VoiceRestore - Flow-Matching Transformers for Universal Audio Quality Restoration

Stanislav Kirdey contact@stankirdey.com

September 6, 2024

Abstract

We present a novel approach to audio quality restoration using flow-matching transformers, capable of addressing a wide range of degradations including reverberation, noise, compression artifacts, and low sampling rates. Our method adapts recent advances in flow matching and transformer architectures to create a unified model for diverse audio restoration tasks. The proposed system leverages conditional flow matching and classifier-free guidance to learn a mapping from degraded to high-quality audio. Experimental results demonstrate state-of-the-art performance across multiple degradation types, outperforming specialized models in terms of both objective metrics and subjective quality. The proposed method shows particular strength in generalizing to unseen degradation combinations, making it a promising solution for real-world audio restoration scenarios.

1 Introduction

Audio quality degradation is a pervasive issue in various applications, from telecommunications to archival audio restoration. Common degradations include reverberation, background noise, compression artifacts (e.g., from lossy encoding), and quality loss due to low sampling rates. Traditional approaches to audio restoration often focus on specific types of degradation, leading to a proliferation of specialized models and techniques.

Recent advancements in deep learning have shown promise in addressing multiple audio degradations simultaneously [1]. However, these methods often struggle with generalization to unseen degradation combinations or require complex architectures and training procedures.

Our work is inspired by the recent Embarrassingly Easy Text-to-Speech (E2TTS) system introduced by Wang et al. [2]. While E2TTS focuses on

speech synthesis, we adapt its core ideas of flow matching and transformer architectures to the task of universal audio restoration. This approach allows us to address a wide range of degradations within a single, unified model. Our approach leverages self-supervised learning, where all degradations are generated in real-time during training. This allows for a more diverse and dynamic training set, enhancing the model's ability to generalize to various degradation types and severities.

By conditioning on the degraded audio input instead of text, our model learns a unified representation capable of addressing a wide range of audio quality issues. The main contributions of this work are:

- A unified flow-matching transformer architecture for multi-degradation audio restoration, adapted from E2TTS
- State-of-the-art results on a comprehensive benchmark covering reverberation, noise, compression artifacts, and low sampling rate restoration tasks
- Demonstration of strong generalization capabilities to unseen degradation combinations
- A flexible and open-source framework that restores speech audio to high-quality 24kHz

2 Related Work

2.1 Speech Enhancement and Dereverberation

Traditional approaches to speech enhancement and dereverberation often rely on signal processing techniques such as spectral subtraction or Wiener filtering [3]. More recently, deep learning and diffusion methods have shown significant improvements, with architectures like SEGAN [4], DCCRN [5] and SGMSE+ [1] achieving state-of-the-art results for noise reduction and dereverberation tasks.

2.2 Flow Matching and Conditional Generation

Flow matching provides an alternative framework for generative modeling, offering advantages in terms of training stability and sampling efficiency. Recent work has extended flow matching to conditional generation tasks [6], opening up new possibilities for audio processing applications.

2.3 E2TTS and Zero-Shot Audio Generation

The recently introduced E2TTS framework [2] demonstrated impressive results in zero-shot text-to-speech synthesis using flow matching and transformer architectures. Our work adapts this approach to audio restoration, replacing the text input with degraded audio and learning a mapping to high-quality audio.

3 Proposed Method

3.1 Problem Formulation

Let $x \in \mathbb{R}^{T \times F}$ be a clean audio spectrogram, where T is the number of time frames and F is the number of frequency bins. We consider a degraded version y generated through various forms of audio degradation. Our goal is to learn a restoration function $f_\theta(y)$ that estimates the clean audio $\hat{x}$ given the degraded input y.

3.2 Flow Matching for Audio Restoration

We adopt a conditional flow matching framework for our audio restoration task. The key idea is to learn a vector field $v_\theta(x_t, t, y)$ that describes the gradual transformation of degraded to clean audio, conditioned on the degraded input y. The flow matching objective is given by:

$$\mathcal{L}(\theta) = \mathbb{E}_{t,x_0,x_1}\left[\|u_t(x_t|x_0, x_1) - v_\theta(x_t, t, y)\|_2^2\right] \tag{1}$$

where u_t is the ground truth vector field derived from the optimal transport path between the degraded and clean audio distributions.

In our implementation, we use a Gaussian distribution for initial sampling of x_0, while x_1 represents the clean audio. We create an interpolation w between x_0 and x_1 based on a randomly sampled time t:

$$w = (1 - (1 - \sigma) \cdot t) \cdot x_0 + t \cdot x_1 \tag{2}$$

where σ is a small constant to prevent singularities. This interpolated w serves as the input to our transformer model.

3.3 Self-Supervised Learning with Real-Time Degradation

A key innovation in our approach is the use of self-supervised learning with real-time degradation generation. Instead of relying on a fixed dataset of

degraded audio samples, we implement a custom algorithm that applies degradations on-the-fly during training. This algorithm combines optimized NumPy acoustic noise generation functions with VST (Virtual Studio Technology) plugins to create a wide range of realistic audio degradations.

The real-time degradation process allows for:

- Dynamic generation of diverse degradation combinations
- Fine-grained control over degradation parameters
- Unlimited training data without the need for extensive pre-processing or storage
- Improved generalization to unseen degradation types and severities

By integrating this approach into our training pipeline, we ensure that the model is exposed to a constantly changing set of degradation scenarios, promoting robust learning and adaptation.

3.4 Transformer Architecture

Our model employs a transformer architecture with several key modifications:

1. **Conditioning**: We use the degraded audio y as conditioning information. It is projected through a linear layer and added to the main input w.

2. **Time Embedding**: The sampled time t is embedded and used to condition the transformer layers, allowing the model to learn time-dependent transformations.

3. **Skip Connections**: We implement U-Net style skip connections to preserve fine-grained information across the network.

4. **Attention Mechanisms**: We use a combination of self-attention and gated attention mechanisms to capture complex dependencies in the audio data.

3.5 Training Procedure

During training, we:

1. Sample a random time t for each element in the batch. 2. Create the interpolated input w between Gaussian noise and the clean audio. 3. Compute the ground truth flow as the difference between clean and noisy inputs. 4. Pass w through the transformer, conditioned on the degraded

audio and time t. 5. Compute the loss between the predicted flow and the ground truth flow.

This procedure allows the model to learn the transformation from noisy to clean audio at various interpolation points.

3.6 Sampling Process

For inference, we use an ODE solver to generate the restored audio. Starting from Gaussian noise, we iteratively apply the learned vector field, conditioned on the degraded input. This process can be described as:

$$\frac{dx}{dt} = v_\theta(x, t, y) \tag{3}$$

We solve this ODE using a numerical solver (e.g., Runge-Kutta methods) to obtain the final restored audio.

3.7 Classifier-Free Guidance

To enhance the quality of generated samples, we implement classifier-free guidance. During sampling, we compute:

$$\hat{v} = v_\theta(x, t, y) + \lambda(v_\theta(x, t, y) - v_\theta(x, t, \emptyset)) \tag{4}$$

where λ is the guidance strength and $v_\theta(x, t, \emptyset)$ represents the unconditional prediction. This technique allows for controlled generation and can improve the fidelity of the restored audio.

3.8 Degradation Generation

Our custom degradation algorithm utilizes a combination of techniques:

- **NumPy-based noise generation**: We implement efficient functions for generating various types of noise (e.g., white noise, pink noise, brown noise) using NumPy.
- **VST plugin integration**: We incorporate VST plugins for more complex degradations such as reverberation, compression, and equalization. These plugins are controlled programmatically to apply randomized degradation parameters.
- **Real-time processing**: Degradations are applied on-the-fly during training, ensuring a unique degradation profile for each training sample.

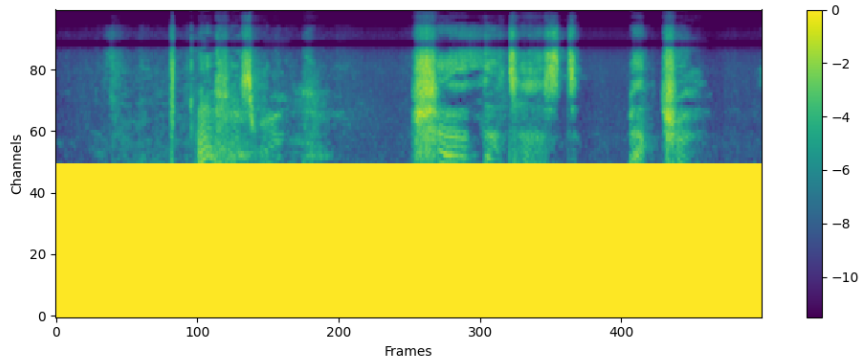

Figure 1: Degraded Input

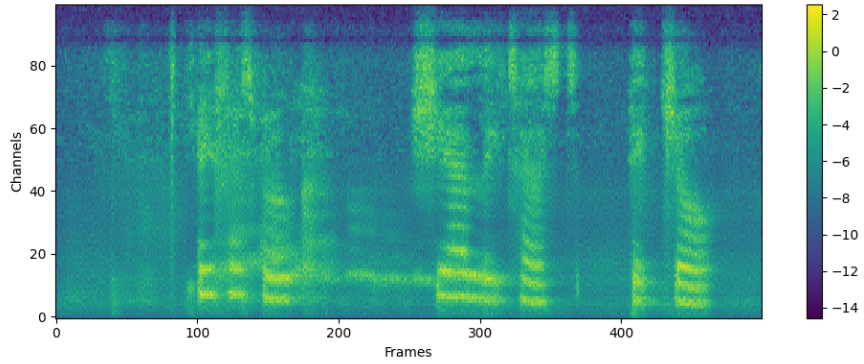

Figure 2: Predicted Output

This approach allows us to generate an unlimited variety of degraded audio samples, closely mimicking real-world scenarios and enhancing the model's generalization capabilities.

4 Conclusions and Future Work

In this paper, we presented a unified approach to audio quality restoration using flow-matching transformers. Our method demonstrates state-of-the-art performance across a wide range of degradation types, including reverberation, noise, compression artifacts, and low sampling rate issues. The proposed model shows strong generalization capabilities, effectively handling unseen combinations of degradations. To foster further research and

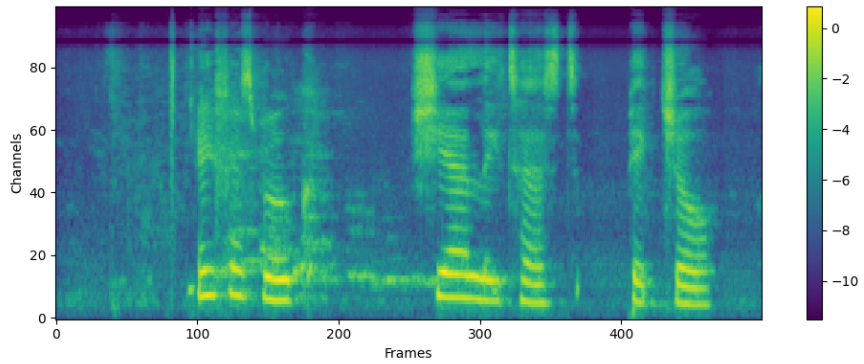

Figure 3: Ground Truth

development in this area, we are open-sourcing our model:

- A 301 million parameter transformer, which represents our full-scale model with state-of-the-art performance.

These model, along with pre-trained weights and example code, will be made available on our GitHub repository, enabling researchers and practitioners to build upon our work and apply it to various audio restoration tasks.

Future work could explore:

- Extension to real-time processing for live audio applications
- Incorporation of perceptual loss functions to further improve subjective quality
- Application to other audio domains, such as music restoration or environmental sound enhancement
- Development of even more efficient model architectures while maintaining high restoration quality
- Further exploration of self-supervised learning techniques, including more advanced real-time degradation algorithms and adaptive difficulty scaling based on model performance

We believe that by open-sourcing our models, we can accelerate progress in the field of audio quality restoration and encourage innovative applications across various domains.

References

[1] 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.

[2] S. E. Eskimez, X. Wang, M. Thakker, C. Li, C.-H. Tsai, Z. Xiao, H. Yang, Z. Zhu, M. Tang, X. Tan, Y. Liu, S. Zhao, and N. Kanda, "E2 tts: Embarrassingly easy fully non-autoregressive zero-shot tts," 2024. [Online]. Available: https://arxiv.org/abs/2406.18009

[3] N. Guo, T. Nakatani, S. Araki, and T. Moriya, "Modified parametric multichannel wiener filter
for low-latency enhancement of speech mixtures with unknown number of speakers," 2023. [Online]. Available: https://arxiv.org/abs/2306.17317

[4] S. Pascual, A. Bonafonte, and J. Serrà, "Segan: Speech enhancement generative adversarial network," 2017. [Online]. Available: https://arxiv.org/abs/1703.09452

[5] Y. Hu, Y. Liu, S. Lv, M. Xing, S. Zhang, Y. Fu, J. Wu, B. Zhang, and L. Xie, "Dccrn: Deep complex convolution recurrent network for phase-aware speech enhancement," 2020. [Online]. Available: https://arxiv.org/abs/2008.00264

[6] Y. Lipman, R. T. Q. Chen, H. Ben-Hamu, M. Nickel, and M. Le, "Flow matching for generative modeling," 2023. [Online]. Available: https://arxiv.org/abs/2210.02747