proposed architecture using the Fréchet Audio Distance (FAD) computed with CLAP embeddings on the *FMA* and the *OpenSinger* datasets. The results obtained show that our model consistently achieves competitive or superior FAD scores compared to four strong baselines: UNet-1D, NCSN++, CQTdiff+, and LDM. In particular, MR-CQTdiff outperforms all other models across most training stages, especially on the OpenSinger dataset, demonstrating improved handling of harmonically rich and transient vocal content.

It is important to note that our experiments were conducted on relatively small-scale models. While latent diffusion models (LDMs) offer clear advantages in scalability and training efficiency, they still underperform MR-CQTdiff at comparable model sizes and training iterations. This highlights the potential of time-frequency-based approaches in high-fidelity audio generation, despite imposing scalability challenges. Future work could explore making MR-CQTdiff more efficient and scaling it to larger architectures and datasets, besides conducting a more comprehensive subjective quality assessment through controlled listening tests.

Finally, we emphasize that the focus of this study was on model performance rather than computational efficiency. Although LDMs remain more efficient in terms of resource usage, our results show that MR-CQTdiff is a compelling alternative when quality is the primary concern.

References

[1] Z. Rafii, "The constant-q harmonic coefficients: A timbre feature designed for music signals [lecture notes]," *IEEE Signal Processing Magazine*, vol. 39, no. 3, pp. 90–96, 2022.

[2] A. Riou, S. Lattner, G. Hadjeres, and G. Peeters, "Pesto: Pitch estimation with self-supervised transposition-equivariant objective," in *International Society for Music Information Retrieval Conference (ISMIR 2023)*, 2023.

[3] A. Marafioti, N. Perraudin, N. Holighaus, and P. Majdak, "Adversarial generation of time-frequency features with application in audio synthesis," in *Proc. ICML*. PMLR, 2019, pp. 4352–4362.

[4] C. Donahue, J. McAuley, and M. Puckette, "Adversarial audio synthesis," in *Proc. ICLR*, 2019.

[5] eloimoliner.github.io/MR-CQTdiff/

[5] J. Engel, K. K. Agrawal, S. Chen, I. Gulrajani, C. Donahue, and A. Roberts, "Gansynth: Adversarial neural audio synthesis," in *Proc. ICLR*, 2019.

[6] S. Huang, Q. Li, C. Anil, X. Bao, S. Oore, and R. B. Grosse, "Timbretron: A wavenet (cyclegan (cqt (audio))) pipeline for musical timbre transfer," in *International Conference on Learning Representations*.

[7] J. Richter, S. Welker, J.-M. Lemercier, B. Lay, and T. Gerkmann, "Speech enhancement and dereverberation with diffusion-based generative models," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 31, pp. 2351–2364, 2023.

[8] E. Moliner, J. Lehtinen, and V. Välimäki, "Solving audio inverse problems with a diffusion model," in *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2023, pp. 1–5.

[9] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," *Advances in neural information processing systems*, vol. 33, pp. 6840–6851, 2020.

[10] Y. Song, J. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon, and B. Poole, "Score-based generative modeling through stochastic differential equations," in *Proc. ICLR*, 2020.

[11] T. Karras, M. Aittala, T. Aila, and S. Laine, "Elucidating the design space of diffusion-based generative models," *Advances in neural information processing systems*, vol. 35, pp. 26 565–26 577, 2022.

[12] V. Popov, I. Vovk, V. Gogoryan, T. Sadekova, and M. Kudinov, "Grad-tts: A diffusion probabilistic model for text-to-speech," in *Proc. ICML*. PMLR, 2021, pp. 8599–8608.

[13] M. Jeong, H. Kim, S. J. Cheon, B. J. Choi, and N. S. Kim, "Diff-tts: A denoising diffusion model for text-to-speech," in *Proc. Interspeech*, 2021.

[14] J.-M. Lemercier, J. Richter, S. Welker, E. Moliner, V. Välimäki, and T. Gerkmann, "Diffusion models for audio restoration: A review [special issue on model-based and data-driven audio signal processing]," *IEEE Signal Processing Magazine*, vol. 41, no. 6, pp. 72–84, 2025.

[15] 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 *Proc. ICML*. PMLR, 2023, pp. 21 450–21 474.

[16] 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.

[17] E. Moliner and V. Välimäki, "Diffusion-based audio inpainting," *Journal of the Audio Engineering Society*, vol. 72, no. 3, pp. 100–113, 2024.

[18] N. Chen, Y. Zhang, H. Zen, R. J. Weiss, M. Norouzi, and W. Chan, "Wavegrad: Estimating gradients for waveform generation," in *International Conference on Learning Representations*, 2021.

[19] S. Pascual, G. Bhattacharya, C. Yeh, J. Pons, and J. Serrà, "Full-band general audio synthesis with score-based diffusion," in *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2023, pp. 1–5.

[20] S. Rouard and G. Hadjeres, "Crash: Raw audio score-based generative modeling for controllable high-resolution drum sound synthesis," in *ISMIR*, 2021.

[21] Y. Koizumi, H. Zen, K. Yatabe, N. Chen, and M. Bacchiani, "Specgrad: Diffusion probabilistic model based neural vocoder with adaptive noise spectral shaping," *arXiv preprint arXiv:2203.16749*, 2022.

[22] S.-L. Wu, C. Donahue, S. Watanabe, and N. J. Bryan, "Music controlnet: Multiple time-varying controls for music generation," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 32, pp. 2692–2703, 2024.

[23] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-resolution image synthesis with latent diffusion models," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 10 684–10 695.

[24] F. Schneider, O. Kamal, Z. Jin, and B. Schölkopf, "Mo\^ usai: Text-to-music generation with long-context latent diffusion," *arXiv preprint arXiv:2301.11757*, 2023.

[25] J. Nistal, M. Pasini, C. Aouameur, M. Grachten, and S. Lattner, "Diff-a-riff: Musical accompaniment co-creation via latent diffusion models," *arXiv preprint arXiv:2406.08384*, 2024.

[26] 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.

[27] S. Dieleman, "Generative modelling in latent space," 2025. [Online]. Available: https://sander.ai/2025/04/15/latents.html

[28] G. Daras, H. Chung, C.-H. Lai, Y. Mitsufuji, J. C. Ye, P. Milanfar, A. G. Dimakis, and M. Delbracio, "A survey on diffusion models for inverse problems," *arXiv preprint arXiv:2410.00083*, 2024.
[29] Z. Kong, K. J. Shih, W. Nie, A. Vahdat, S.-g. Lee, J. F. Santos, A. Jukic, R. Valle, and B. Catanzaro, "A2sb: Audio-to-audio schrodinger bridges," *arXiv preprint arXiv:2501.11311*, 2025.
[30] J.-M. Lemercier, E. Moliner, S. Welker, V. Välimäki, and T. Gerkmann, "Unsupervised blind joint dereverberation and room acoustics estimation with diffusion models," *IEEE/ACM TASLP*, 2025.
[31] P. Vincent, "A connection between score matching and denoising autoencoders," *Neural computation*, vol. 23, no. 7, pp. 1661–1674, 2011.
[32] Y. Lipman, R. T. Chen, H. Ben-Hamu, M. Nickel, and M. Le, "Flow matching for generative modeling," in *The Eleventh International Conference on Learning Representations*.
[33] G. A. Velasco, N. Holighaus, M. Dörfler, and T. Grill, "Constructing an invertible constant-q transform with non-stationary gabor frames."
[34] N. Holighaus, M. Dörfler, G. A. Velasco, and T. Grill, "A framework for invertible, real-time constant-q transforms," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 21, no. 4, pp. 775–785, 2012.
[35] M. Tancik, P. Srinivasan, B. Mildenhall, S. Fridovich-Keil, N. Raghavan, U. Singhal, R. Ramamoorthi, J. Barron, and R. Ng, "Fourier features let networks learn high frequency functions in low dimensional domains," in *Proc. NeurIPS*, vol. 33, 2020, pp. 7537–7547.
[36] M. Defferrard, K. Benzi, P. Vandergheynst, and X. Bresson, "Fma: A dataset for music analysis," *18th International Society for Music Information Retrieval Conference (ISMIR)*, 2017. [Online]. Available: https://github.com/mdeff/fma
[37] R. Huang, F. Chen, Y. Ren, J. Liu, C. Cui, and Z. Zhao, "Multi-singer: Fast multi-singer singing voice vocoder with a large-scale corpus," in *Proceedings of the 29th ACM International Conference on Multimedia (ACM MM)*, 2021. [Online]. Available: https://github.com/Multi-Singer/Multi-Singer.github.io
[38] B. Elizalde, S. Deshmukh, M. Al Ismail, and H. Wang, "Clap learning audio concepts from natural language supervision," in *Proc. ICASSP*. IEEE, 2023, pp. 1–5.
[39] A. Gui, H. Gamper, S. Braun, and D. Emmanouilidou, "Adapting frechet audio distance for generative music evaluation," in *Proc. ICASSP*, 2024. [Online]. Available: https://arxiv.org/abs/2311.01616