Generative Audio Synthesis in Sound, Speech, and Music

Xinlei Niu

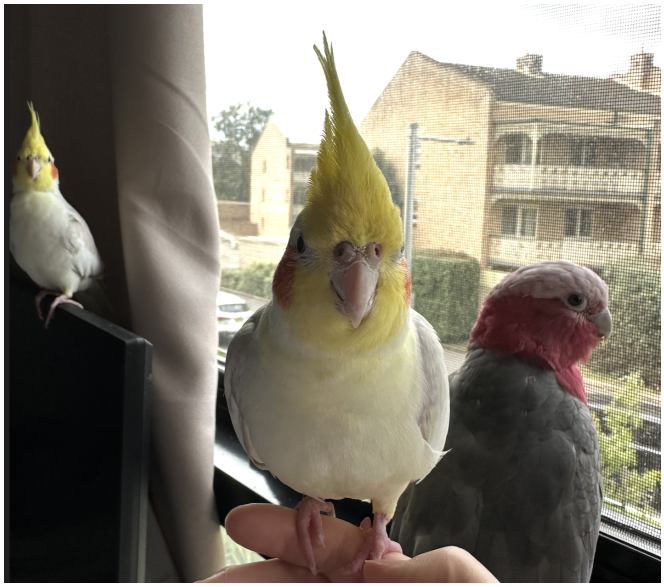
A thesis submitted for the degree of
Doctor of Philosophy
The Australian National University

August 2026

Except where otherwise indicated, this thesis is my own original work.

Xinlei Niu
13 August 2026

To my beloved family, my dearest husband, and my sweetest feathered friends,
whose love and support sustained me throughout this journey.



Declaration

I acknowledge the use of ChatGPT to proofread my writing and provide grammar corrections during the editing of this thesis.
I critically reviewed the ChatGPT feedback and, where appropriate, revised the writing using my own words and expressions.

Xinlei Niu
13 August 2026

Acknowledgments

This thesis would not have been possible without the generous support, guidance, and encouragement of many people.

First and foremost, I would like to express my deepest gratitude to my advisors, Dr. Charles Martin, Dr. Christian Walder, and Dr. Jing Zhang, for their invaluable guidance, insightful feedback, and constant encouragement throughout my PhD journey. Their expertise and vision have profoundly shaped both my research and my academic development. I would especially like to thank Dr. Jing Zhang for her unwavering support and encouragement; without her, completing this thesis would not have been possible.

I would also like to thank Dr. Miaomiao Liu for her valuable suggestions and encouragement, and I am especially grateful to my undergraduate supervisor, Dr. Yuhui Deng, who first introduced me to the field of machine learning and inspired my journey into research.

I am deeply grateful to my industry internship mentors, Dr. Kin Wai Cheuk (SonyAI), Mr. Dylan Harper-Harris (Dolby), and Dr. Jianbo Ma (Dolby), whose guidance, inspiration, and support have been a constant source of motivation throughout my research journey.

I extend my appreciation to my collaborators and colleagues during my internships at SonyAI and Dolby, whose discussions, feedback, and companionship greatly enriched my research experience. Special thanks to Mr. Naoki Murata, Dr. Chieh-Hsin Lai, Dr. Michele Mancusi, Dr. Woosung Choi, Dr. Giorgio Fabbro, Dr. Wei-Hsiang Liao, Dr. Yuki Mitsufuji, Dr. Junghyun Koo, Dr. Marco Martinez, and Mr. Yukara Ikemiya for their contributions, support, and inspiration during this journey.

I am grateful to The Australian National University for providing financial support and resources that made this research possible.

On a personal note, I owe endless thanks to my friends Huiyu Gao, Shidi Li, Wei Mao, Ruyi Zha, Xiaoxiao Sun, Yifu Wang, Zhenyang Yu, Ziang Cheng, Yue Cao, Xiangyu Zhang, Ashkan Taghipour, Ruiqi Wang, Matthew Aitchison, Gabriel Raya, Sam Cantrill, Minsik Choi, Yichen Wang, Sandy Ma, and Soumya Sai Vanka for their encouragement and joy during challenging times. I am also grateful for the comfort and companionship of my dearest feathered friends, Fifteen and Sixteen (cockatiels) and TT (galah), who brightened many difficult days throughout these four years.

Finally, I dedicate this thesis to my family. To my husband, Jiayu Yang, for his patience, understanding, and unwavering support. And to my parents for their unconditional love, support, and sacrifices that made my education possible.

Authorship Attribution Statement

The following papers are published or submitted as first-author papers, where I have done substantial work in ideation, research, experiments, analysis, and writing. All these six papers are available as open-access publications, and no additional permissions were required. The development of ideas and research was undertaken under the guidance of my supervisory panels, Dr. Charles Martin, Dr. Christian Walder, and Dr. Jing Zhang, and my industry internship mentors. All published papers presented in the thesis were written collaboratively with them.

Peer-Reviewed Publications

1. **Niu, X.**, Zhang, J., Walder, C., & Martin, C. P. (2024, April). Soundlocd: An Efficient Conditional Discrete Contrastive Latent Diffusion Model For Text-to-Sound Generation. In ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (pp. 261-265). IEEE. (In Chapter 3)

2. **Niu, X.**, Walder, C., Zhang, J., & Martin, C. P. (2024, July). Latent Optimal Paths by Gumbel Propagation for Variational Bayesian Dynamic Programming. In International Conference on Machine Learning (pp. 38316-38343). PMLR. (In Chapter 5)

3. **Niu, X.**, Zhang, J., & Martin, C. P. (2024). HybridVC: Efficient Voice Style Conversion with Text and Audio Prompts. In Proc. Interspeech 2024 (pp. 4368-4372). (In Chapter 6)

4. **Niu, X.**, Cheuk, K. W., Zhang, J., Murata, N., Lai, C. H., Mancusi, M., ... & Mitsufuji, Y. (2026, March). Steermusic: Enhanced musical consistency for zero-shot text-guided and personalized music editing. In Proceedings of the AAAI Conference on Artificial Intelligence (Vol. 40, No. 3, pp. 2000-2010). (In Chapter 8)

5. **Niu, X.**, Ma, J., Harper-Harris, D., Zhang, X., Martin, C. P., & Zhang, J. (2026, May). Beyond Video-to-SFX: Video to Audio Synthesis with Environmentally Aware Speech. In ICASSP 2026-2026 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (pp. 22697-22701). IEEE. (In Chapter 4)

Preprints

1. **Niu, X.**, Zhang, J., & Martin, C. P. (2024). SoundMorpher: Perceptually-Uniform Sound Morphing with Diffusion Model. arXiv preprint arXiv:2410.02144. (In Chapter 7)

Abstract

Audio is an important modality through which humans perceive and interpret the world. From speech and environmental sounds to music, it contains linguistic, emotional, and contextual information that can represent artistic expression. With the rapid progress of machine learning and generative AI, generative audio synthesis has emerged as a powerful approach to automatically producing realistic and diverse sounds, speech, and music content. However, current methods have challenges in requiring large computational resources, lacking flexibility across audio content, and offering limited controllability to users in practice.

This thesis aims to tackle these limitations by developing audio synthesis techniques that are efficient, flexible, and controllable. This thesis explores how generative models can be designed to operate effectively under training resource constraints, adapt across different and diverse creative audio content, and allow for fine-grained and customized control to users. To achieve these goals, six complementary methods are proposed in this thesis. Among them, *SoundLoCD* in chapter 3 and *BDP* in chapter 5 focus on training efficiency, introducing a lightweight diffusion framework and a discrete latent optimal path modeling framework that preserve quality while reducing training computational requirements and strategies. *HybridVC* in chapter 6 and *SoundMorpher* in chapter 7 advance flexibility by enabling flexible voice conversion and seamless sound morphing across multi-modal conditions. *BVS* in chapter 4 and *SteerMusic* in chapter 8 enhance controllability of synthesis, which allows fine-grained alignment between audio and visual context and empowers user-driven editing in music generation.

In summary, these contributions form a unified perspective on AI-generated content of audio synthesis, which treats speech, sound, and music not as isolated problems but as interconnected domains of users' immersive auditory experience. The resulting proposed methods move toward a future where audio generation is accessible, adaptive, and deeply integrated into creative, assistive, and interactive applications.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Overview

Audio is a fundamental modality through which humans perceive and interact with the world [Oxenham, 2018; O'Callaghan, 2009]. It contains rich information that humans perceive by hearing, ranging from linguistic content in speech, environmental scenes in sounds, and structured harmony in music. Techniques for automatically synthesizing audio have a broad real-world impact. They allow natural and intelligible speech synthesis for assistive purposes, high-fidelity sound effects for virtual environments, and music generation for both professional composition and consumer creativity. These applications make audio synthesis central to entertainment, education, accessibility, and interactive media, enabling more immersive and creative human–machine interactions.

With advances in machine learning and generative AI, generative audio synthesis is the task of producing diverse, realistic, and controllable audio content using generative AI. Unlike traditional approaches, generative models, such as variational autoencoders (VAEs), generative adversarial networks (GANs), and diffusion probabilistic models (DPMs), enable learning from large real audio datasets, which can efficiently estimate complex statistical distributions from real audio samples and generate synthesized samples from the learned distribution. Generative audio can be broadly categorized into three main domains according to the contents: *speech*, which prioritizes the intelligibility and naturalness of human voice; *environmental sounds*, which represent general real-world context and environmental scenes; and *music*, which emphasizes structured compositions and creative contents. Although each domain presents distinct challenges, they share the common goal of producing audio that is perceptually convincing and contextually coherent.

In practice, the boundaries between these domains are fluid rather than strictly isolated from each other. fig. 1.1 provides an example of the intersection of different audio domains, which leads to different applications in real-world scenarios. For example, film production not only requires Foley sounds but also environmentally aware speech and background music that together create a coherent auditory experience. Similarly, virtual reality and video game environments require systems that can seamlessly integrate speech, sounds, and music into a unified audio scene. Motivated by the needs of real-world applications, this thesis explores a unified perspec-

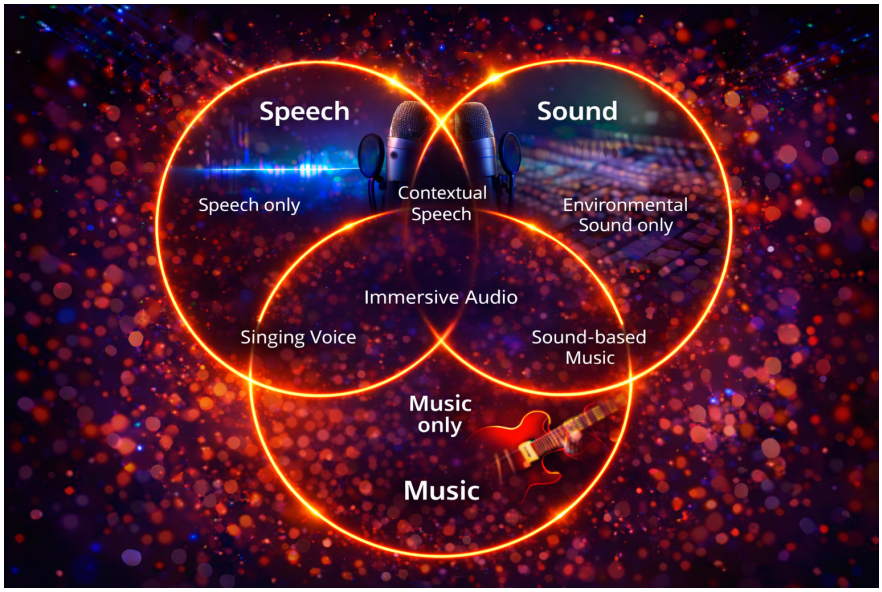


Figure 1.1: An example of intersections in generative audio domains.

tive on generative audio synthesis, treating the generated audio content in speech, sound, and music not as strictly isolated tasks but as intersecting and complementary domains. The goal is to develop AI-generated content (AIGC) methods that can specialize within an audio domain or generalize across different domains of the audio content, depending on the requirements of different needs of users in applications, thereby enabling more flexible and contextually adaptive audio generation in real-world applications.

1.2 Motivation

AIGC in the audio domain leverages generative models to automatically create diverse audio content, including speech, music, and environmental sounds that mimic or augment human creativity and perception. In the speech synthesis domain, methods such as WaveNet, FastSpeech, VALL-E, and lip2speech [van den Oord et al., 2016; Ren et al., 2019; Wang et al., 2023a; Hegde et al., 2022] emphasized the intelligibility and naturalness of synthesized speech from modalities such as transcripts or videos. In the sound generation domain, models like Diff-Foley, DiffSound and AudioGen [Yang et al., 2023b; Kreuk et al., 2023; Chen et al., 2020b] produced ambient sound effects that aligned with the given conditions, such as text descriptions and videos. While in the music synthesis domain, approaches such as Jukebox, PopMusic Transformer, MusicGen, and MusicLM [Dhariwal et al., 2020; Huang and Yang, 2020; Copet et al., 2023; Agostinelli et al., 2023] generated musical compositions from cues such as text descriptions, MIDI, or lyrics. While these task-specific methods demonstrated high-quality results for their targeted applications, they were inherently restricted to narrowly defined tasks and may not be easily extended to

accommodate broader generative demands or mixed-modality scenarios.

Recent years have seen a shift toward general and multimodal audio generation frameworks, such as AudioLDM2 [Liu et al., 2024c], Fugatto [Valle et al., 2025], UniAudio [Yang et al., 2024a], and Make-An-Audio [Huang et al., 2023a]. These models can generate audio conditioned on diverse modalities, such as text, images, or video, and represent a step toward unifying previously independent tasks. Despite their progress, such frameworks still face important limitations: they often rely on huge computational resources and large sets of high-quality data for training, produce a single auditory modality at a time (i.e., one type of audio content such as speech only or music only), and provide limited fine-grained control over the semantic structure. Consequently, they remain challenging to scale efficiently, inflexible in enabling creative editing or multi-domain fusion, and may not meet user demands to cross auditory modalities. While general models demonstrate strong versatility across audio generation tasks, specialist models are still necessary to achieve the high quality, precision, and efficiency required in specific applications.

Despite rapid progress in domain-specific and generalized audio generation models, several key challenges hinder their broader applicability and practical use. These challenges primarily revolve around three critical dimensions essential for real-world applications:

- **Efficiency**: State-of-the-art generative audio synthesis methods, such as DPM-based models and transformer-based language models, achieve impressive quality on generated audio content. Such models demand large computational requirements and incur slow sampling, as noted in diffusion-based methods [Ho et al., 2020a; Yang et al., 2023b] and autoregressive architectures such as WaveNet [van den Oord et al., 2016]. Even recent large-scale models [Borsos et al., 2023a; Agostinelli et al., 2023; Hu et al., 2026] remain impractical for resource-constrained environments. This restricts their usability and extensibility in settings with limited resources, such as mobile devices or even individual user computers or offline use. Therefore, it is crucial to improve the training efficiency and reduce the computational requirements of current generative audio synthesis methods while balancing the generation quality.

- **Flexibility**: Audio inherently contains complex perceptual attributes such as timbre in speech and music. For instance, in the application of music editing, creators usually require diverse manipulation capabilities, such as changing instrument timbres and editing music styles, to support creative workflows. Neural audio models often struggle to support nuanced edits such as fine-grained timbre manipulation or style modification, as highlighted in speech [Li et al., 2023b,a] and music systems [Dhariwal et al., 2020; Engel et al., 2020; Plitsis et al., 2024]. This lack of flexibility limits their usefulness in creative applications.

- **Controllability**: While existing methods can produce high-quality audio, they usually lack fine-grained control over semantic content or temporal structure.

Environmental sounds, for example, involve complex, time-ordered events that are difficult to reproduce accurately. Prior work in text-to-sound and video-to-audio synthesis highlights challenges in achieving fine-grained semantic and temporal alignment [Yang et al., 2023b; Kreuk et al., 2023; Luo et al., 2023; Lee et al., 2024a; Wang et al., 2024b], and similar issues arise in speech synthesis, where prosodic and contextual control remain limited [Wang et al., 2023a; Lee et al., 2024b]. Therefore, enabling fine-grained semantic and temporal control of generative audio synthesis remains an open and significant challenge.

Addressing these challenges is essential for advancing generative audio synthesis toward methods that are more efficient, flexible, and controllable, which are capable of supporting real-world applications such as film production, gaming, virtual reality, and assistive technologies.

1.3 Research Questions

Building on the motivations outlined in section 1.2, this thesis aims to explore solutions for the three central research questions that address the core challenges of efficiency, flexibility, and controllability in generative audio synthesis.

- **Research Question 1 (Efficiency):** Can generative audio models be designed to efficiently extend to new tasks or datasets while using limited computational resources, without sacrificing synthesis quality?
- **Research Question 2 (Flexibility):** Can a model support flexible audio generation across diverse domains and user demands, enabling seamless manipulation or cross-domain fusion of speech, sounds, and music?
- **Research Question 3 (Controllability):** Can generative models provide fine-grained control over semantic, temporal, and acoustic perception dimensions to accurately fulfill complex user requests?

Together, these research questions aim to push generative audio synthesis toward models that are **efficient**, **flexible**, and **controllable**, capable of adapting to the diverse requirements of real-world applications.

1.4 Contribution

In this thesis, we present six methods designed to address the three primary research challenges in generative audio synthesis (i.e., (1) efficiency, (2) flexibility, and (3) controllability) as outlined in section 1.3. These methods are developed and evaluated across six core tasks in the sound, speech, and music synthesis domain: text-to-sound generation, video-to-audio generation, text-to-speech and singing voice synthesis, voice conversion, sound morphing, and text-guided music editing. Together, they form a coherent line of research that advances the state-of-the-art performance

and demonstrates the potential of generative audio systems to support real-world applications. An overview of the six methods is provided in table 1.0, with each method corresponding to a published paper and addressing at least one of the identified research challenges. The specific contributions of this thesis are summarized as follows:

1. **RQ1: Efficiency** We introduce methods to improve training efficiency under limited computational resources and pipelines in generative audio models without sacrificing quality.

   - In chapter 3, we propose a LoRA-based discrete latent diffusion framework (called SoundLoCD) that reduces computational overhead while strengthening condition–output alignment in text-to-sound generation.
   - In chapter 5, we develop a theoretical foundation for discrete latent modeling using Gumbel propagation and variational dynamic programming (called BDP) that enables efficient search and optimization over structured token spaces.

   Together, these contributions provide scalable strategies for training and inference, ensuring that generative audio models can be deployed in resource-constrained or real-time settings.

2. **RQ2: Flexibility**

   We design pipelines that extend beyond narrow task boundaries, enabling a single framework to support diverse applications and modalities.

   - In chapter 6, we introduce HybridVC, a controllable voice conversion model that supports hybrid prompts (text and speaker embeddings), demonstrating flexible transfer while preserving linguistic and speaker identity.
   - In chapter 7, we propose a diffusion-based sound morphing framework guided by a perceptual distance metric called SoundMorpher, which supports static, dynamic, and cyclostationary morphing without retraining on task-specific data.

   These works collectively highlight the potential of flexible generative audio systems to adapt across tasks and domains in an open-world setting.

3. **RQ3: Controllability**

   We propose methods that give users more precise control over generative output, addressing the need for intelligibility, alignment, and user-driven editing.

   - In chapter 4, we introduce BVS, a two-stage video-to-audio framework that disentangles speech and ambient sounds, ensuring intelligible speech while aligning ambient effects with visual context.

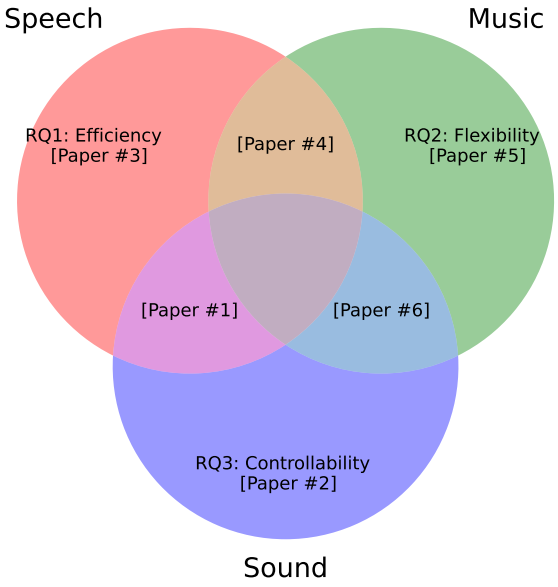


Figure 1.2: Research questions and paper map in this thesis.

- In chapter 8, we propose SteerMusic, which demonstrates controllability in music by enabling style transfer and fine-grained user-driven edits while maintaining instruction-irrelevant music attribute consistency.

These contributions advance the ability of generative audio systems to respect multimodal conditions and user intent, moving towards practical, interactive use cases.

As shown in fig. 1.2, the six papers form a coherent research program to the three research questions introduced in section 1.3. Some proposed methods, such as SteerMusic (paper 6), span multiple RQs by addressing both flexibility and controllability, while HybridVC (paper 4) links efficiency and flexibility and SoundLoCD (paper 1) enhances both controllability and training efficiency on existing works. This overlapping structure highlights a central theme of the thesis: progress in generative audio synthesis often requires balancing efficiency, flexibility, and controllability rather than treating them in isolation. Together, the six works advance the field toward open-world generative audio systems capable of producing high-quality, adaptable, and user-aligned audio across speech, sound, and music domains.

1.5 Thesis Outline

As illustrated in fig. 1.3, this thesis is organized as the following nine chapters:

- chapter 2 Background and Literature Review. This chapter includes the technical backgrounds and literature review of related tasks in the generative audio

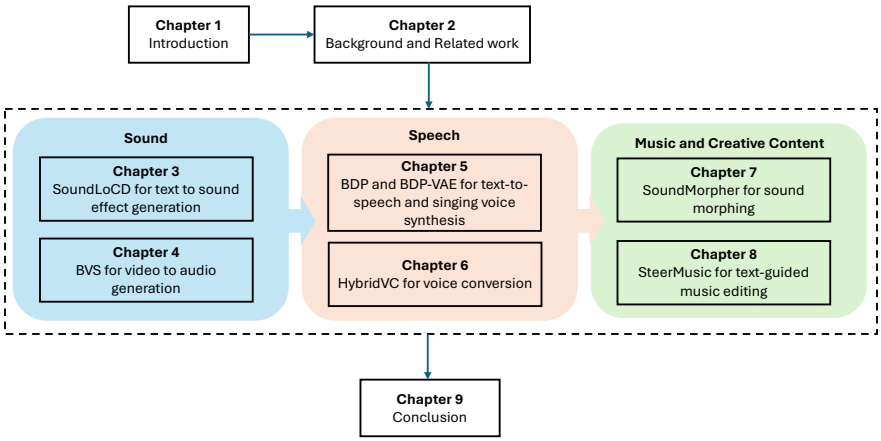


Figure 1.3: An overview of the thesis structure.

synthesis domain.

- chapter 3 An Efficient Conditional Discrete Contrastive Latent Diffusion Model for Text-to-Sound Generation. This chapter proposes SoundLoCD, which is a text-to-sound effect generation model designed for improving the training efficiency and controllability of existing text-to-sound methods.
- chapter 4 Video to Audio Synthesis with Environmentally Aware Speech. This chapter proposes Beyond Video-to-SFX (BVS), a two-stage generative model for improving controllability of existing methods in the video-to-audio task by integrating environmentally aware speech synthesis in parallel.
- chapter 5 Latent Optimal Paths by Gumbel Propagation for Variational Bayesian Dynamic Programming. This chapter provides a theoretical perspective on probabilistic generative frameworks for capturing unobserved structured latent optimal paths. We showed our method successfully allows a non-end-to-end text-to-speech and singing voice synthesis model to become an end-to-end training pipeline, at the same time ensuring the train and test consistency on capturing the discrete unobserved structured optimal paths in the latent space.
- chapter 6 Efficient Voice Style Conversion with Text and Audio Prompts. This chapter proposes HybridVC, a flexible voice conversion model, that supports text and audio prompts. Compared to existing approaches, HybridVC supports both flexible text-guided voice conversion and precise audio-guided voice conversion.
- chapter 7 Smooth and Consistent Sound Morphing with Diffusion Model. This chapter proposes SoundMorpher, an open-world sound morphing pipeline, that allows for diverse sound morphing tasks without the need of retraining on specific datasets. SoundMorpher opens a new direction on sound morphing research by making use of a pre-trained audio generation model to perform flexible sound morphing.

- chapter 8 Enhanced Musical Consistency for Zero-Shot Text-Guided and Personalized Music Editing. This chapter proposes two methods for the music editing task, which are SteerMusic and SteerMusic+. Both methods aim to solve the limitation of existing text-guided music editing pipelines, that fail to preserve the source music content during style editing. Beyond text-guided music editing, SteerMusic+ is proposed for personalized editing, which allows for fine-grained control in text-guided music editing.
- chapter 9 Conclusions and Future Directions. In this chapter, we summarize our contributions, limitations of our work, and future research directions to further enhance the immersive audio experiences in real-world applications.

Table 1.1: Overview of the aims, methods, and results from each of the six papers included in this thesis.

	Paper	Aim	Methods	Results
ICASSP 2024 (Oral)	**[1] SoundLoCD** [Niu et al., 2024d]	Improve conditioning and training efficiency in text-to-sound generation	LoRA-based discrete latent diffusion model; contrastive alignment between text and outputs	Strong text-to-sound with lower compute (*efficiency*); improved condition–output correspondence (*controllable*).
ICASSP 2026	**[2] Beyond Video-to-SFX (BVS)** [Niu et al., 2026b]	Improve video-to-audio generation with intelligible speech	Two-stage model: audio semantic token prediction and acoustic generation conditioned on video and phonetic cues	Synchronize ambient sounds with intelligible speech (*controllable*); flexible to various real-world applications.
ICML 2024	**[3] Bayesian Dynamic Programming (BDP)** [Niu et al., 2024a]	Capturing unobserved structured relationships in conditional audio generative tasks	Gumbel propagation for variational Bayesian dynamic programming; BDP-VAE	Discrete optimal path for gradient optimisation; end-to-end efficient training strategy (*efficiency*).
InterSpeech 2024	**[4] HybridVC** [Niu et al., 2024b]	Promptable and controllable voice conversion	Latent generative model of a pre-trained VAE with text and audio prompts; contrastive multimodal alignment	Flexible conversion with hybrid prompts (*flexible*); efficient training with limited time and resources (*efficiency*).
In submission	**[5] SoundMorpher** [Niu et al., 2024c]	Smooth and consistent sound morphing across different audio morphing tasks	Diffusion-based morphing with embedding interpolation; smoothness refinement with binary search	Flexible morphing method without retraining on specific dataset (*flexible*); smoother and more natural transitions.
AAAI 2026 (Oral)	**[6] SteerMusic** [Niu et al., 2026a]	User-controlled music editing with enhanced source melody consistency	Score distillation; personalized concept-token manipulation	Coarse-grained zero-shot text-guided music editing (*flexible*); and fine-grained personalized music editing (*controllable*).

Chapter 2

Background and Related Work

This chapter establishes the technical background of generative audio synthesis necessary for contextualizing the contributions of this thesis. Research in generative audio synthesis can be traced back to early speech synthesis systems, notably the Voder [Story, 2019] demonstrated at the 1939 World's Fair, followed by decades of rule-based and parametric approaches that gradually improved intelligibility and control. The field has undergone a major transformation over the past decade with the emergence of deep learning, which has enabled data-driven audio generation and high-fidelity and diverse audio content synthesis across speech, sound effects, and music [van den Oord et al., 2016; Kreuk et al., 2023; Rouard et al., 2025; Wang et al., 2017b; Agostinelli et al., 2023]. While acknowledging this broader historical trajectory, this chapter primarily focuses on developments from the modern generative AI framework that directly inform the research presented in this thesis.

We begin by introducing typical generative modeling frameworks in section 2.1, followed by a review of major audio generation tasks across domains in section 2.2, audio editing tasks in section 2.3, and the sound morphing task in section 2.4. To support these discussions, section 2.5 summarizes commonly used audio representations in contemporary generative systems, while section 2.6 introduces neural vocoders that reconstruct waveforms from intermediate representations. Finally, section 2.7 provides a comprehensive overview of the objective evaluation metrics employed throughout this thesis.

2.1 Generative Models for Audio Synthesis

In this section, we introduce the technical background underlying generative frameworks for audio synthesis. We focus on several common generative frameworks that form the foundation of modern audio synthesis systems. These frameworks aim to model the underlying distribution of real audio data, enabling the generation of new audio samples by sampling from the estimated distribution. These models provide the conceptual and practical basis for the generative audio synthesis tasks discussed throughout this thesis.

In audio generative frameworks, the main objective is to approximate the under-

lying data distribution $p_{\text{data}}(x)$ of real-world audio samples $x \in \mathcal{X}$, where $\mathcal{X}$ represents the set of real-world audio data. The audio data can be a form of audio representation introduced in section 2.5. By learning a parametric model $p_\theta(x) \approx p_{\text{data}}(x)$ to approximate the real-world data distribution, these methods can generate new audio samples by sampling from the estimated distribution $x' \sim p_\theta(x)$, thereby, capturing the complex temporal, spectral, and perceptual characteristics present in natural sounds.

2.1.1 Autoregressive Models

Autoregressive (AR) models, such as WaveNet [van den Oord et al., 2016], model the intrinsic nature of audio sequences and generate audio sample by sample and predict the probability distribution of the next audio sample based on all the previous samples in an audio sequence. This autoregressive property ensures that the model only uses past information to predict the future, crucial for generating coherent and natural-sounding audio. Let an audio signal be represented as a discrete time-domain sequence $x = x_{1:S} = \{x_1, x_2, ..., x_S\}$, where x_s represents the s-th element in the sequence. An AR model factorizes the joint distribution over the full sequence according to the chain rule,

$$p_\theta(x_{1:S}) = \prod_{s=1}^{S} p_\theta(x_s \mid x_{<s}), \tag{2.1}$$

where S represents the number of timesteps in the audio sequence. $x_{<s}$ denotes all previous samples before sequence timestep s. These models achieved unprecedented quality in audio synthesis across speech and music domains but suffered from slow inference on long sequence generation due to the sequential generation property.

2.1.2 Generative Adversarial Networks

Generative adversarial networks (GANs) [Goodfellow et al., 2020] introduce an adversarial training strategy between a generator G and a discriminator D. The generator learns to produce realistic samples $G(z)$ from a latent prior $z \sim p_z(z)$, while the discriminator learns to distinguish between real audio samples $x \sim p_{\text{data}}(x)$ and generated samples. The goal of a GAN is to generate realistic data that the discriminator cannot distinguish as fake. It learns from the discriminator's feedback, using its mistakes to improve the output. A GAN is trained by the standard minimax objective, which is defined as

$$\mathcal{L}_{\text{GAN}} = \min_G \max_D \mathbb{E}_{x \sim p_{\text{data}}}[\log D(x)] + \mathbb{E}_{z \sim p_z}[\log(1 - D(G(z)))]. \tag{2.2}$$

GANs are known for generating high-quality audio with efficient sampling, but training stability and mode collapse remain ongoing challenges.

2.1.3 Normalizing Flows

Normalizing flow (NF) models, such as WaveGlow [Prenger et al., 2019] and Glow-TTS [Kim et al., 2020], are a class of likelihood-based generative models that learn an invertible transformation between a simple prior distribution to the complex distribution of real audio data p_{data}. Formally, an NF defines a bijective function $f_\theta : z \leftrightarrow x$, where z is a latent variable drawn from a simple prior (e.g., $z \sim N(0, \mathbf{I})$). The probability of x can be computed exactly using the change of variables formula,

$$p_\theta(x) = p(z)|\det\frac{\partial f_\theta^{-1}(x)}{\partial x}| \tag{2.3}$$

During training, the model learns the parameters θ to maximize the log-likelihood of the data. Since f_θ is invertible and differentiable, NFs support both efficient sampling via $\hat{x} = f_\theta(z)$ and likelihood estimation via $z = f_\theta^{-1}(x)$. The applicability of NFs is limited to transformations that are differentiable and invertible.

2.1.4 Variational AutoEncoder

Variational AutoEncoder (VAEs) [Kingma and Welling, 2013] introduce a probabilistic framework that learns a latent representation z from a prior distribution $z \sim p(z)$ for data x. An encoder parameterizes an approximate posterior $q_\phi(z|x)$, while a decoder reconstructs the data as $p_\theta(x|z)$. VAEs are optimized via the evidence lower bound (ELBO), which balances a reconstruction loss with a Kullback-Leibler (KL) divergence term, creating a trade-off between the encoder and decoder. The ELBO is defined as

$$\mathcal{L}_{\text{ELBO}} = \mathbb{E}_{q_\theta(z|x)}[log p_\theta(x|z)] - \lambda \mathcal{D}_{\text{KL}}(q_\theta(z|x)||p(z)). \tag{2.4}$$

where $\mathcal{D}_{\text{KL}}$ denotes the KL divergence and λ is a scale parameter. The ELBO loss optimizes the decoder by the reconstruction term and the encoder by the KL divergence regularization. This procedure is conceptually similar to the Expectation–Maximization (EM) algorithm. This is a trade-off between reconstruction accuracy and latent regularization. However, this tension can cause posterior collapse [van den Oord et al., 2017a], where the decoder is too powerful and the latent variable carries little information.

2.1.5 Diffusion Probabilistic Models

Diffusion probabilistic models (DPMs) [Ho et al., 2020a] reformulate generation as a gradual the denoising process. DPMs can be interpreted by three complementary views: the variational view, the score-based view, and the flow-based view [Lai et al., 2025]. In this part, we introduce the DPMs by the variational view. The forward diffusion process gradually adds Gaussian noise to data x over T diffusion timesteps through a Markov chain, as

$$q(x_t|x_{t-1}) = \mathcal{N}(\sqrt{1-\alpha_t}x_{t-1}, \alpha_t\mathbf{I}). \tag{2.5}$$

where α_t is the variance schedule that controls the magnitude of the noise at the diffusion timestep t. The reverse process learns to invert this corruption by approximating $p_\theta(x_{t-1}|x_t)$, effectively denoising a sampled Gaussian noise $x_t \sim \mathcal{N}(0, \mathbf{I})$ to data step by step. DPMs can be optimized by the variational lower bound (VLB) [Kingma et al., 2021], which measures how well the learned reverse process approximates the true posterior of the forward diffusion. Specifically, the evidence lower bound decomposes into a sum of per-timestep KL divergences between the true reverse posterior $q(x_{t-1}|x_t, x_0)$ and the model distribution $p_\theta(x_{t-1}|x_t)$, and a reconstruction term at time step $t = 0$

$$\mathcal{L}_{\text{VLB}} = \mathbb{E}_q[D_{\text{KL}}(q(x_t|x_0)||p_\theta(x_t)) + \sum_{t=2}^{T} D_{\text{KL}}(q(x_{t-1}|x_t, x_0)||p_\theta(x_{t-1}|x_t)) - log p_\theta(x_0|x_1)] \quad (2.6)$$

Ho et al. [2020a] shows that this objective can be simplified into a denoising score-matching objective by reparameterizing the forward process into a closed-form noising step $q(x_t|x_0)$. The resulting training loss reduces to predicting the added Gaussian noise ϵ at each timestep:

$$\mathcal{L}_{\text{DPM}} = \mathbb{E}_{x_0,\epsilon,t}[\|\epsilon - \epsilon_\theta(x_t, t)\|^2] \quad (2.7)$$

DPMs achieve state-of-the-art perceptual quality and diversity in audio generation, which is widely applied to different audio domains. However, Denoising Diffusion Probabilistic Models (DDPMs) suffer from inherently slow sampling, as they require a large number of sequential denoising steps to approximate the reverse diffusion process. We then introduce an alternative solution to improve the sampling efficiency of DPMs while maintaining the sample quality.

2.1.5.1 Denoising Diffusion Implicit Model (DDIM)

To reduce the inference time of DPMs, DDIM provides an alternative solution and enables significant sampling efficiency while maintaining high generation quality. Given DPM parameterized by ϕ and a diffusion process defined as $q(x_t|x_0) := \mathcal{N}(x_t; \sqrt{\alpha_t}x_0, (1-\alpha_t)I)$, where the α_t represents the variance of the forward diffusion process at time step t, x_t represents the noised latent representation of the data x_0. To reduce computational demands on DPM inference, DDIM [Song et al., 2021a] defines an update rule in the reverse diffusion process, which the formulation is given by

$$x_{t-1} = \underbrace{\sqrt{\alpha_{t-1}}(\frac{x_t - \sqrt{1-\alpha_t}\epsilon_\phi^{(t)}(x_t)}{\sqrt{\alpha_t}})}_{\text{'predicted } x_0\text{'}} + \underbrace{\sqrt{1-\alpha_{t-1}-\sigma_t^2}\epsilon_\phi^{(t)}(x_t)}_{\text{'direction pointing to } x_t\text{'}} + \underbrace{\sigma_t\epsilon_t}_{\text{'random noise'}} \quad (2.8)$$

where σ_t is a free variable that controls stochasticity in the reverse process. The insight of DDIM is using a deterministic forward process that shares the same training objective as the original DPMs, while allowing a non-Markovian reverse diffusion process. Moreover, setting σ_t to 0 yields a deterministic update rule that establishes a deterministic mapping between the clean sample x_0 and its latent representation x_T. The inverse mapping from x_0 to x_T is referred to as *DDIM inversion*. It can be formulated as

$$\frac{x_{t+1}}{\sqrt{\alpha_{t+1}}} - \frac{x_t}{\sqrt{\alpha_t}} = \left(\sqrt{\frac{1-\alpha_{t+1}}{\alpha_{t+1}}} - \sqrt{\frac{1-\alpha_t}{\alpha_t}} \right) \epsilon_\phi^{(t)}(x_t). \tag{2.9}$$

In practice, DDIM constructs a sequence of time steps $\{t_1, t_2, ..., t_S\}$ with $S \ll T$, where T is the number of steps used during training. The model then performs sampling only along this subset of steps, effectively skipping intermediate diffusion steps. This property enables DDIM to accelerate inference time substantially compared to DDPM, while maintaining good sampling quality.

2.1.5.2 Classifier-Free Guidance (CFG)

A key challenge in conditional diffusion models is to balance the fidelity and conditional alignment. Classifier guidance can improve conditional consistency but requires training an external classifier on noisy inputs, which is computationally expensive and often unstable. CFG [Ho and Salimans, 2021] addresses this issue by removing the classifier entirely and instead training the diffusion model to produce both conditional and unconditional predictions. This enables guidance at sampling time without additional networks or noise-aware classifiers. CFG is widely used in both image and audio diffusion models for its simplicity, stability, and strong improvements in conditional generation quality.

Given a diffusion model jointly trained on conditional and unconditional embeddings, in the sampling phase, samples can be generated using CFG [Ho and Salimans, 2021]. The prediction with the conditional and unconditional estimates is defined by the following equation

$$\epsilon_\phi^\omega(x_t, y, t) := \omega \epsilon_\phi(x_t, y, t) + (1-\omega)\epsilon_\phi(x_t, \varnothing, t) \tag{2.10}$$

where ω is the guidance scale that controls the trade-off between mode convergence and sample fidelity, and $\varnothing$ is a null token used for unconditional prediction.

2.2 Audio Generation Tasks

In this section, we move our focus to the audio generation domain, where the generative frameworks introduced in section 2.1 are applied to produce diverse forms of audio content. Building on the foundational principles of generative modeling, we provide a comprehensive review of related work across major audio generation

tasks, including generalized audio generation methods and specialized audio generation methods, which highlights how these generative frameworks enable practical and high-quality audio generation.

Audio generation refers to the task of producing desired audio signals using generative models and represents a central component of modern audio synthesis [Latif et al., 2023; Agostinelli et al., 2023; Yang et al., 2023b; Gutiérrez et al., 2025]. Existing approaches to audio generation can be broadly categorized into specialized models and generalized models. Generalized models aim to provide a unified framework capable of generating diverse audio types, such as speech, music, and environmental sounds, within a single system. These models typically leverage large-scale multimodal datasets and cross-domain learning strategies to capture shared acoustic representations across tasks. In contrast, specialized models focus on a specific domain, optimizing for high fidelity, controllability, and expressiveness in a well-defined task, such as text-to-speech synthesis, music composition, or sound effect generation. This section reviews related works in both generalized models and specialized models in the audio generation domain, highlighting the evolution from domain-specific systems to unified, cross-domain generative frameworks that advance toward generalized audio intelligence.

2.2.1 Generalized Audio Generation and Audio LLMs

Generalized audio generation models aim to synthesize diverse forms of audio, including speech, sound effects, and music, within a unified framework capable of capturing temporal structure and semantic conditioning. One of the pioneering approaches in this direction is WaveNet [van den Oord et al., 2016], which directly modeled raw waveforms and demonstrated high-quality audio synthesis for both speech and music. Building on this idea, Jukebox [Dhariwal et al., 2020] introduced a hierarchical VQ-VAE [van den Oord et al., 2017a]-based architecture capable of generating high-fidelity music with lyrics. Jukebox showed that a single model could learn complex and long-form audio dependencies while being conditioned on attributes such as genre, artist, and lyrics. This marks an important step toward universal audio generation through hierarchical tokenization and conditioning strategies.

Subsequent research advanced this paradigm by leveraging self-supervised learning (SSL) and language modeling techniques applied to audio token sequences. AudioLM [Borsos et al., 2023a] exemplified this shift by introducing a pipeline that first extracts semantic and acoustic tokens from raw audio using pre-trained encoders and then models these token sequences using Transformer-based language models. AudioLM achieved impressive results across both speech and music generation, demonstrating the feasibility of treating audio as a discrete sequence modeling problem analogous to natural language. SoundStorm [Borsos et al., 2023b] further improved the acoustic modeling stage by incorporating residual vector quantization (RVQ) and a parallel decoding framework, leading to higher fidelity and faster synthesis.

With the rapid success of diffusion-based generative models in vision and language, several works extended this paradigm to the audio domain. Notable exam-

ples include Make-A-Audio [Huang et al., 2023d,b], AudioLDM [Liu et al., 2023a], AudioGen [Kreuk et al., 2022], and TANGO [Ghosal et al., 2023; Majumder et al., 2024]. These models aim to achieve general-purpose, multi-modal conditions for guided audio generation across multiple audio domains, which span speech, sound effects, and music. For instance, AudioLDM employs a latent diffusion framework trained on large-scale datasets and supports generation from both text and audio prompts, while AudioGen uses autoregressive Transformers to model discrete audio tokens for coherent sound and speech synthesis. More recent extensions such as Valle et al. [2025]; Wang et al. [2025a]; Liu et al. [2024c]; Huang et al. [2023a]; Zhang et al. [2026]; Wang et al. [2024c] have further integrated large language models (LLMs) into the generation pipeline, enabling semantically richer conditioning and cross-modal capabilities such as text-to-audio and image-to-audio generation.

More recently, following the success of multimodal large language models (MLLMs) and audio large language models (LLMs), research has increasingly focused on integrating audio understanding and generation within a unified framework. Such models leverage shared multimodal representations, which are often built on top of large language model backbones, to support both understanding and generative tasks within the same architecture. This trend has enabled general-purpose systems capable of performing audio generation alongside broader multimodal reasoning [Xu et al., 2025; Hu et al., 2026; Tian et al., 2025b; Lu et al., 2024; Kong et al., 2024; Evans et al., 2026]. Compared with diffusion-based approaches, these models provide stronger semantic reasoning and cross-modal capabilities, while diffusion models remain advantageous in high-fidelity generation and fine-grained controllability. Integration of audio LLMs with diffusion-based generation represents a promising future research direction.

These developments mark a clear path from domain-specific audio synthesis pipelines toward general, unified, and multimodal audio generation frameworks, ultimately laying the groundwork for universal audio understanding and creative sound modeling.

Although generalized audio generation models provide a unified framework for producing diverse types of audio, audio generation remains particularly challenging due to the high variability, temporal complexity, and context-specific nature of real-world sounds. In many applications, such as film post-production, virtual environments, and video game development, sound effects must be precisely aligned with semantic or visual events, requiring fine-grained control over timing, texture, and contextual semantics. The following sections provide a comprehensive literature review on specialized models of audio generation tasks in the sound, speech, and music domains.

2.2.2 Sound Generation

Early sound generation methods mainly relied on label-to-sound mappings [Liu and Jin, 2024b,a], while conventional pipelines often relied on sample concatenation or manually engineered synthesis workflows incorporating handcrafted audio at-

tributes such as pitch, timbre, and temporal envelope [Engel et al., 2020]. To improve the flexibility of existing pipelines, recent research has explored *text-to-sound (T2S) generation*, which leverages natural language as a flexible and expressive conditioning modality. By conditioning on textual descriptions, these models enable semantically rich and controllable sound synthesis. DiffSound [Yang et al., 2023b] pioneered this direction with a discrete latent diffusion model trained on paired audio–caption data (e.g., AudioCaps [Kim et al., 2019a]), demonstrating that text prompts can guide environmental sound generation more effectively than label-based methods. Following this, AudioGen [Kreuk et al., 2023] proposed a GAN-based autoregressive model that generates waveforms using neural audio codecs [Zeghidour et al., 2021; Li et al., 2021] and a text encoder derived from a transformer language model [Raffel et al., 2020]. However, its autoregressive formulation results in high inference latency, making it less suitable for long-duration audio. Subsequent works [Liu et al., 2023a; Ghosal et al., 2023; Yuan et al., 2023] adopted latent diffusion models (LDMs) [Gu et al., 2022a; Song et al., 2021b; Ho et al., 2020b] trained on multiple datasets, enabling more general text-guided synthesis in the continuous latent space of pre-trained VAEs. Despite these advances, maintaining precise text–sound correspondence remains a major challenge: current models often fail to capture fine-grained acoustic attributes or temporal cues implied by language. In chapter 3, we introduce a method that explicitly strengthens text–sound correspondence during training, thereby improving both semantic consistency and generation quality.

However, text-guided audio generation often lacks fine-grained temporal control and precise synchronization, which are critical requirements in applications such as film production and interactive media, where Foley sounds must align accurately with frame-level visual cues. This limitation motivates the task of *video-to-audio (V2A) generation*, where the goal is to produce synchronized sounds that semantically match visual events. DiffFoley [Luo et al., 2023] was among the first to tackle this task, using a latent diffusion model combined with a contrastive audio–video pretraining (CAVP) framework for automatic Foley sound generation. Subsequent studies [Wang et al., 2024b; Xue et al., 2024; Lee et al., 2024a; Chen et al., 2024b; Wang et al., 2024a; Viertola et al., 2025] extended this paradigm, focusing on improving the synthesis quality and temporal alignment of ambient sounds. More recent models, such as VATT and MMAudio [Liu et al., 2024d; Cheng et al., 2025] incorporate text prompts alongside video inputs, enabling richer scene understanding, for example, generating the composite soundscape of "a man speaking while playing tennis.". However, in practice, videos include not only environmental sounds but also audio associated with human actions and speech. Despite these improvements, current V2A systems generally focus on the ambient sound-event level, which produces sound effects that are temporally synchronized but lacking contextual coherence and intelligible speech. This limits their applicability in scenarios requiring context-aware spoken content. In chapter 4, we propose a new V2A framework that addresses this limitation by generating synchronized ambient audio together with contextually appropriate, intelligible speech from video prompts. The proposed method enhances the controllability of existing V2A methods, which allows for fine-grained synchro-

nized audio generation with intelligible speech.

Although sound generation models in principle produce a wide variety of audio signals, including human voices and background music, they remain primarily focused on ambient and environmental sounds. Their architectures are typically not optimized for the linguistic structures inherent in speech or the temporal–harmonic organization characteristic of music composition. We then extend our review to the speech synthesis domain, where specialized models address these domain-specific challenges through structured conditioning, prosody modeling, and long-term temporal coherence.

2.2.3 Speech Synthesis

Speech synthesis aims to generate natural and intelligible speech, with applications ranging from virtual assistants to human–machine communication. Within this domain, *text-to-speech (TTS) synthesis* is the task that generates intelligible speech from given transcripts. Early TTS systems relied on statistical parametric synthesis [Zen et al., 2009], which modeled acoustic features such as pitch, spectral envelope, and duration using hidden Markov models (HMMs). Although these approaches produced intelligible speech, they lacked naturalness and expressive variability. The advent of deep learning revolutionized this field: WaveNet [van den Oord et al., 2016], an autoregressive model directly generating raw waveforms, achieved unprecedented perceptual quality. Subsequent TTS methods, such as Tacotron [Wang et al., 2017a] and Tacotron 2 [Shen et al., 2018b], improved speech naturalness by predicting mel-spectrograms from phoneme text, later converted to waveforms using neural vocoders such as WaveGlow [Prenger et al., 2019] and HiFi-GAN [Kong et al., 2020a]. However, these autoregressive models suffered from high inference latency. To address this, non-autoregressive (NAR) models like FastSpeech and FastSpeech 2 [Ren et al., 2019, 2020] introduced duration predictors and parallel generation pipelines, relying on pre-trained external phoneme aligners such as the Montreal Forced Aligner (MFA) [McAuliffe et al., 2017a] to retain intelligibility while accelerating synthesis. However, the performance of these models relies strongly on the ability of external aligners. To address this problem, end-to-end TTS systems, including Glow-TTS [Kim et al., 2020], VAENAR-TTS [Lu et al., 2021], and VQ-VAE-TTS [Yasuda et al., 2021], later emerged by employing dynamic programming, attention mechanisms, or CTC-based alignment [Graves et al., 2006]. Yet these methods struggle to maintain consistent phoneme-duration alignment between training and inference due to the discrete nature of duration modeling. In chapter 5, we address this limitation by introducing a probabilistic softening approach to dynamic programming, enabling a consistent end-to-end training framework for TTS and related tasks.

More recently, diffusion-based frameworks have shown promise in speech synthesis. Grad-TTS [Popov et al., 2021] and Diff-TTS [Jeong et al., 2021a] model mel-spectrogram generation as a denoising diffusion process, progressively refining speech representations from noise. These models yield superior fidelity and prosody diver-

sity compared to deterministic predictors. Further advances incorporate large-scale generative models [Wang et al., 2023a; Guo et al., 2023; Le et al., 2023; Wang et al., 2025b; Casanova et al., 2022; Wang et al., 2025c; Zhang et al., 2024a; Xinyuan et al., 2025], achieving multilingual, zero-shot speaker adaptation, and prompt-based controllability. Another relevant direction is *environmental context-aware speech synthesis*, which aims to produce speech that naturally blends with environmental acoustics and background noise. VoiceLDM [Lee et al., 2024b] pioneered this area using a latent diffusion model trained on noisy speech datasets and guided by a pre-trained ASR model [Rekimoto, 2023] to maintain intelligibility under noisy conditions. VoiceDiT [Jung et al., 2025] extends this work by incorporating image-conditioned synthesis. However, these models are constrained by the scarcity of large-scale datasets designed for context-aware speech generation, often requiring multi-stage training—first on synthetic noisy data, then fine-tuning with ASR-transcribed real-world speech. Related methods [Glazer et al., 2025; He and Liu, 2025] explore acoustic scene blending, mixing clean speech with ambient sounds to enhance perceptual realism, but they lack fine-grained temporal synchronization with dynamic audiovisual content.

2.2.4 Music Generation

Parallel to advances in sound and speech synthesis, music generation has emerged as a major research domain focused on creating structured, expressive, and aesthetically coherent compositions [Batlle-Roca et al., 2025]. Early music synthesis approaches relied on low-level control signals to ensure temporal alignment, such as pitch contours [Engel et al., 2020], lyrics-conditioned models [Yu et al., 2021; Dhariwal et al., 2020], and MIDI-conditioned systems [Yang et al., 2017]. While effective for constrained tasks, these methods required detailed symbolic or aligned data, limiting their generalization to open-domain music. Inspired by advances in natural language processing and multimodal learning, recent studies have shifted toward high-level conditioning using semantic prompts (e.g., text or images). Models such as [Kundu et al., 2024; Huang et al., 2023d; Chowdhury et al., 2024; Kreuk et al., 2022; Huang et al., 2023c; Lan et al., 2024; Le Lan et al., 2024; Li et al., 2024a; Saito et al., 2025] interpret natural-language descriptions (e.g., "a calm piano melody in a rainy atmosphere") to produce stylistically coherent and emotionally expressive music. This paradigm, termed text-guided music generation, enables intuitive human–AI collaboration in creative composition. Complementary research investigates melody-conditioned music generation [Copet et al., 2024; Novack et al., 2024b,a; Hou et al., 2025], where reference melodies provide temporal and tonal constraints, ensuring harmonic coherence. Recent work such as DreamSound [Plitsis et al., 2024] and Jen-1 Style [Chen et al., 2024a] explores personalized diffusion models [Ruiz et al., 2023; Gal et al., 2022; Kumari et al., 2023], which learn user-specific musical concepts from reference tracks. These models enable personalized music generation, allowing the synthesis of new compositions that reflect individual styles, emotions, or artistic identities through learned concept tokens embedded in textual

prompts. In summary, these developments reflect a clear trend toward semantically grounded, user-controllable music generation, bridging structured symbolic modeling and open-domain creative synthesis. Alongside advances in sound and speech synthesis, they represent a step toward unified multimodal generative audio frameworks capable of producing coherent, expressive, and contextually adaptive audio across domains.

2.2.5 Dynamic Programming in Audio Generation

Dynamic programming (DP) is a general algorithmic paradigm used to solve optimization problems efficiently that exhibit overlapping subproblems and optimal substructure. In speech and music generation, DP-based methods, such as dynamic time warping (DTW) or alignment algorithms, are often employed to model or infer temporal correspondences between sequences. These techniques are widely used in tasks like speech synthesis, speech recognition, and other sequence-to-sequence frameworks to estimate alignments, find optimal paths, or enforce structural dependencies between acoustic features and conditioning inputs.

Since traditional DP finding optimal paths is non-differentiable, there exist many alternative works that integrate DP into neural networks by involving a convex optimization problem [Amos and Kolter, 2017; Djolonga and Krause, 2017]. Instead, Mensch and Blondel [2018] proposed a unified DP framework by turning a broad class of DP problems into a DAG and obtaining the optimal path by a max operator smoothed with a strongly convex regularizer. This work can be applied in structured prediction tasks [BakIr et al., 2007] under supervised learning. Inspired by Mensch and Blondel [2018], we proposed a probabilistic softening solution to seek stochastic optimal paths with a path distribution under a DAG. Graphical models such as Bayesian networks [Heckerman, 1998] learn dependencies of random variables based on a DAG, our method treats the paths of a DAG, not the nodes, as random variables of a Gibbs distribution.

In many conditional audio generative tasks such as speech and music generation, models usually rely on structured dependencies of data and the given condition (e.g., phoneme text and music notations), in which the dependencies are unobserved. Ren et al. [2020] and Liu et al. [2022] use a multiple training strategy by obtaining the sparse dependencies from an external DP-based technique[McAuliffe et al., 2017b] at first, then use the outputs as additional inputs to the model. Kim et al. [2020] integrates DP on a Glow-based model to obtain unobserved monotonic alignment in parallel. Other models with structured latent representations, such as HMMs [Rabiner, 1989] and PCFGs [Petrov and Klein, 2007] make strong assumptions about the model structure, which could limit their flexibility and applicability. In contrast to these approaches, we propose a unified framework to enable the VAEs [Kingma and Welling, 2014] to capture structural latent variables (i.e., sparse optimal paths), allowing for flexible adaptation to a variety of downstream tasks.

The attention mechanism [Vaswani et al., 2017] is widely applied for obtaining unobserved dependencies in many seq-to-seq tasks. Deng et al. [2018] makes use

of the properties of VAEs and captures the unobserved non-structural dependencies by learning latent alignment with attention in VAEs. However, to obtain structural constraints, attention strongly relies on model structures and other techniques such as DP to extract marginalization of the attention alignment distribution [Yu et al., 2016b,a]. Different from latent alignment with attention, we target solving the optimal path problem in a unified framework that can be easily adapted to any structural constraint by defining DAGs (e.g., structural alignment). By capturing unobserved sparse shortest paths under the defined DAG in a latent space, our BDP-VAE facilitates the development of more explicit structural unobserved dependencies for a variety of applications. Secondly, attention mechanisms can be computationally expensive, especially for large input sequences. Solutions such as Chiu and Raffel [2018] reduce the computational complexity by involving DP. In chapter 5, we seek stochastic optimal paths that occur in linear time with respect to edge numbers of DAGs (theorem 5.3.11).

2.3 Audio Editing Tasks

Beyond audio generation, audio editing constitutes an important subdomain of generative audio synthesis research. It focuses on modifying existing audio content while preserving perceptual naturalness, structural coherence, and contextual consistency. In this section, we provide a comprehensive review of generative audio editing methods across sound, speech, and music, highlighting the techniques and tasks that enable controlled and high-fidelity audio editing.

Typical audio editing subtasks include conversion, inpainting, outpainting, and style transfer [Wang et al., 2023c; Zhang et al., 2024b]. Conversion modifies specific attributes of an audio signal, such as speaker identity, instrument timbre, and background, while preserving its structure and semantic content. Inpainting restores or reconstructs missing segments within an audio clip, whereas outpainting extends the audio beyond its original boundaries in a contextually coherent manner. Style transfer alters the expressive or stylistic characteristics of audio (e.g., emotion, genre, performance style) without changing its underlying structure or meaning. These operations require models to selectively alter certain acoustic properties while maintaining overall fidelity. Early studies, such as Ebner and Eltelt [2020], applied GAN-based architectures to perform targeted editing tasks, such as audio inpainting. AUDIT [Wang et al., 2023c] later introduced a general environmental sound editing framework that performs text-guided inpainting and outpainting from natural language instructions. Despite these developments, much of the research in this domain has focused on speech and music editing, reflecting their practical importance in applications such as dubbing, personalized voice creation, and music production [Mokrỳ and Rajmic, 2020; Borsos et al., 2022].

2.3.1 Voice Conversion

Within the speech domain, voice conversion (VC) is one of the most representative audio-editing tasks. VC aims to modify the vocal characteristics of a source speech signal to match a target speaker's identity while preserving its linguistic content. Traditional VC systems depend on parallel training data—paired utterances from different speakers—to learn frame-level mappings between source and target voices [Toda et al., 2007; Desai et al., 2009; Sun et al., 2015; Zhang et al., 2019]. However, this requirement severely limits scalability and adaptability to unseen speakers. To overcome these constraints, recent work has focused on non-parallel any-to-any (A2A) or one-shot VC frameworks [Qian et al., 2019], which disentangle speaker identity from linguistic content to enable flexible cross-speaker conversion using only a short target voice sample. Various modeling strategies have been explored, including instance normalization [Park et al., 2023], vector quantization [Wu and Lee, 2020; Wu et al., 2020; Wang et al., 2021], transfer learning from ASR or TTS models [Zhang et al., 2023; Li et al., 2023b; Hussain et al., 2023; Rekimoto, 2023], and adversarial training [Li et al., 2023a; Li and Wei, 2022]. With the rise of large language models (LLMs), natural language has emerged as an intuitive control interface for cross-modal generation. Motivated by this, recent research has introduced text-guided voice conversion, where vocal style is specified using natural-language prompts rather than explicit audio references. PromptVoice [Gölge and Davis, 2022] first demonstrated the feasibility of transferring vocal style through text descriptions. Text-GuidedVC [Kuan et al., 2023] extended this idea, conditioning voice conversion on linguistic attributes such as tone, accent, or emotional style (e.g., "a calm female voice with a British accent"). Similarly, PromptVC [Yao et al., 2023] employed a latent diffusion model (LDM) [Rombach et al., 2022a] within a VITS-based [Kim et al., 2021] framework to generate speaker-style embeddings from text. While text-based VC improves accessibility and diversity, it lacks the fine-grained acoustic precision achievable with audio-based prompts. To address this, we propose HybridVC in chapter 6, which is a flexible VC framework that jointly leverages audio and text prompts. This dual-prompt design combines the semantic interpretability of natural language with the acoustic precision of voice-based conditioning, supporting diverse real-world applications such as personalized voice creation for avatars, social media, and interactive communication.

2.3.2 Music Editing

In the music domain, music editing aims to modify existing musical recordings to satisfy specific conditions or stylistic goals while preserving the original musical structure. Earlier approaches trained dedicated editing models from scratch [Copet et al., 2023; Wang et al., 2023c; Mariani et al., 2023] or fine-tuned pre-trained generative models for specific tasks [Zhang et al., 2024b; Tsai et al., 2024; Han et al., 2023; Liu et al., 2023b; Rouard et al., 2025]. Recent research has focused on zero-shot text-guided music editing [Zhang et al., 2024c; Manor and Michaeli, 2024a; Liu et al., 2024b], which relies on large pre-trained audio generation models for edit-

ing tasks such as timbre transfer, instrument replacement, and style transformation based solely on textual instructions. This paradigm allows flexible and semantically meaningful editing without retraining or dataset-specific adaptation. A key subtask in this area is timbre transfer, which involves converting a musical piece performed by one instrument (source) into the same composition played by another instrument (target) while preserving melody, rhythm, and other musical attributes [Comanducci et al., 2023; Jain et al., 2020; Li et al., 2024b]. Timbre transfer parallels voice conversion in speech, as both aim to retain content while transforming style. However, instrument timbre is influenced not only by instrument type but also by performer technique and instrument quality, making this problem highly complex. Recent advances such as DreamSound [Plitsis et al., 2024] extend text-guided generation to personalized music editing, where models learn style concepts from user-provided reference tracks to achieve fine-grained and individualized editing. This approach enables more expressive and user-aligned transformations than generic text-guided methods. Nonetheless, current zero-shot editing pipelines often struggle to maintain musical consistency when adapting to new style concepts. To address this, we propose SteerMusic in chapter 8, a framework that enhances source consistency during text-guided music style editing, achieving both coarse-grained and fine-grained control.

In short, generative audio editing research has evolved from rule-based and supervised transformation systems toward multimodal, prompt-driven editing frameworks capable of complex style control and content preservation. These developments, spanning speech, music, and environmental sound, represent a natural extension of the generative audio paradigm, unifying generation and manipulation within a single modeling framework.

2.4 Sound Morphing

Sound morphing is the task of smoothly transforming one sound into another by continuously interpolating its perceptual or acoustic characteristics. The goal is to generate a sequence of intermediate sounds that perceptually transition from a source sound to a target sound in a natural and coherent manner. Sound morphing has broad applications in music production, sound design, and auditory scene synthesis, where smooth and controllable transitions between sound textures or instruments are desired.

Traditional sound morphing methods are based on interpolating parameters of a sinusoidal sound synthesis model [Tellman et al., 1995; Osaka, 1995; Williams et al., 2014; Primavera et al., 2012]. To achieve more effective and continuous morphing, [Williams et al., 2014; Brookes and Williams, 2010; Caetano and Rodet, 2010, 2011; Roma et al., 2020; Caetano, 2011] explore perceptual spectral domain audio features by digital signal processing techniques, such as MFCCs, spectral envelope, etc. A hybrid approach has been explored that extracts audio descriptors to morph accordingly and interpolates between the spectrotemporal fine structures of two endpoints

according to morph factors [Kazazis et al., 2016].

Machine learning sound morphing methods offer advantages such as high morphing quality by interpolation of semantic representations within a model instead of using traditional audio features. Zou et al. [2021] proposes a non-parallel many-to-one static timbre morphing framework that integrates and fine-tunes the machine learning technique (i.e., DDSP-autoencoder [Engel et al., 2020]) with spectral feature interpolation [Caetano and Rodet, 2013]. Kim et al. [2019b] synthesizes music corresponding to a note sequence and timbre, using nonlinear instrument embeddings as timbre control parameters under a pre-trained WaveNet [Engel et al., 2017] to achieve timbre morphing between instruments. Luo et al. [2019] learns latent distributions of VAEs to disentangle representations of the pitch and timbre of musical instrument sounds. Tan et al. [2020] uses a GM-VAE to achieve style morphing to generate realistic piano performance in the audio domain following temporal style conditions for piano performance, which morphs the conditions such as onset roll and MIDI note into input audio. MorphGAN [Gupta et al., 2023] targets audio texture morphing by interpolating within conditional parameters training the model on a water-wind texture dataset. Sheng et al. [2024] introduce a voice morphing method, which focuses on blending or transforming one speaker's voice into another. This often involves creating an intermediate voice that incorporates characteristics of both source and target voices, allowing for gradual transitions between them. A recent work, MorphFader [Kamath et al., 2025], uses a pre-trained AudioLDM [Liu et al., 2023a] to morph sound between two text prompts. In chapter 7, we introduce a flexible sound morphing solution, which relies on a large pre-trained audio generation model to perform diverse sound morphing tasks, including timbre morphing, music morphing, and environmental sound morphing. In contrast to MorphFader, we focus on classical sound morphing, where the morphing process is performed directly between two given sounds rather than between text prompts. A key advantage of our method is its ability to provide precise guidance during the morphing process, since the target audio delivers exact information on how the source sound should evolve, something that text prompts cannot always achieve, for example, morphing between two music compositions.

A related research direction is *synthesizer preset interpolation*, which seeks to achieve sound morphing by interpolating between parameter configurations of non-differentiable synthesizers [Le Vaillant and Dutoit, 2023a,b, 2024]. Instead of directly manipulating audio waveforms, these models treat the synthesizer as a black box and perform interpolation in its parameter space, which may include both continuous and categorical controls. While synthesizer preset interpolation can yield perceptually smooth transitions between timbres, it fundamentally differs from classical sound morphing in that it operates on symbolic control parameters rather than the acoustic domain itself.

Figure 2.1: Visualization of an audio waveform.

2.5 Audio Representations in Generative Audio Synthesis

Audio representation defines how sound is expressed numerically for analysis, modeling, and synthesis. The choice of representation significantly affects the learning efficiency, perceptual quality, and controllability of generative models. In generative audio synthesis, representations can be broadly grouped into four types: time-domain waveforms, time–frequency representations, learned discrete audio tokens, and self-supervised learning audio features.

2.5.1 Waveform Representation

The most fundamental representation of audio is the waveform, a time-domain signal that encode instantaneous air-pressure fluctuations as a sequence of amplitude samples. As illustrated in fig. 2.1, a waveform depicts the evolution of a sound's amplitude over time. Waveforms preserve complete temporal and phase information, offering maximal fidelity and enabling precise modeling and manipulation of fine-grained audio structures, which is critical for tasks in generative audio synthesis and signal processing. However, raw waveform modeling poses substantial challenges. Typical sampling rates (e.g., 16 kHz–48 kHz) produce tens of thousands of samples per second, resulting in huge long-range temporal dependencies (e.g., sequence length $\approx$ sampling rate $\times$ duration in seconds) that significantly increase computational complexity. While autoregressive models such as WaveNet [van den Oord et al., 2016] demonstrated that high-fidelity waveform generation is achievable, they often suffer from slow inference and limited capacity to capture long-term context. As a result, subsequent research has focused on more compact or perceptually meaningful representations that reduce sequence length while preserving essential acoustic information, facilitating efficient and scalable generative modeling.

2.5.2 Spectrogram Representations

To make learning more efficient, many audio generative models [Yang et al., 2023c; Ren et al., 2019; Hou et al., 2025] transform waveforms into time–frequency domain

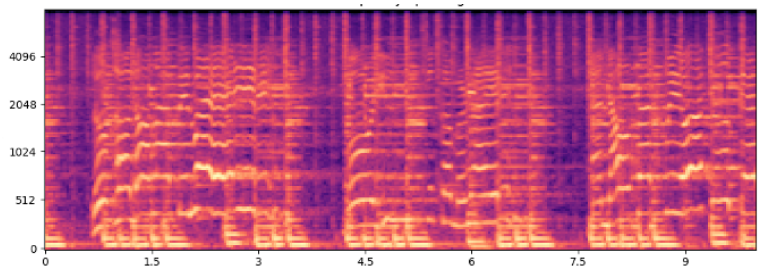


Figure 2.2: Visualization of a mel-spectrogram of an audio segment.

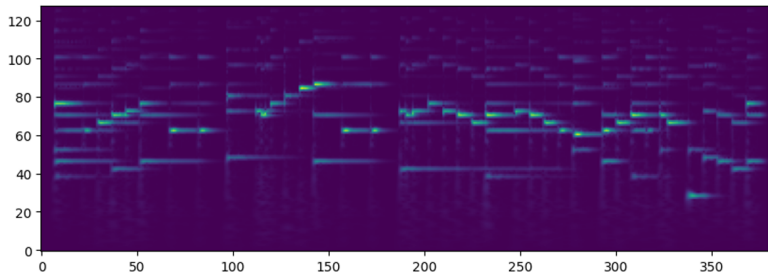


Figure 2.3: Visualization of a CQT-spectrogram of a piano music segment.

audio representations. These representations are typically in the form of spectrograms, which encode the magnitude of frequency components over time. Spectrograms can be viewed as pseudo-3D representations, visualizing how the intensity of different frequency components evolves over time.

A widely used variant is the mel-spectrogram (fig. 2.2), which applies a mel scale filter bank [Kubichek, 1993] to the magnitude spectrum produced by the Short-Time Fourier Transform (STFT), warping the frequency axis according to perceptual frequency spacing. The mel scale warps the frequency axis according to human auditory perception, offering finer resolution at lower frequencies (i.e., where the human ear is more sensitive) and coarser resolution at higher frequencies. This produces a compact, perceptually informed feature map that balances information richness with computational efficiency, making mel-spectrograms a standard choice in speech and sound generation tasks.

Another commonly used time–frequency representation is the Constant-Q Transform (CQT) spectrogram (fig. 2.3), which maps a signal onto a logarithmically spaced frequency axis using the constant-Q transform [Brown, 1991]. The CQT provides higher frequency resolution at lower frequencies and coarser resolution at higher frequencies, which makes CQT spectrograms particularly well-suited for music-related tasks, such as pitch analysis, transcription, and music generation, where fine resolution at low frequencies is crucial. Like mel-spectrograms, CQT spectrograms capture the temporal evolution of spectral content but with a frequency representation tailored to musical signals.

Spectrogram-based audio generation methods typically predict spectrogram rep-

resentations as intermediate outputs, which are later converted back into waveforms using a neural vocoder (see section 2.6). This separation simplifies the generation task by letting models focus on spectral patterns rather than raw waveform samples.

2.5.3 Learned Discrete Audio Tokens

Recent advances have introduced learned audio representations through vector quantization (VQ) and neural codecs such as VQ-VAE [van den Oord et al., 2017a], SoundStream [Zeghidour et al., 2021], and EnCodec [van den Oord et al., 2017b]. These methods compress continuous waveforms into lower dimensional sequences of discrete audio tokens or latent code, but preserve the information in the original waveform's acoustic features. This tokenization converts audio generation into a discrete sequence modeling problem, allowing researchers to apply techniques from natural language processing, such as transformer-based language models or discrete diffusion models, operating in token space. Discrete representations offer key advantages, as their lower temporal resolution reduces sequence length and model complexity, making them more suitable for multi-modal fusion. These representations have become central to recent multi-modal audio generation frameworks, forming the basis of text-to-audio, video-to-audio, and music-generation systems.

2.5.4 Self-Supervised Learning (SSL) Features

SSL features are extracted from self-supervised learning models, such as wav2vec 2.0 [Baevski et al., 2020], HuBERT [Hsu et al., 2021], and WavLM [Chen et al., 2022b], which learn continuous feature embeddings directly from unlabeled audio. These features capture rich semantic and perceptual information, such as phonetic information, timbre, and event type, without explicit supervision. In generative audio synthesis tasks, SSL features are often applied as conditioning inputs or perceptual representations, enabling generative models to encode audio information more efficiently and produce more semantically coherent outputs.

2.6 Neural Vocoders

A neural vocoder (voice encoder–decoder) is a machine learning-based model that reconstructs or synthesizes a time-domain waveform from intermediate low-dimensional audio representations, such as mel-spectrograms. Neural vocoders, such as HiFiGAN [Kong et al., 2020a] and MelGAN [Kumar et al., 2019], play a critical role in modern generative audio methods, acting as the final stage that converts generated intermediate audio representations into perceptually realistic waveforms. They bridge the gap between high-level generative modeling (e.g., text-to-mel or latent diffusion) and low-level waveform synthesis, determining much of the final perceptual quality. Traditional signal-processing vocoders (e.g., the Griffin–Lim algorithm [Griffin and Lim, 1984]) are limited by phase reconstruction artifacts and

low naturalness. Neural vocoders address these limitations by learning an explicit mapping from acoustic features to waveform samples using data-driven neural networks. Given a conditioning feature sequence c (e.g., mel-spectrogram), a neural vocoder learns a function $f_\theta(c, z) \rightarrow x$, where z is a latent variable and x is the synthesized audio waveform. By learning this mapping directly from paired data, neural vocoders can produce high-fidelity, phase-accurate audio that closely resembles natural recordings.

2.7 Evaluation Metrics in Generative Audio Synthesis

Evaluating generated audio is essential, and early research relies on subjective listening tests to assess perceptual quality. However, such evaluations are costly, time-consuming, and susceptible to human bias. To mitigate these limitations, recent works have increasingly adopted objective evaluation metrics as reproducible alternatives. In this thesis, we exclusively employ objective evaluation methods, focusing on quantifiable measures that enable consistent comparisons across models and tasks. This section introduces all objective metrics used throughout the thesis, which are commonly used in the generative audio synthesis domain. These objective metrics cover multiple evaluation dimensions, including audio quality, sampling diversity, audio similarity, speech intelligibility, structural contour correlation, perceptual similarity, temporal alignment, and inference efficiency.

2.7.1 Audio quality

Audio quality is a fundamental evaluation dimension in generative audio synthesis tasks, assessing how natural, realistic, and perceptually consistent the synthesized audio is when compared with real-world data.

Fréchet Audio Distance (FAD) is a distribution-based audio quality metric, which quantifies how close the distribution of generated audio features is to that of real audio samples. Given embeddings of real audio samples $\{f_r^i\}_{i=1}^{N_r}$ and generated audio samples $\{f_g^i\}_{i=1}^{N_g}$ extracted from a pre-trained audio representation model such as VGGish, PANNs, or CLAP [Chen et al., 2020a; Kong et al., 2020b; Wu et al., 2023a]. The FAD score is computed by

$$\text{FAD} = ||\mu_r - \mu_g||_2^2 + Tr(\Sigma_r + \Sigma_g - 2(\Sigma_r\Sigma_g)^{1/2}). \tag{2.11}$$

where μ represents the mean and Σ represents the covariances.

Non-Intrusive Speech Quality Assessment (NISQA) is a learned perceptual speech quality metric using machine learning [Mittag et al., 2021]. NISQA predicts human mean opinion scores (MOS) for speech quality without requiring a reference. Let x be an input speech signal and $f_\theta(x)$ denote the output neural model trained to approximate human perceptual quality ratings.

$$\text{NISQA}(x) = f_\theta(x) \in [1, 5]. \tag{2.12}$$

where the score represents a predicted MOS value with 1 = poor quality and 5 = excellent quality.

2.7.2 Sampling diversity

Sampling diversity reflects the variability of generated outputs. For example, the generated environmental sound should not only match the condition but also remain diverse, as real-world sounds are inherently varied and unpredictable. Diversity ensures realism, avoids mode collapse, and enhances the model's creative and generative capacity.

Kullback–Leibler (KL) Divergence measures the distributional discrepancy between the real data distribution p_{data} and the generated data distribution p_θ. The KL score can be computed by

$$\mathrm{KL} = \int p_{data}(x) log \frac{p_{data}(x)}{p_\theta(x)} dx \tag{2.13}$$

In practice, both $p_{data}(x)$ and $p_\theta(x)$ are approximated by feature embeddings extracted from real and generated audio samples. A small KL score implies that the generated audio samples are statistically similar and diverse relative to the reference distribution.

Kernel Inception Distance (KID) [Bińkowski et al., 2018] measures the squared Maximum Mean Discrepancy (MMD) between the embeddings of real and generated audio features that extracted from a pre-trained classifier (e.g., PANNs [Kong et al., 2020b] used in this thesis). The KID is formulated as

$$\mathrm{KID} = \mathbb{E}[k(x, x')] + \mathbb{E}[k(y, y')] - 2\mathbb{E}[k(x, y)] \tag{2.14}$$

Here, x and x' are samples from the real distribution, y and y' are samples from the generated distribution, and $k(a, b) = (a^T b/d + 1)^3$ is the kernel function evaluating the similarity between samples a and b, where d is the dimensionality of the feature vectors. Lower KID values indicate closer alignment between real and generated distributions (i.e., diverse and realistic outputs).

Inception Score (ISc) evaluates the semantic clarity and diversity of generated audio by analyzing how confidently and distinctly a pre-trained classifier (e.g., PANNs [Kong et al., 2020b] used in this thesis) can categorize them. It assumes that high-quality samples yield confident predictions (low entropy per sample), while a diverse set of samples spans many classes (high entropy overall). Let $p(y|x)$ be the conditional label distribution predicted by a pre-trained classifier, and $p(y) = \int p(y|x)p(x)dx$ is the marginal distribution across all samples. The ISc is defined as

$$\mathrm{ISc} = exp(\mathbb{E}_{x\sim p_\theta}[D_{\mathrm{KL}}(p(y|x)||p(y))]) \tag{2.15}$$

A higher ISc indicates that the generated audio is both realistic and diverse.

2.7.3 Audio Similarity

Measuring audio similarity directly evaluates how closely a generated audio resembles a reference signal by comparing spectral or perceptual features. This provides an objective way to assess reconstruction quality and audio content preservation.

mel-cepstral Distortion (MCD) measures spectral distortion between two utterances in the mel-cepstral domain, which is widely used in the speech synthesis domain. Given the reference and generated speech frames with mel-cepstral coefficients (MCC)

$$c = [c_1, c_2, ..., c_D] \text{ and } \hat{c} = [\hat{c}_1, \hat{c}_2, ..., \hat{c}_D], \tag{2.16}$$

for a D-dimentional MCC vector, the MCD is defined as

$$\text{MCD[dB]} = \frac{10}{ln10}\sqrt{2\sum_{d=1}^{D}(c_d - \hat{c}_d)^2}. \tag{2.17}$$

When comparing utterances consisting of multiple frames, the final score is usually averaged across all frames.

Structural Similarity Index (SSIM). measures the structural consistency between a pair of audio spectrograms $(x, \hat{x})$, which is usually applied to the speech and music domain. SSIM can be calculated by

$$\text{SSIM}(x, \hat{x}) = \frac{(2\mu_x\mu_{\hat{x}} + C_1)(2\Sigma_{x\hat{x}} + C_2)}{(\mu_x^2 + \mu_{\hat{x}}^2 + C_1)(\sigma_x^2 + \sigma_{\hat{x}}^2 + C_2)}. \tag{2.18}$$

where μ represents intensity values, $\sigma_x, \sigma_{\hat{x}}$ are variances, $\Sigma_{x\hat{x}}$ is the covariance between x and $\hat{x}$, C_1, C_2 are constants for stability.

2.7.4 Speech Intelligibility

Speech intelligibility metrics directly assess how well the linguistic content of generated speech matches the reference. These measures quantify recognition accuracy or transcription fidelity, providing an objective way to evaluate whether speech remains clear, understandable, and semantically correct after synthesis or transformation.

Word Error Rate (WER) measures the difference between a predicted transcript and the ground-truth transcript at the word level, serving as a proxy for speech intelligibility. It quantifies how accurately the spoken content can be transcribed. Given a reference transcript $R = [r_1, r_2, ..., r_N]$ and a hypothesis transcript $H = [h_1, h_2, ..., h_M]$ which may be transcribed from a synthesized speech. The WER is defined as

$$\text{WER} = \frac{S + D + I}{N} \tag{2.19}$$

where S is the number of incorrect words, D is the number of missing words, I is the number of insertions, and N is the total number of words in the reference transcript. Lower WER indicates higher speech intelligibility.

Character Error Rate (CER) is similar to WER in Equation 2.19, but computed at the character level rather than the word level. CER provides a finer-grained measure of intelligibility than WER and is more sensitive to minor spelling variances.

2.7.5 Structure Contour Correlation

Structure contour correlation metrics are commonly used to assess prosodic or melodic similarity between a generated (or edited) audio signal and its source. These metrics, such as F0-PCC for pitch contour alignment and CQT-1 PCC for harmonic or melodic structure consistency, quantify how well the underlying temporal or musical contours are preserved during editing. Higher correlation values indicate that the model maintains the essential expressive patterns of the original audio.

Fundamental Frequency Pearson Correlation Coefficient (F0 PCC) measures the correlation of pitch contours between two audio clips, which is widely used in the voice conversion task. Given a source speech x^{src} and the converted speech x^{tgt}, the detailed F0 PCC metric can be formulated as

$$\text{F0 PCC} = \frac{\sum_i^T (f_i^{src} - \bar{f}^{src})(c_i^{tgt} - \bar{f}^{tgt})}{\sqrt{\sum_i^T (f_i^{src} - \bar{f}^{src})^2 \sum_i^T (f_i^{tgt} - \bar{f}^{tgt})^2}} \tag{2.20}$$

where f_i represents the F0 value at frame i. F0 PCC score should in the range of $[-1, 1]$, where $+1$ means perfect correlation, 0 means no correlation, and -1 means inverse correlation. Higher F0 PCC indicates better preservation of pitch contours during voice conversion.

Top1 Constant-Q Transform Pearson Correlation Coefficient (CQT-1 PCC) is introduced in Chapter 8, which measures the melody similarity of two music compositions in the music editing tasks by calculating the Pearson correlation coefficient between top-1 CQT bin features. Given source music x^{src} and the edited music x^{tgt}, the detailed CQT-1 PCC metric can be formulated as

$$\text{CQT-1 PCC} = \frac{\sum_i^T (c_i^{src} - \bar{c}^{src})(c_i^{tgt} - \bar{c}^{tgt})}{\sqrt{\sum_i^T (c_i^{src} - \bar{c}^{src})^2 \sum_i^T (c_i^{tgt} - \bar{c}^{tgt})^2}} \tag{2.21}$$

where c_i is the ith index of CQT-1 value. CQT-1 PCC score should be in the range of $[-1, 1]$, where $+1$ means perfect correlation, 0 means no correlation, and -1 means inverse correlation. Higher CQT-1 PCC indicates better preservation of the main music melody during editing.

2.7.6 Perceptual similarity

Perceptual similarity metrics evaluate how closely a generated audio signal aligns with human auditory perception rather than raw signal differences. Metrics such as LPAPS and CDPAM leverage deep neural network embeddings trained on large-scale audio data to capture high-level perceptual attributes. These measures provide a

more human-aligned assessment of audio quality, making them particularly suitable for tasks where signal-level metrics fail to reflect perceived similarity.

Learned Perceptual Audio Patch Similarity (LPAPS) is designed to evaluate perceptual similarity between two audio clips by comparing their learned feature representations from deep neural networks trained for audio classification tasks. It generalizes the concept of LPIPS (Learned Perceptual Image Patch Similarity) [Zhang et al., 2018a] from the vision domain to audio. Given an audio pair (x_1, x_2), LPAPS computes their feature embeddings at multiple layers l of a pre-trained audio representation networks such as VGGish, PANNs and CLAP. The LPAPS score is the weighted sum over Euclidean distance of feature maps between x_1 and x_2 as

$$\text{LPAPS}(x_1, x_2) = \sum_l w_l d_l(x_1, x_2), \tag{2.22}$$

where $d_l(x_1, x_2)$ is the normalized L2 distance of feature maps between x_1 and x_2. Lower LPAPS indicates better audio perceptual similarity.

Contrastive Learning for Perceptual Audio Similarity (CDPAM) explicitly optimizing embeddings for perceptual alignment by training an audio embedding model with a contrastive learning strategy, directly aligned with human perceptual judgments. During inference, the CDPAM similarity between two audio signals (x_1, x_2) is simply the Euclidean distance between their embeddings, as

$$\text{CDPAM}(x_1, x_2) = ||f_\theta(x_1) - f_\theta(x_2)||_2 \tag{2.23}$$

where f_θ is a pre-trained model. Lower CDPAM indicates better perceptual similarity.

2.7.7 Alignments between audio and other modalities

In multi-modal audio generation tasks, it is important to verify how well the synthesized audio content aligns with the given condition. The better alignment indicates the model has better controllability and higher generation quality.

Contrastive Language-Audio Pretraining (CLAP) Score measures the cosine similarity using a pre-trained CLAP model [Wu et al., 2023a], which reflects a semantic level of alignment between the audio and text prompts. Given two CLAP features f_a and f_t, where f_a represents the CLAP audio features and f_t represents the CLAP text embedding, the CLAP score is calculated by the cosine similarity formulation between the two CLAP features.

$$\text{CLAP} = \frac{f_a \cdot f_t}{||f_a|| \cdot ||f_t||} \tag{2.24}$$

The higher CLAP score indicates better alignment.

DeSync quantitatively measures the temporal misalignment between video and audio streams predicted by a pre-trained synchronization network [Iashin et al., 2024], which is commonly used in multimodal generative tasks such as video-to-audio or video-to-speech synthesis. Lower DeSync score represents better audio-visual syn-

chronization while higher DeSync score represents temporal misalignment.

2.7.8 Inference speed

Inference speed is an important factor in generative tasks, which measure the generation efficiency of a model.
Real-Time Factor (RTF) is a commonly used metric to evaluate the inference speed of a model, especially in the speech and audio processing domain. The RTF is calculated by

$$\text{RTF} = \frac{\text{Inference Time}}{\text{Input Duration}} \tag{2.25}$$

Low RTF is better for real-time systems.

2.8 Summary

This chapter has presented a comprehensive review of the literature and the technical foundations that underpin the research conducted in this thesis. Building upon this background, the subsequent chapters address the core research questions outlined in section 1.3.

We begin by investigating generative AI methods for sound synthesis. In chapter 3, we focus on improving the efficiency and controllability of text-to-sound generation models, while in chapter 4, we enhance controllability of speech content in video-to-audio generation.

We then shift our attention to speech generation tasks, which are orthogonal to general sound effect generation. In chapter 5, we propose a unified generative framework and showcase its applicability on text-to-speech and singing voice synthesis tasks, which removes the need for external pretrained components on existing two-step models and makes the model trainable by an end-to-end pipeline. In chapter 6, we introduce a flexible and robust solution for voice conversion.

Finally, we extend our exploration to generative AI for creative content and music applications. In chapter 7, we present a flexible method for sound morphing, and in chapter 8, we develop an enhanced and controllable approach to music editing.

In summary, these contributions systematically address efficiency, controllability, and flexibility across multiple generative audio domains, forming a coherent response to the central research questions of this thesis.

Chapter 3

SoundLoCD: An Efficient Conditional Discrete Contrastive Latent Diffusion Model for Text-to-Sound Generation

Sounds carry rich information about environments and physical events, which shape how we perceive and interact with the world. In real-world applications such as film production, virtual reality, and game development, generating realistic and controllable sound effects is essential to improve immersive user experiences [Kern and Ellermeier, 2020; Bosman et al., 2024; Byun and Loh, 2015; Yang et al., 2023b].

In this chapter, we introduce SoundLoCD, a novel text-to-sound generation framework, which incorporates a low-rank adaptation (LoRA) based conditional discrete contrastive latent diffusion model. Unlike recent large-scale sound generation models, which require extensive computational resources, our method can be efficiently trained under limited computational resources. To enhance the alignment between textual conditions and synthesized sounds, SoundLoCD integrates a contrastive learning strategy that further enhances the connection between text conditions and the generated output, resulting in coherent and high-fidelity performance. Through comprehensive experiments, we demonstrate that SoundLoCD achieves high-fidelity text-to-sound generation while significantly reducing training cost. Compared to the baseline approach, SoundLoCD provides higher controllability with stronger condition–output correspondences and improved training efficiency, making it a practical solution for real-world applications where computational resources are limited[1].

3.1 Introduction

Sound synthesis spans a diverse array of applications, including game development, film post-production, virtual reality, and musical performance. In these domains, automated sound effects generation could address tasks such as replicating environ-

[1]Demonstration page: https://XinleiNIU.github.io/demo-SoundLoCD/

mental ambience and simulating a specific physical event where the generated sound is required to be aligned with predefined scene descriptions. The challenge of synthesizing environmental sound effects based on a natural language text description is then referred to as *text-to-sound* (T2S) generation.

The pioneering T2S work is DiffSound [Yang et al., 2023b], which introduced the concept of using natural language text descriptions as conditions for generating acoustic scenes. Diffsound follows a conditional discrete diffusion pipeline [Gu et al., 2022a], which involves a pre-trained spectrogram VQ-VAE, the text encoder of a contrastive language-image pretraining model [Radford et al., 2021], and a discrete latent diffusion model (LDM). Following Yang et al. [2023b], AudioGen Kreuk et al. [2023] proposed a GAN-based auto-regressive model for audio generation under a neural audio codec technique [Zeghidour et al., 2021; Li et al., 2021], where they used the text encoder of a text-to-text transformer Raffel et al. [2020]. However, the higher inference time of AudioGen makes it unsuitable for modeling long sequences. Some subsequent studies [Liu et al., 2023a; Ghosal et al., 2023; Yuan et al., 2023] employed large LDMs [Gu et al., 2022a; Song et al., 2021b; Ho et al., 2020b] to general audio generation with prompt in the continuous latent space of a VAE. Among them, AudioLDM [Liu et al., 2023a] worked on text-free audio generation. Yuan et al. [2023]; Ghosal et al. [2023] focused on tuning a powerful text encoder based on AudioLDM. These prompt models require large-scale data for training. Meanwhile, Huang et al. [2023e,a] proposed the generalization for multi-modal audio generation. In contrast to those works, we follow DiffSound and focus on a text-to-sound effects LDM model under a discrete latent space of VQ-VAE van den Oord et al. [2017a]. As language is discrete, it is arguably more suitable to integrate textual features into a discrete latent space, facilitated by an LDM.

We observe two main limitations of Diffsound [Yang et al., 2023b]. Firstly, there exist no explicit constraints to achieve effective controllable generation given text conditions. Secondly, the significant requirement for computational resources (see table 3.1) poses challenges for other downstream tasks. For the former, most existing conditional LDM models [Yang et al., 2023b; Liu et al., 2023a; Ghosal et al., 2023; Huang et al., 2023e,a] learn the connection between conditions and outputs by the conditional prior on the variational lower bound Gu et al. [2022a]; Dhariwal and Nichol [2021]. We note that this conditional prior does not invariably guarantee a strong linkage between outputs and conditions throughout each diffusion process. This may critically affect the accuracy of generated sounds. For the latter, we claim that an efficient T2S model is more desirable to scale to other related fields.

In this chapter, we propose a novel T2S framework, SoundLoCD, that achieves effective training on small computational devices while further improving the synthesis quality by enhancing the connection between text conditions and synthesized sounds. SoundLoCD is a conditional discrete contrastive diffusion model [Zhu et al., 2022] that injects a smaller number of trainable parameters on a pre-trained DiffSound [Yang et al., 2023b] transformer by LoRA [Hu et al., 2022]. We make the following contributions:

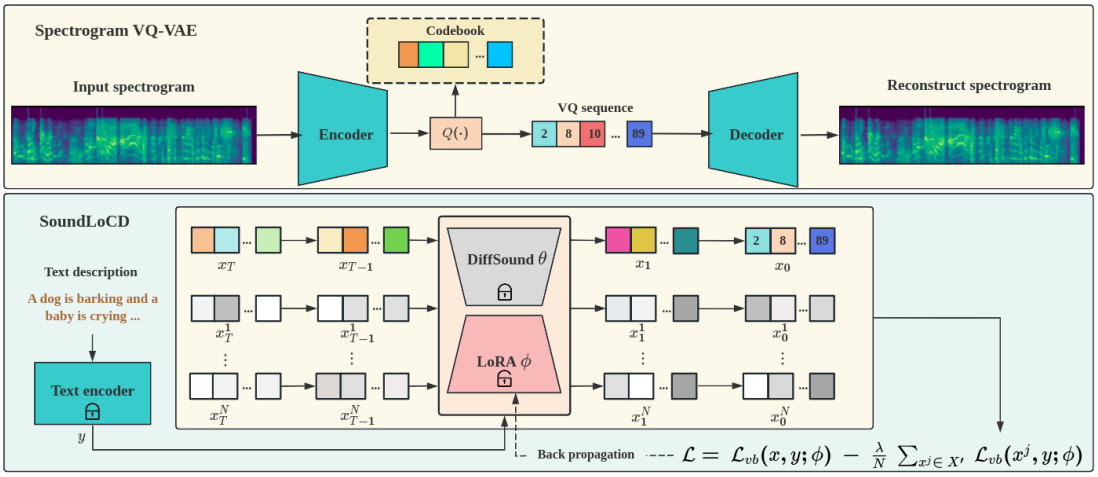


Figure 3.1: Overview pipeline of SoundLoCD, which is performed based on a pre-trained spectrogram VQ-VAE. *SoundLoCD involves $N + 1$ parallel discrete diffusion processes on original data and N randomly shuffled negative data.*

1. We proposed SoundLoCD, which obtains a better performance in generating sound effects that closely correspond to the given text conditions.

2. Our method markedly enhances training efficiency under limited computational resources.

3.2 Method

In this section, we introduce the detailed methodology of SoundLoCD. The overview pipeline of SoundLoCD is in fig. 3.1, where it learns the latent distribution of the spectrogram VQ-VAE given text conditions with a *contrastive latent diffusion model*. Specifically, given a text-spectrogram pair, we first obtain the VQ sequence $x_0 \in \mathbb{Z}^D$ of the spectrogram with a pre-trained VQ-VAE (same as the VQ-VAE used in Yang et al. [2023b]). The encoder encodes the spectrogram into a VQ sequence x_0 during training, where D is the length of x_0. The i^{th} token in x_0 takes the index that specifies the VQ codebook entries with size K. The decoder is used to reconstruct the generated VQ sequence x_0 of SoundLoCD back to a spectrogram during inference. Meanwhile, text descriptions are encoded by a pre-trained text encoder to extract the text feature representation defined as $y \in \mathbb{Z}^M$.

3.2.1 SoundLoCD

SoundLoCD is a LoRA-based *conditional discrete contrastive diffusion* [Zhu et al., 2022] (CDCD) model, which contains two sets of parameters $\{\theta, \phi\}$. θ is the set of parameters in DiffSound transformer, which is frozen during training, and ϕ is the set of LoRA parameters to be learned. As a diffusion model, the forward process

corrupts x_0 to pure noise x_T by fixed T time steps, and the reverse process gradually denoises the latent variable to x_0 by sampling from $q(x_{t-1}|x_t, x_0)$ Song et al. [2021a]. Thus, SoundLoCD is trained to approximate the conditional transit distribution $p_{\{\theta,\phi\}}(x_{t-1}|x_t, y)$.

In order to further enhance the connection between y and the generated VQ sequence x_0, SoundLoCD involves a contrastive learning strategy and performs a *parallel diffusion process* on a set of negative data $X' = \{x^1, x^2, ..., x^N\}$, which contains N randomly shuffled VQ sequences given x_0. As shown in Zhu et al. [2022], the idea of involving X' is to pull the well-aligned conditions and the VQ sequence closer, while pushing other pairs apart. Thus, with contrastive learning, SoundLoCD aims to produce sound effects that correspond accurately with the condition of text description feature y.

3.2.2 LoRA

We notice that the performance of existing models relies on huge computational resources, making it hard to further edit the models. In SoundLoCD, LoRA [Hu et al., 2022] is applied to adapt a pre-trained DiffSound transformer by injecting a smaller number of trainable parameters ϕ into the transformer, leading to an efficient training system. Given a pre-trained DiffSound transformer with a weight matrix $W_0 \in \mathbb{R}^{d\times k}$, LoRA constrains its update by representing the latter with a *low-rank decomposition* $W_0 + \Delta W = W_0 + BA$ where $B \in \mathbb{R}^{d\times r}$, $A \in \mathbb{R}^{r\times k}$, and the rank $r \ll min(d, k)$.

Thus the forward pass of SoundLoCD becomes:

$$h = W_0 x + \Delta W x = W_0 x + BAx, \tag{3.1}$$

where $\{A, B\} \in \phi$. At the beginning of training, A, B are initialized by a Gaussian distribution and zeros respectively. Then $\Delta W x$ is scaled by $\frac{\alpha}{r}$, α is a constant and r is the rank.

3.2.3 Conditional Discrete Contrastive Diffusion

SoundLoCD involves $N + 1$ parallel discrete diffusion processes for each of the samples in the original data x_0 and X'.

Discrete diffusion process. To enhance clarity for readers, we present a detailed discrete diffusion process by excluding negative data $x^j \in X'$. The forward discrete diffusion process gradually corrupts the x_0 via a fixed Markov chain $q(x_t|x_{t-1})$ in a fixed number of timesteps T.

$$q(x_t|x_{t-1}) = \mathbf{v}^T(x_t)\mathbf{Q}_t\mathbf{v}(x_{t-1}), \tag{3.2}$$

where $\mathbf{v}(x)$ is a one-hot column vector whose length is K and only the entry x is 1. $\mathbf{Q}$ is a transition matrix $[\mathbf{Q}_t]_{mn} = q(x_t = m|x_{t-1} = n) \in \mathbb{R}^{K\times K}$. The categorical distribution over x_t is given by the vector $\mathbf{Q}_t\mathbf{v}(x_{t-1})$.

Since the Markov chain can marginalize out the intermediate steps and derive the probability of x_t at an arbitrary timestep directly from x_0 as follows:

$$q(x_t|x_0) = \mathbf{v}^T(x_t)\overline{\mathbf{Q}}_t\mathbf{v}(x_0), \text{ with } \overline{\mathbf{Q}}_t = \mathbf{Q}_t \cdots \mathbf{Q}_1 \tag{3.3}$$

The non-Markovian posterior $q(x_{t-1}|x_t, x_0)$ of the diffusion process can be computed according to eq. (3.2) and eq. (3.3), as

$$q(x_{t-1}|x_t, x_0) = \frac{(\mathbf{v}^T(x_t)\mathbf{Q}_t\mathbf{v}(x_{t-1}))(\mathbf{v}^T(x_{t-1})\overline{\mathbf{Q}}_{t-1}\mathbf{v}(x_0))}{\mathbf{v}^T(x_t)\overline{\mathbf{Q}}_t\mathbf{v}(x_0)}. \tag{3.4}$$

Mask-and-replace diffusion strategy. Following Gu et al. [2022a], the transition matrix $\mathbf{Q}_t \in \mathbb{R}^{(K+1)\times(K+1)}$ is defined as

$$\mathbf{Q}_t = \begin{bmatrix} \alpha_t + \beta_t & \beta_t & \beta_t & \cdots & 0 \\ \beta_t & \alpha_t + \beta_t & \beta_t & \cdots & 0 \\ \beta_t & \beta_t & \alpha_t + \beta_t & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \gamma_t & \gamma_t & \gamma_t & \cdots & 1 \end{bmatrix}, \tag{3.5}$$

where each ordinary token has a probability of γ_t to be a [MASK] token and has a chance of $K\beta_t$ to be uniformly diffused, leaving a probability of $\alpha_t = 1 - K\beta_t - \gamma_t$ to be unchanged. The [MASK] token always keeps its own state. Thus $q(x_t|x_0)$ can be calculated according to

$$\overline{\mathbf{Q}}_t\mathbf{v}(x_0) = \overline{\alpha}_t\mathbf{v}(x_0) + (\overline{\gamma}_t - \overline{\beta}_t)\mathbf{v}(K+1) + \overline{\beta}_t, \tag{3.6}$$

where $\overline{\alpha}_t = \prod_{i=1}^{t}\alpha_i, \overline{\gamma}_t = 1 - \prod_{i=1}^{t}(1-\gamma_i)$, and $\overline{\beta}_t = (1 - \overline{\alpha}_t - \overline{\gamma}_t)/K$. The prior $p(x_T)$ is defined as

$$p(x_T) = [\overline{\beta}_T, \overline{\beta}_T, \cdots, \overline{\beta}_T, \overline{\gamma}_T]^T. \tag{3.7}$$

Learning. To train SoundLoCD, DiffSound parameters θ are frozen and only LoRA parameters ϕ have been optimized. The overall loss of SoundLoCD is defined as

$$\mathcal{L} = \mathcal{L}_{vb}(x, y; \phi) - \frac{\lambda}{N}\sum_{x^j \in X'}\mathcal{L}_{vb}(x^j, y; \phi), \tag{3.8}$$

where N is the number of negative shuffled VQ sequences in X' and λ is the contrastive loss weight. $\mathcal{L}_{vb}(x, y)$ is a conditional variant of the diffusion loss function [Gu et al., 2022a; Zhu et al., 2022],

$$\mathcal{L}_{vb}(x, y; \phi) = \mathbb{E}_{q(x_0)}[D_{KL}[q(x_T|x_0)||p(x_T)]+ \tag{3.9}$$

$$\sum_{t=1}^{T}\mathbb{E}_{q(x_t|x_0)}[D_{KL}(q(x_{t-1}|x_t, x_0)||p_{\{\theta,\phi\}}(x_{t-1}|x_t, y))]$$

$$= \mathcal{L}_0 + \mathcal{L}_1 + \cdots + \mathcal{L}_{T-1} + \mathcal{L}_T, \tag{3.10}$$

where

$$\begin{aligned} \mathcal{L}_0 &= -\log p_{\{\theta,\phi\}}(x_0|x_1, y) \\ \mathcal{L}_{t-1} &= D_{KL}(q(x_{t-1}|x_t, x_0)||p_{\{\theta,\phi\}}(x_{t-1}|x_t, y)) \\ \mathcal{L}_T &= D_{KL}(q(x_T||x_0)||p(x_T)) \end{aligned} \tag{3.11}$$

3.3 Experiments and Results

In this section, we demonstrate the performance of SoundLoCD for text-to-sound generation with a comprehensive set of experiments. We first describe the datasets, implementation details, and evaluation metrics and then analyze our results with baseline method comparison and ablation studies.

3.3.1 Dataset

We evaluate our approach on AudioCaps [Kim et al., 2019a] and ESC50 [Piczak, 2015]. AudioCaps (≈50K audio clips) using 49256, 494, and 957 audio clips for training, testing, and validation respectively. Each of the audio clips in the training set contains a human-annotated caption while audio clips in the testing set and validation set contain five captions. We use the training set to train our model and verify the model with the validation set. ESC50 contains 2000 environmental sound clips for 50 classes prearranged into 5 folders. We use the first four folders for training and the 5th folder for testing.

3.3.2 Implementation details

We extract the log mel-spectrogram on a 22050 Hz sampling rate with 1024 frame size, 25% overlapping, and 80 Mel filter bins. All the audio clips are padded to 10s. The SoundLoCD is trained with a batch size of 24 and $5e^{-5}$ contrastive loss weight (λ) with 10 shuffled sample sizes (N) with a maximum of 200 epochs. All the experiments are performed on two NVIDIA GeForce RTX 3090 GPUs. We use a pre-trained MelGAN [Kumar et al., 2019] as the vocoder to obtain waveforms from generated spectrograms.

3.3.3 Evaluation metrics

Following Yang et al. [2023b], we use a series of objective evaluation metrics to verify our method including *Fréchet audio distance (FAD), inception score (ISc), Kullback-Leibler divergence (KL)*, and *kernel inception distance (KID)*. Among them, FAD and KL are calculated by a pre-trained VGGish [Chen et al., 2020a], while ISc and KID are computed from a pre-trained PANNs [Kong et al., 2020b]. The FAD is commonly used to verify the similarity and consistency between generated samples and real samples, while KL measures dissimilarity between the two probability distributions of generated samples and real samples. ISc and KID evaluate sample quality and diversity. For details on the calculation of these metrics, please refer to section 2.7.

Table 3.1: Model comparison on AudioCaps dataset. DiffSound⋆ was obtained from the released official checkpoint on AudioCaps. DiffSound∗ is our reproduced result using the original code on AudioCaps. T5-S stands for the T5-small text encoder.

Model	Train Config.	Params.	FAD ↓	ISc ↑	KL ↓	KID ↓
DiffSound	16 NVIDIA V100 (2 days)	434.23M	13.47	-	4.95	-
DiffSound⋆	16 NVIDIA V100 (2 days)	434.23M	12.45	16.95 ± 0.98	**3.85**	0.00233 ± 0.00025
DiffSound∗	2 NVIDIA RTX3090 (14 days)	434.23M	14.96	15.87 ± 1.15	4.39	0.00424 ± 0.00022
SoundLoCD+T5-S	2 NVIDIA RTX3090 (3 days)	2.38M	23.99	9.84 ± 0.39	5.23	0.00868 ± 0.00041
SoundLoCD+CLAP	2 NVIDIA RTX3090 (3 days)	2.38M	36.28	5.12 ± 0.42	6.77	0.01903 ± 0.00048
SoundLoCD+CLIP	2 NVIDIA RTX3090 (3 days)	2.38M	**12.27**	**17.36 ± 1.74**	**3.85**	**0.00212 ± 0.00021**

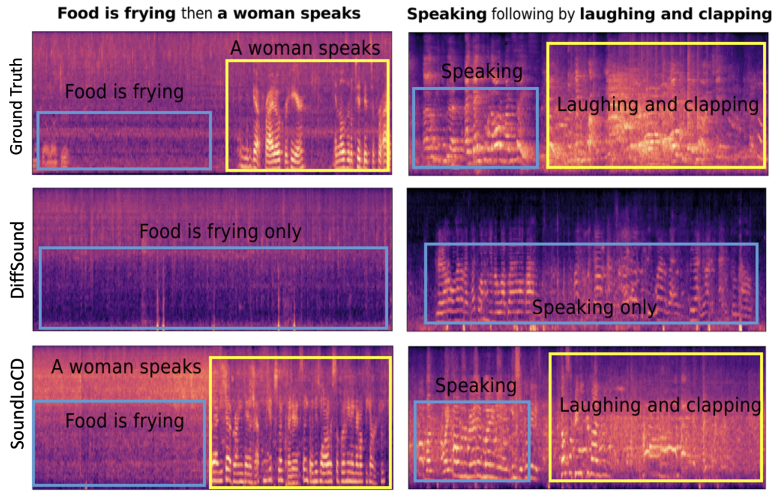


Figure 3.2: The visualization of generated samples by DiffSound and SoundLoCD compared with ground truth. *Samples generated by SoundLoCD have a stronger connection to the text condition compared with DiffSound.*

3.3.4 Results and Analysis

In our experiments, SoundLoCD is built upon a pre-trained DiffSound[2] with a codebook size of 256 and 100 diffusion steps, and is trained exclusively on the AudioCaps dataset. We select DiffSound as the primary baseline because SoundLoCD is a direct extension of DiffSound, differing only in the proposed conditional discrete contrastive learning objective and parameter-efficient LoRA adaptation. This comparison therefore isolates the contribution of our proposed method while keeping the backbone architecture, training data, and evaluation protocol identical. Although AudioCaps contains captions describing both single-event and complex acoustic scenes involving sequential or simultaneous sound events, neither DiffSound nor SoundLoCD explicitly models event composition. Instead, both methods are trained to synthesize complete audio clips directly from the same text descriptions. Consequently, the comparison evaluates the overall text-to-audio generation capability of

[2]https://github.com/yangdongchao/Text-to-sound-Synthesis

Table 3.2: Model comparison for fine-tuning on the ESC50 dataset.

	FAD ↓	ISc ↑	KL ↓	KID ↓
DiffSound	6.47	11.27 ± 1.21	2.54	0.00090 ± 0.00008
SoundLoCD	**6.18**	**10.85 ± 0.92**	**2.37**	**0.00075 ± 0.00007**

Table 3.3: Performance comparison of different LoRA implementation methods in SoundLoCD with CLIP, where W_q,W_k,W_v, and W_p represent weights of query, key, value, and projection layers that we adapt with LoRA in the transformer.

LoRA config.	Rank (r)	Params.	FAD ↓	ISc ↑	KL ↓	KID ↓
W_q & W_k	4	583K	13.70	17.12 ± 1.59	3.91	0.00231 ± 0.00023
W_q & W_k & W_v	4	836K	12.54	17.16 ± 1.31	4.06	0.00226 ± 0.00021
W_q & W_k & W_v & W_p	4	1.11M	13.31	16.84 ± 1.19	3.94	0.00270 ± 0.00022
W_q & W_k	8	1.11M	12.51	16.41 ± 1.79	3.88	0.00243 ± 0.00025
W_q & W_k & W_v	8	1.63M	**12.15**	17.30 ± 0.57	4.00	0.00218 ± 0.00021
W_q & W_k & W_v & W_p	8	2.38M	<u>12.27</u>	**17.36 ± 1.74**	**3.85**	<u>0.00212 ± 0.00021</u>
W_q & W_k	16	2.23M	12.85	16.78 ± 1.51	3.85	0.00247 ± 0.00023
W_q & W_k & W_v	16	3.27M	12.87	16.91 ± 1.42	4.01	0.00240 ± 0.00024
W_q & W_k & W_v & W_p	16	4.45M	12.48	17.32± 1.78	3.90	**0.00209 ± 0.00019**

the two models under identical conditions. We do not compare with more recent latent diffusion-based audio generation models [Liu et al., 2023a; Ghosal et al., 2023; Yuan et al., 2023; Huang et al., 2023e,a], as these models are trained on substantially larger mixtures of datasets for generalized audio generation, making direct comparisons in both generation quality and training efficiency less meaningful.

3.3.4.1 Model comparison

We inject LoRA parameters on the whole attention blocks (W_q, W_k, W_v, W_p) in the transformer with $r = 8$ and $\alpha = 16$. We observed that SoundLoCD achieves better overall performance (see table 3.1) on both general sound effect synthesis fidelity, diversity, and text-generated sound correspondences (ISc, KID, and FAD). Meanwhile, SoundLoCD significantly reduces the computational requirements for training as it only contains 2.38M trainable parameters. Visualizations in fig. 3.2 indicate that samples generated by SoundLoCD are more related to the text conditions than those generated by DiffSound. To validate the extensibility of models, we fine-tuned both DiffSound and SoundLoCD on the ESC-50 dataset, as shown in table 3.2, which further demonstrates the superior performance of SoundLoCD.

3.3.4.2 Text encoders

We then study whether text encoders affect the performance of SoundLoCD. We select three typical text encoders from pre-trained models of CLIP [Radford et al., 2021], CLAP [Wu et al., 2023a], and T5-small [Raffel et al., 2020]. CLIP is a contrastive language-image pretraining model, T5 is a text-to-text transformer, and CLAP is a contrastive language-audio pretraining model. All of these text encoders embed text tokens into a feature space of dimension 512. Our experiments indicate that SoundLoCD gets better performance with the CLIP text encoder which is not surprising as DiffSound checkpoint is trained with the CLIP text encoder. One possible explanation is that the CLIP embedding space aligns better with the discrete latent representation used by DiffSound.

3.3.4.3 Sensitivity of LoRA configuration

We also study the best configuration to inject LoRA parameters in SoundLoCD. In table 3.3, we inject LoRA parameters in different types of weights in attention blocks and r where the range of r is selected referring to Hu et al. [2022]. Injecting LoRA weights into the whole attention block reaches the best performance when fixing the r. This is consistent with the results of Hu et al. [2022] that it is preferable to adapt LoRA on more weight matrices. The rank is another critical factor of SoundLoCD; a smaller rank (e.g., $r = 4$) cannot capture enough information in ΔW. However, the model performance does not continue improving as the rank increases. with the increased rank. We choose the rank in SoundLoCD to achieve a trade-off between model performance and trainable parameters of the model.

3.3.5 Ablation Study

We then conduct an ablation study to verify the effectiveness of each component in SoundLoCD. According to table 3.4, our ablation study shows that directly applying LoRA on DiffSound with $r = 8$ to fine-tune the whole attention block fails to improve model performance, although the number of trainable parameters is significantly reduced. This indicates that DiffSound with LoRA may suffer from overfitting. In SoundLoCD, LoRA parameters are optimized by a contrastive learning strategy among $N + 1$ parallel diffusion processes, which further enhances the connection between text conditions and generating samples beyond DiffSound, leading to improved performance compared with the baseline.

3.3.6 Limitations and Discussion

Despite its improvements in training efficiency and semantic alignment, SoundLoCD has several limitations. As an adaptation of DiffSound, its generation capability is fundamentally bounded by the capacity of the underlying pre-trained backbone. While the proposed conditional discrete contrastive learning and LoRA adaptation substantially improve training efficiency and text-audio correspondence, they do not

Table 3.4: Ablation study on CDCD and LoRA in SoundLoCD.

	LoRA	CDCD	FAD ↓	ISc ↑
DiffSound			12.45	16.95 ± 0.87
DiffSound	✓		15.31	13.44 ± 0.73
SoundLoCD	✓	✓	**12.27**	**17.36 ± 1.74**

fundamentally improve the perceptual quality of the generated audio, which remains bounded by the underlying pre-trained DiffSound backbone.

3.4 Summary

In this chapter, we focus on improving the controllability and efficiency of existing methods in the T2S task. To this end, we proposed SoundLoCD, which efficiently reduces the model training cost and further enhances the connections between text conditions and generated samples by a contrastive discrete conditional diffusion model with LoRA, leading to better synthesis results. Compared with other discrete latent diffusion T2S models, it significantly reduces the training cost while achieving better performance on general fidelity. We studied how different text encoders and LoRA configurations affect SoundLoCD performance and conducted an ablation study to verify the contribution of each part of SoundLoCD. SoundLoCD could support small-scale training of T2S models for creative arts applications and lead to future studies that leverage fine-tuning techniques on large-scale models to improve training efficiency with limited resources.

Statement of Authorship

This chapter is based on the following publication:

Niu, X., Zhang, J., Walder, C., & Martin, C. P. (2024, April). Soundlocd: An Efficient Conditional Discrete Contrastive Latent Diffusion Model For Text-to-Sound Generation. In ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (pp. 261-265). IEEE.

Contributions of Candidate

I conceived the research idea, designed the proposed SoundLoCD framework, implemented the model, conducted all experiments, analyzed the results, and prepared the manuscript.

Contributions of Co-authors

Dr. Christian Walder, Dr. Charles Patrick Martin, and Dr. Jing Zhang supervised the research, provided technical discussions and feedback on the methodology, and contributed to the manuscript revision.

Chapter 4

Beyond Video-to-SFX: Video to Audio Synthesis with Environmentally Aware Speech

In chapter 3, we introduced a method for text-to-sound generation that produces high-quality sound effects closely aligned with complex textual descriptions. In practice, however, sound events often exhibit intricate temporal structures that are difficult to precisely capture through language description alone. Generating realistic, context-aware audio is particularly important in real-world applications such as video game development, where accurate temporal alignment and contextual coherence are crucial. This motivates a related task: video-to-audio (V2A) generation, which aims to synthesize temporally synchronized ambient sounds directly from video input. Videos inherently capture rich visual cues of physical events, which are often accompanied by background conversations or spoken interactions. However, existing V2A methods primarily focus on generating ambient sounds and struggle to produce intelligible and contextually appropriate speech. On the other hand, current environmental speech synthesis approaches are largely text-driven and fail to align speech generation temporally with the dynamic content of videos.

In this chapter, we introduce Beyond Video-to-SFX (BVS), a method to generate synchronized audio with environmentally aware intelligible speech for given videos. We introduce a two-stage modeling method: (1) stage one is a video-guided audio semantic (V2AS) model to predict unified audio semantic tokens conditioned on phonetic cues; (2) stage two is a video-conditioned semantic-to-acoustic (VS2A) model that refines semantic tokens into detailed acoustic tokens. Experiments demonstrate the effectiveness and flexibility of BVS in different scenarios, including video-to-context-aware speech synthesis and immersive audio background conversion[1].

[1]Demonstration page https://xinleiniu.github.io/BVS-demo/

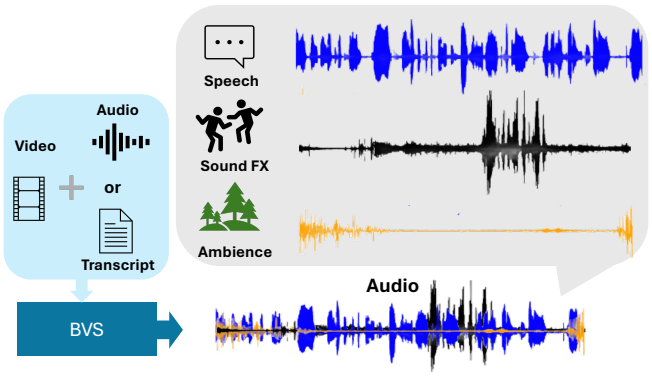


Figure 4.1: The proposed BVS framework.

4.1 Introduction

The generation of realistic, high-fidelity audio, including sound effects and speech, has become increasingly important for real-world applications such as film production, virtual reality (VR), and game development. To this end, useful generative models must generate audio that is coherent, expressive, and contextually aligned with the content of the given condition. Within this domain, video-to-audio (V2A) focuses on generating audio that is temporally and semantically aligned with visual cues in videos. In many scenarios, the audio track includes not only visually grounded sounds but also speech, and the speaker may not be directly observable. For example, a video clip of someone washing dishes might include background conversations that are not visually depicted. Motivated by this challenge, our work aims to develop a method capable of synthesizing audio that incorporates environmentally aware speech. The goal is to generate audio that is temporally aligned with the video while ensuring semantic consistency and incorporating context-aware spoken content.

Previous V2A studies [Luo et al., 2023; Wang et al., 2024b; Xue et al., 2024; Lee et al., 2024a; Chen et al., 2024b; Viertola et al., 2025] focus on Foley sound generation, aiming to synthesize ambient sounds that are temporally and semantically aligned with the video. More recent approaches Liu et al. [2024d]; Cheng et al. [2025] extend this paradigm by incorporating text prompts alongside video input. The inclusion of textual context enables models to capture more complex scenarios, such as the sound scene of 'a man speaking while he is playing tennis'. However, these methods remain limited to sound-event-level alignment driven by visual and/or textual cues and lack the capability to generate coherent and intelligible speech. Consequently, they often fail to generate natural spoken content, which restricts their real-world application where context-aware spoken content is essential.

A closely related task is video-to-speech (V2S) [Choi et al., 2025; Liang et al., 2025], which focuses on generating speech from videos with talking heads. However, V2S methods primarily address clean speech synthesis and do not handle sound effect generation, making them complementary but distinct from V2A approaches. A

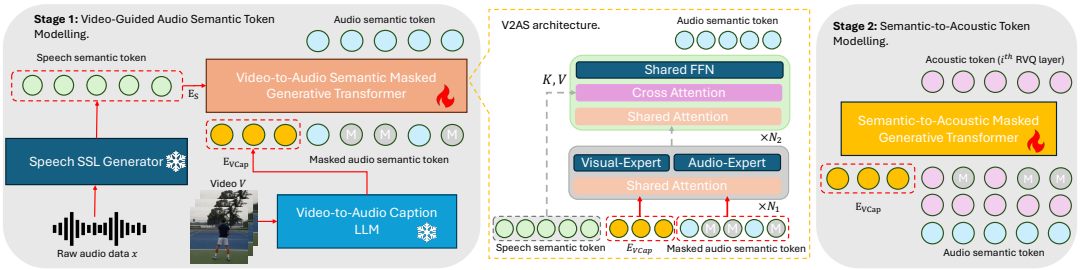


Figure 4.2: Overview of the BVS training pipeline. *In two stages, BVS generates unified audio semantic tokens from video and phonetic cues, then refines them into acoustic tokens. Snowflakes mark frozen pre-trained models, and fire icons represent trainable components.*

concurrent work by Tian et al. [2025a] demonstrates V2S generation with integrated sound effects; however, their approach is primarily designed for videos with talking heads. In parallel, environmental context-aware speech synthesis [Lee et al., 2024b; Jung et al., 2025] focuses on generating speech that incorporates background noise consistent with the given text prompts. These methods typically leverage a complex training strategy to learn the speech information. They first pretrain on synthetic noisy speech datasets and then fine-tune on real-world datasets transcribed by automatic speech recognition (ASR) models to improve speech modeling. Other related approaches [Glazer et al., 2025; He and Liu, 2025] explore mixing clean speech with ambient sounds to create contextually consistent audio environments. While these techniques are effective in achieving semantic consistency between speech and background context, they generally lack fine-grained temporal control over sound events, a capability that is critical for synchronizing sound effects with dynamic video content.

We identify two major limitations in previous works. Firstly, a key challenge is the scarcity of speech datasets that simultaneously provide context-rich environmental sounds, paired video streams, and ground-truth transcripts. Previous methods Lee et al. [2024b]; Jung et al. [2025] first train their models with synthesized noisy speech by mixing clean speech data for text-to-speech (TTS) training with environmental sound data, and then fine-tune using real-world unlabeled audio data containing speech with transcripts obtained from ASR models. However, ASR transcripts often suffer from inaccuracies and unstable segmentation in noisy speech, which introduces incorrect labels during speech learning. Meanwhile, synthetic noisy speech lacks natural video grounding since corresponding video streams cannot be created, limiting multimodal consistency. Secondly, current V2A models often fail to generate intelligible and natural speech, and often lack control over the speech content. As a result, additional post-production is often needed to remove unintelligible vocals and mix clean speech with ambient sounds for contextual alignment, which reduces their practicality in real-world applications.

In this chapter, we propose Beyond Video-to-SFX (BVS), a method for synthesizing video-synchronized audio that incorporates environmental-aware speech. As shown in fig. 4.1, BVS generates soundtracks that are temporally and semantically

aligned with visual content while integrating context-aware intelligible speech, guided by provided transcripts or audios containing spoken content. We make the following contributions:

1. We propose BVS, an improved V2A framework enabling intelligible speech synthesis, addressing a core limitation of previous approaches.

2. Our method is trained directly on raw audio data, eliminating the need for complex preprocessing steps such as synthetic noisy datasets or filtering ASR-generated transcripts.

3. We demonstrate that BVS is a flexible framework for various downstream tasks, including audio background conversion and video to context-aware speech synthesis.

4.2 Method

In this section, we present the detailed methodology of our proposed solution for generating synchronized ambient sounds with context-aware, intelligible speech from video input.

Generating realistic audio that accurately reflects a visual scene is inherently challenging, as different audio components, such as speech, sound effects, and ambience, are influenced by distinct factors. For example, speech typically occurs with irregular onset and offset times, is often produced by off-screen sources, and depends more on phonetic and linguistic cues than on visual context. In contrast, sound effects and ambient sounds are more directly associated with visible objects and acoustic scene elements, exhibiting a wide range of temporal dependencies based on physical interactions and environmental dynamics.

These fundamental differences present significant challenges for unified audio generation. A single model tasked with generating audios containing both intelligible speech and natural ambient sounds must effectively integrate and balance heterogeneous cues (i.e., both phonetic and visual cues) while maintaining temporal consistency. Moreover, to create a convincing immersive audio experience for a given video, all audio tracks must remain coherent and share consistent environmental characteristics.

To address this, we introduce an audio semantic token space as a unified representation of speech and ambient sounds inspired by two-stage TTS pipelines [Wang et al., 2025b,c]. The semantic tokens capture abstract audio information, reducing the gap between multi-modal conditions and acoustic outputs. As shown in fig. 4.2, our BVS pipeline operates in two stages: (1) fuse information across speech, visual, and contextual modalities into audio semantic tokens, and (2) refine these audio semantic tokens into acoustic outputs. This separation reduces the burden of a single-stage model that needs to learn both multimodal fusion and acoustic generation simultaneously.

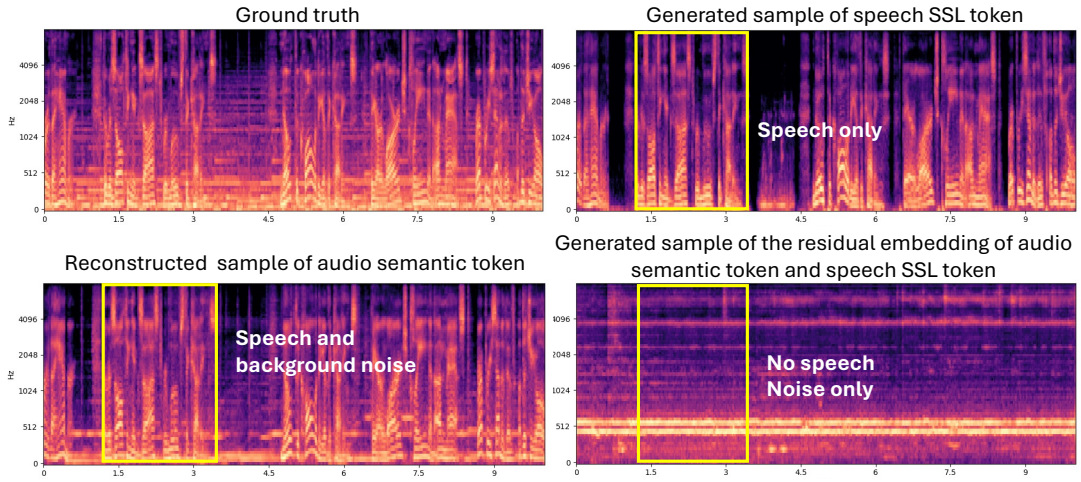


Figure 4.3: The generated sample of residual SSL highlights background noise, indicating sound effect is independent to speech information.

4.2.1 Video-Guided Audio Semantic Token Modeling

In the first stage, we design a video-to-audio semantic (V2AS) masked generative transformer, which generates audio semantic tokens by jointly integrating the speech and sound effects from video and phonetic cues. As mentioned before, speech, sound effects, and ambient sounds are each related to different objects. We split the audio semantic token generation into two components: (1) clean speech semantic token generation in the form of speech SSL token; (2) audio semantic fusion according to speech and video cues.

Noisy Audio to Speech Semantic Token. We aim to mitigate inaccurate transcription in previous methods [Lee et al., 2024b; Jung et al., 2025] and directly train models on raw noisy audios containing speech. To guide the model in identifying speech regions within raw audio, we leverage a pre-trained speech SSL token generator [Wang et al., 2025c] trained for the speech enhancement (SE) task to extract speech information from noisy inputs. Rather than employing the full model to directly separate clean speech waveform, we use its first-stage generative transformer to predict speech self-supervised learning (SSL) token sequences in the quantized W2V-BERT feature space [Chung et al., 2021], obtained via a VQ-VAE [Wang et al., 2025b]. Formally, given a raw audio input x, the pre-trained SE masked generative transformer captures speech SSL tokens S_s from $p_\phi(S_s|x)$, where ϕ denotes the parameter of the pre-trained SE model. This design emphasizes capturing of speech information, rather than reconstructing the clean speech waveform, which offers two key advantages: (1) it disentangles speech information more effectively under noisy conditions of the raw audio, and (2) it reduces downstream decoding errors by using compact speech SSL tokens.

Audio Semantic Fusion. While speech semantic tokens allow the model to be aware of speech regions directly from raw audio, the remaining challenge is to fuse the

speech information with video context while generating synchronized audios from visual cues. To address this, we make use of the quantized W2V-BERT feature space as the unified audio semantic token space, which is motivated by our empirical findings: Although trained primarily on speech, SSL models also encode non-speech acoustic cues. This is also pointed out in Chang et al. [2025]. We leveraged a pretrained semantic to acoustic generative model to reconstruct the waveform for a given SSL token sequence, including both speech and background noise. We empirically observed that the reconstructed audio contains both intelligible speech and matching background noise. fig. 4.3 further illustrates this empirical finding. We also provide the audio samples of our Demonstration page. Reconstructions of the W2V-BERT SSL tokens preserve both speech and non-speech information. Our empirical analysis further shows that the residual difference between speech SSL embeddings and full audio semantic embeddings potentially corresponds to non-speech components such as background noise or ambient sounds. This suggests that speech and non-speech information can be disentangled and recombined in the SSL embedding space. Motivated by this observation, the V2AS model predicts audio semantic token sequences in the same quantized W2V-BERT feature space that fuses non-speech acoustic information with speech information into audio semantic tokens using complementary cues from speech SSL tokens and videos. This process is shown on the right-hand side of stage 1 in fig. 4.2. By predicting semantic tokens rather than raw acoustic outputs, the model focuses on learning cross-modal relationships between speech, ambient sounds, and visual context.

Training Objective. We use the mask-and-predict learning strategy from [Chang et al., 2022] and mask the ground truth audio semantic token sequence S_a at each time step t with a corresponding binary mask M_t. Items in S_a are replaced with a special [MASK] token if $m_t^i = 1$, otherwise, S_a^i remain unchanged. The resulting masked token sequence is denoted as S_a^M. Each m_t^i is independent and identically distributed (i.i.d) to a Bernoulli distribution with parameter $\gamma(t) \in (0, 1]$. Thus, the V2AS model is formulated as $p_{\theta_{\text{V2AS}}}(S_a|S_a^M, V, S_s)$, where V represents video conditions. We train the model using cross-entropy loss between predicted audio semantic tokens $\hat{S}_a$ and ground truth S_a as

$$\mathcal{L}_{\text{V2AS}} = -\sum_{s \in S_a^M} \mathbb{I}(s = [\text{MASK}]) \log[P_\theta(\hat{s} = s|S_a, S_s, V)] \tag{4.1}$$

Architecture. To further enhance the model's video understanding ability, we incorporate visual representations enriched by audio captions generated with a V2Cap large language model (LLM) Liu et al. [2024d], which we denote as E_{VCap}. The detailed V2AS architecture is shown in fig. 4.2 (in the yellow dash box), which extends a V2A transformer framework from Su et al. [2024] by injecting speech SSL embeddings into extra cross-attention blocks as K and V, where $N_1 = 12$ and $N_2 = 14$. This integration improves speech localization and intelligibility, and captures non-speech details guided by video context. During training, we concatenate the masked audio semantic embeddings E_a^M and E_{VCap} along the temporal axis. During inference, the

V2AS model generates audio semantic token sequences conditioned on the video prompt and speech SSL tokens. The speech SSL token sequences can be obtained by a unified speech generative transformer such as Metis stage 1 [Wang et al., 2025c], which takes transcripts, audio, or other inputs.

4.2.2 Semantic-to-Acoustic Token Modeling

In the second stage, we train a video-guided semantic-to-acoustic (VS2A) model to obtain acoustic token sequences conditioned on audio semantic token sequences and E_{VCap}. The VS2A model refines generation by mapping abstract audio semantic representations into acoustic tokens while leveraging video context to enhance the alignment. Following Wang et al. [2025b], the VS2A model is a masked generative transformer that produces a multi-layer acoustic $A^{1:K}$, where K denotes the number of codebooks, using iterative parallel decoding within each layer. For simplicity, we adopt separate VS2A models: one dedicated to predicting the first-layer acoustic codec and another for the subsequent layers. We denote the i-th acoustic token layer as A^i, and formulate the model as $p_{\theta_{\text{VS2A}}}(A^i|V, A^{1:i-1})$. During training, we simply sum the embeddings of the audio semantic tokens and the embeddings of the acoustic codec tokens from layer 1 to i, which is denoted as $E^M_{C_i}$. Then we concatenate E_{VCap} and $E^M_{C_i}$ as input to the model. We use cross-entropy loss between predicted acoustic tokens and ground truth acoustic tokens as

$$\mathcal{L}_{\text{VS2A}} = -\sum_{a^i \in A^i} \mathbb{I}(a^i = [\text{MASK}]) \log[P_\theta(\hat{a}^i = a^i|S_a, V)] \tag{4.2}$$

During inference, the VS2A model generates acoustic tokens progressively from coarse to fine across layers, employing iterative parallel decoding within each layer.

4.3 Experiments and Results

In this section, we conduct comprehensive experiments to evaluate the proposed method, including two downstream applications (i.e., video to audio with context-aware speech and immersive audio background conversion) and ablation studies.

4.3.1 Dataset

We train our model on the filtered English subset of AudioSet-Speech [Gemmeke et al., 2017; Lee et al., 2024b] (AS-Speech), which contains approximately 580k audio–video pairs. Each sample has a 10-second duration and contains both speech and various environmental sound events. For evaluation, we adopt AC-filtered[2], which includes audio sourced from AudioCap [Kim et al., 2019a], captions, and transcripts. As AudioCaps is derived from AudioSet, we retrieve the corresponding

[2]https://github.com/glory20h/voiceldm-data

video streams via the YouTube IDs, resulting in a curated set of 152 samples, which we refer to as AS-filtered dataset.

4.3.2 Implementation Details

We train V2AS and VS2A models on three NVIDIA A100 (80GB) GPUs for four and three days, respectively, using a batch size of 32, a learning rate of 2×10^{-4}, 20k and 32k warm-up steps, for approximately 80 and 34 epochs, respectively. We optimise these models with the AdamW Loshchilov and Hutter [2019] optimiser. We adopt VATT [Liu et al., 2024d] transformer with extra cross-attention blocks for our V2AS model, and we adopt the S2A model architecture in Wang et al. [2025b] for the VS2A model. For the video modality, we employ the EVA-CLIP image encoder Sun et al. [2023] to extract mean-pooled frame features sampled at 5 fps, yielding a visual sequence of size 50×768 for a 10-second video. Following Liu et al. [2024d], the visual sequence passes through a pre-trained V2Cap LLaMa and we extract its hidden state as input features. We add two learnable modality-specific embeddings to the concatenated inputs in the V2AS model. We use Metis SE stage 1 model to obtain speech SSL tokens Wang et al. [2025c]. Audio SSL features are extracted from W2V-BERT 2.0 [Chung et al., 2021], which are then quantized into discrete tokens at 50 tokens per second by the pre-trained VQ-VAE in Wang et al. [2025b]. Acoustic tokens are obtained from 16kHz Encodec [Défossez et al., 2022] with four codebooks at a 50 Hz token rate. During inference, we use 16 steps for the V2AS model and [20, 10, 1, 1] steps for the VS2A model. The classifier-free guidance (CFG) scale is set to 5.0 for V2AS and 2.5 for VS2A.

4.3.3 Evaluation Metrics

We evaluate our method along three key dimensions: speech intelligibility, temporal alignment, and semantic perceptual similarity. For speech intelligibility, we report both the word error rate (WER) and ΔWER. WER is computed between the ground truth transcripts and those obtained from transcribing our generated audio using whisper-large-v3. Since ASR models are trained on transcribing clean speech, it may produce inaccurate transcripts on noisy inputs. We additionally report ΔWER, which measures the absolute difference in WER between the ground truth and generated results, as $\Delta\text{WER} = \frac{1}{N}\sum_i^N |\text{WER}(\text{gt}_i, t_i) - \text{WER}(\text{pred}_i, t_i)|$, where t_i represents the transcripts for audio sample i. To assess temporal alignment between video and generated audio, we follow Cheng et al. [2025] and report the DeSync score [Iashin et al., 2024]. For semantic and perceptual similarity, we follow Manor and Michaeli [2024a] and calculate the FAD and LPAPS scores, which are computed using the clap-laion-audio model in music-speech-audioset. At last, we report real-time factor (RTF) for the inference speed. For details on the calculation of these metrics, please refer to section 2.7.

Table 4.1: Model comparison of video to audio with context-aware speech. VoiceLDM is a text-guided context-aware TTS model, which supports text prompt only.

Method	Dataset	WER ↓	ΔWER ↓	FAD ↓	DeSync ↓	LPAPS ↓	RTF ↓
GT	-	27.4 %	3.2 %	-	-	-	-
VoiceLDM	Multiple	**17.2** %	22.5 %	**0.36**	1.49	6.48	0.29
VATT	VGGSound	98.9 %	75.6 %	0.43	**1.33**	6.65	0.12
BVS	AudioSet-Speech	<u>26.4</u> %	**21.9** %	<u>0.41</u>	**1.33**	**6.38**	0.32

Table 4.2: Experiment on immersive audio background conversion.

Method	ΔWER ↓	FAD ↓	DeSync ↓	LPAPS_S ↓	LPAPS_T ↓
SE + VATT	27.8 %	0.39	**1.22**	6.69	6.47
BVS	**26.9** %	**0.38**	1.32	**6.42**	**6.28**

4.3.4 Video to Audio with Context-Aware Speech

We evaluate the effectiveness of BVS on video-to-audio with context-aware speech by incorporating a pre-trained text-to-speech SSL generator in MaskGCT [Wang et al., 2025b]. As there are no existing methods that directly address this task, we select two most relevant baselines: VoiceLDM [Lee et al., 2024b], a text-to-context-aware TTS model that generates speech aligned with the semantic content of text prompts, and VATT [Liu et al., 2024d], a V2A model that produces ambient sounds temporally synchronized with visual input through audio captions.

We evaluate both baseline methods with their official checkpoints, and the results are reported in table 4.1. Although VoiceLDM focuses on synthesizing context-aware speech given text prompts and is trained on both AudioSet-Speech and multiple synthesized noisy speech datasets with a pre-trained CLAP text encoder [Wu et al., 2023a], its worse DeSync and LPAPS scores suggest that it struggles to generate audios with synchronized sound effects given prompts that contain events in temporal order. Conversely, VATT is designed to produce temporally aligned audio from videos; however, the high WER and ΔWER indicate VATT fails to generate intelligible speech. Compared to both baselines, BVS overcomes these limitations by generating temporally and semantically synchronized audio with intelligible speech given transcripts and videos. Notably, despite being trained solely on AudioSet-Speech, BVS achieves WER values closer to the ground truth with a lower ΔWER, which demonstrates the robustness of BVS in generating synchronized audio with context-aware intelligible speech. Finally, we report the overall inference time in seconds. Although BVS is a two-stage model, its inference time is comparable to the single-stage baselines, indicating that BVS maintains competitive efficiency despite its more complex architecture while simultaneously achieving greater controllability in video-to-audio with context-aware speech.

4.3.5 Immersive Audio Background Conversion

To show the flexibility of BVS in the real-world applications, we then demonstrate the effectiveness of BVS in immersive audio background conversion. Unlike the setting in section 4.3.4, this task takes noisy speech as input and converts it into a new context-aware speech that is naturally integrated within the given video's content. We randomly sample 500 video–audio pairs from the AS-filtered dataset, which serve as target video and source audio. Since no direct baselines exist, we construct a proxy baseline by first applying a Metis speech enhancement model [Wang et al., 2025c] to obtain clean speech, which is then directly mixed with sound effects generated by VATT. To prevent VATT from generating unintelligible human voices, we exclude caption inputs and make the model focus on generating sound effects. To measure the consistency of converted audio between source and target audio, we report LPAPS_T and LPAPS_S, where LPAPS_T measures similarity to the audio track of target video, and LPAPS_S measures similarity to the source audio. As shown in table 4.2, although the baseline achieves a lower DeSync score, our method outperforms it in terms of LPAPS and FAD, reflecting higher perceptual similarity to ground truth. In contrast, simple post hoc mixing of clean speech with generated sound effects often results in poor semantic alignment between speech and environmental context, whereas BVS produces coherent and context-aware audio.

4.3.6 Ablation study

We then conduct an ablation study to verify each of the proposed components in BVS. We first validate whether the audio semantic tokens obtained from W2V-BERT have the ability to represent non-speech information. As shown in the first row of table 4.3, we evaluate reconstruction results on AS-filtered using our VS2A model conditioned on ground-truth audio semantic tokens and videos. The metrics of reconstructed samples reflect that the VS2A model can generate audio that contains both speech and non-speech sounds, with better scores than generation results in section 4.3.4 and section 4.3.5. We then conduct ablation studies to evaluate the importance of the speech semantic token from the SE model and the audio semantic tokens in our two-stage method. Firstly, we remove the audio semantic tokens from BVS, converting it into a single-stage model that directly predicts acoustic tokens, which we denote as BVS-Single. Secondly, we remove speech SSL tokens obtained from raw audios by replacing the SE speech SSL generator with a transcript-based speech SSL generator used in MaskGCT [Wang et al., 2025b]. We use Whisper [Radford et al., 2023] to enable alignment of speech onset and offset times. We use filtered transcripts obtained from an ASR model by Lee et al. [2024b] to guide phonetic information. We refer to this as BVS-Text. As illustrated in table 4.3, both methods fail to learn speech information effectively with significantly higher WER and FAD. In BVS-Text, although the model is given transcripts with speech onset-offset timestamps, the model still fails to learn the complex alignment of speech and sound effects from video and transcripts. This indicates that the models fail to capture the relationship between video, speech, and context information. In contrast, our method with both speech SSL to-

Table 4.3: Ablation study. SP-SSL stands for speech semantic tokens, and audio SSL stands for unified audio semantic tokens.

Method	SP-SSL	Audio SSL	ΔWER ↓	FAD ↓	DeSync ↓	LPAPS ↓
VS2A	-	-	19.8 %	0.25	1.02	5.12
BVS-Single	✓		99.1 %	1.18	1.38	6.78
BVS-Text		✓	98.7 %	1.12	1.49	7.01
BVS (Ours)	✓	✓	**21.9** %	**0.41**	**1.33**	**6.38**

kens obtained directly from raw audios, and a two-stage design explicitly guides the model to disentangle, fuse, and refine the information of speech and sound effects.

4.3.7 Limitation and Discussion

Despite these promising results, BVS has several limitations. First, as BVS relies on speech self-supervised learning (SSL) models to construct a unified audio semantic token space, its performance on non-speech sound effects is inherently constrained by the representational capacity of speech-oriented SSL models. Second, due to the limited availability of large-scale video-to-audio datasets with accurate lip–speech correspondence, achieving precise lip synchronization remains challenging. Furthermore, BVS currently supports voice cloning using only a single speaker prompt. Consequently, for complex video scenes involving multiple speakers, the model may struggle to maintain consistent speaker identities and accurately align each speech segment with the corresponding speaker. These limitations suggest several promising directions for future research. In particular, developing more powerful unified speech–audio SSL representations may improve the generation of diverse environmental sounds, while incorporating richer multimodal datasets with accurate audio-visual alignment could further enhance lip synchronization. Extending the framework to support multi-speaker voice conditioning would also enable more realistic and coherent audio generation for complex conversational scenes.

4.4 Summary

In this chapter, we introduce BVS, a controllable video-to-audio generation framework that addresses the challenge of synthesizing synchronized audio with both intelligible and environmentally aware speech. By leveraging a pre-trained speech self-supervised learning (SSL) generator during training, BVS disentangles speech representations from noisy audio and effectively integrates complementary cues from video, speech, and audio modalities through a two-stage generation framework. Extensive experiments demonstrate that BVS can be applied to a variety of real-world scenarios, including both video-to-audio generation and speech-guided audio conversion tasks. By bridging the gap between sound generation and speech generation, BVS enables the synthesis of synchronized environmental sounds and speech within a unified framework, opening a new direction for controllable video-to-audio gen-

eration and providing a foundation for future research on multimodal audiovisual synthesis.

Statement of Authorship

This chapter is based on the following publication:

Niu, X., Ma, J., Harper-Harris, D., Zhang, X., Martin, C. P., & Zhang, J. (2026, May). Beyond Video-to-SFX: Video to Audio Synthesis with Environmentally Aware Speech. In ICASSP 2026-2026 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (pp. 22697-22701). IEEE.

This work was done during the internship at Dolby.

Contributions of Candidate

I conceived the research idea, designed the proposed BVS framework, implemented the model, conducted all experiments, analyzed the results, and prepared the manuscript.

Contributions of Co-authors

Dr. Jianbo Ma, Mr. Dylan Harper-Harris, Dr. Charles Patrick Martin, and Dr. Jing Zhang supervised the research, contributed to technical discussions, provided feedback on the methodology, and reviewed and revised the manuscript. Mr. Xiangyu Zhang, a fellow intern, contributed valuable discussions that helped shape the ideas underlying this work.

Chapter 5

Latent Optimal Paths by Gumbel Propagation for Variational Bayesian Dynamic Programming

In the previous chapters, we explored methods to improve training efficiency, flexibility, and controllability in the broader domain of sound generation. We first examined text-to-sound synthesis, where the challenge lies in mapping natural language conditions to diverse sound effects. We then extended this line of work to video-to-audio generation with context-aware speech, where speech, ambient sounds, and visual cues must be jointly modeled to produce coherent auditory scenes. Together, these studies emphasized the importance of multi-modal conditioning and controllable synthesis strategies for real-world applications. Building on these foundations, we now narrow our focus from general sound synthesis to the specialized domain of speech synthesis, where the objective is to generate natural and intelligible speech conditioned on transcripts or other phonetic cues. Speech synthesis is not only one of the most mature subfields of generative audio but also a crucial enabler of practical applications such as virtual assistants, accessibility tools, and human–computer interaction systems.

In this chapter, we take a theoretical perspective and propose a new formulation based on optimal paths to address the structural complexity inherent in speech synthesis. We propose the stochastic optimal path, which solves the classical optimal path problem by a probability-softening solution to tackle the structural dependency in speech synthesis. This is a unified approach that transforms a wide range of dynamic programming (DP) problems into directed acyclic graphs in which all paths follow a Gibbs distribution. We show the equivalence of the Gibbs distribution to a message-passing algorithm by the properties of the Gumbel distribution and give all the ingredients required for variational Bayesian inference of a latent path, namely Bayesian dynamic programming (BDP). We demonstrate the usage of BDP in the latent space of variational autoencoders (VAEs) and propose BDP-VAE which captures structured sparse optimal paths as latent variables. This enables end-to-end training for generative tasks in which models rely on unobserved structural information. In our experiment, we validate the behavior of the proposed approach and show-

case its applicability in two speech synthesis tasks: text-to-speech and singing voice synthesis[1].

5.1 Introduction

Optimal paths are often required in many generative tasks such as speech, music, and language modeling [Kim et al., 2020; Li et al., 2022; Cai et al., 2019]. These tasks involve the simultaneous identification of structured relationships between data and conditions. Finding optimal paths given a graph constraint with learned weights can highlight unobserved structural relationships that are not immediately apparent, thus potentially improving the interpretability and effectiveness of the models. In the context of optimal path problems, such as finding the shortest path in a graph, dynamic programming (DP) efficiently computes the solution by breaking the problem down into several sub-problems and finding optimal solutions within the sub-problems recursively.

Since the DP algorithm finds shortest paths using the max operator, it is non-differentiable which limits the usage of optimal paths in neural networks where gradient back-propagation is applied. As a workaround, previous works have approximated the max operator with smoothed functions to allow differentiation of DP algorithms [Verdu and Poor, 1987]. However, smoothed approximations lose the sparsity of solutions which makes hard assignments become soft assignments. Alternatively, some real-world generative applications [Ren et al., 2019, 2020; Jeong et al., 2021b; Li et al., 2022; Halperin et al., 2019; Peng et al., 2020; Liu et al., 2022; Popov et al., 2021], that require to integrate structured optimal paths, split the training strategies in which neural network models depend on sparse outputs from external DP aligners [McAuliffe et al., 2017b; Hasegawa-Johnson et al., 2005] or pre-trained models [Li et al., 2018]. However, these external components involve more than one training phase thus the model performance critically relies on them.

In this chapter, we explore a novel and unified method to obtain structured sparse optimal paths with DP and showcase its application in the latent space of variational autoencoders (VAEs). Instead of a smoothed approximation for the classical optimal path problem, we propose the stochastic optimal path, which is a probabilistic softening solution by defining a Gibbs distribution where the energy function is the path score. We show this to be equivalent to a message-passing algorithm on the directed acyclic graphs (DAG) using the max and shift properties of the Gumbel distribution. To learn the latent optimal paths, we give tractable closed-form ingredients for variational Bayesian inference (i.e., likelihood and KL divergence) using DP, namely Bayesian dynamic programming (BDP), as well as an efficient sampling algorithm, which enables VAEs to obtain latent optimal paths within a DAG and achieve end-to-end training on generative tasks that rely on sparse unobserved structural relationships. We make the following contributions:

[1]Implementation code: https://github.com/XinleiNIU/LatentOptimalPathsBayesianDP.

1. We present a unified framework that gives a probabilistic softening of the classical optimal path problem on DAGs. We notably give efficient algorithms in linear time for sampling, computing the likelihood, and computing the KL divergence, thereby providing all the ingredients required for variational Bayesian inference with a latent optimal path.

2. We introduce the BDP-VAE framework that learns sparse optimal paths in the latent space. In the case of conditional generation, the data is not observed during inference; it is difficult to form the distribution statistics (i.e., edge weights) in the prior encoder. We give an alternative and flexible method to form the distribution statistics on the conditional prior by making use of a flow-based model.

3. We demonstrate how BDP-VAE achieves end-to-end training on two generation tasks in the speech synthesis domain, including text-to-speech (TTS) and singing voice synthesis (SVS).

5.2 Preliminary

This section provides background on the notation definition of a DAG, a definition of the traditional optimal path problem given a DAG, and properties of the Gumbel random variable for the method introduced in this chapter.

5.2.1 Definition of a Graph

We denote $\mathcal{R} = (\mathcal{V}, \mathcal{E})$ be a directed acyclic graph with nodes $\mathcal{V}$ and edges $\mathcal{E}$. Assume without loss of generality that the nodes are numbered in topological order, such that $\mathcal{V} = (1, 2, \ldots, N)$ and $u < v$ for all $(u, v) \in \mathcal{E}$. Further, we assume that 1 is the only node without parents and N the only node without children. We denote the edge weights $\mathbf{W} \in \mathbb{R}^{N \times N}$ with $w_{i,j} = -\infty$ for all $(u, v) \notin \mathcal{E}$. Let $\mathcal{Y}(1, v)$ be the set of all paths from 1 to v. Associate with each path $\mathbf{y} = (y_1, y_2, \ldots, y_{|\mathbf{y}|})$ a score obtained by summing edge weights along the path, defined as $||\mathbf{y}||_{\mathbf{W}} = \sum_{i=2}^{|\mathbf{y}|} w_{y_{i-1}, y_i} = \sum_{(u,v) \in \mathbf{y}} w_{u,v}$, where the final expression introduces the notation $(u, v) \in \mathbf{y}$ for the edges (u, v) that make up path $\mathbf{y}$. Denote the set of parents of node v by $\mathcal{P}(v) = \{u : (u, v) \in \mathcal{E}\}$ and the set of children of node u by $\mathcal{C}(u) = \{v : (u, v) \in \mathcal{E}\}$.

5.2.2 Non-Stochastic Optimal Paths

The traditional optimal path problem is to find the highest scoring path from node 1 to node N,

$$\mathbf{y}^* = \operatorname*{argmax}_{\mathbf{y} \in \mathcal{Y}(1,N)} ||\mathbf{y}||_{\mathbf{W}}. \tag{5.1}$$

This can be solved in $O(|\mathcal{E}|)$ time by iterating in topological order. The score $\xi(\cdot)$ is defined as

$$\xi(1) = 0 \tag{5.2}$$

$$\forall v \in \{2, 3, \ldots, N\}, \quad \xi(v) = \max_{u \in \mathcal{P}(v)} \xi(u) + w_{u,v}, \tag{5.3}$$

after which $\mathbf{y}^*$ is obtained (in reverse) by tracing from N to 1, following the path of nodes u for which the maximum was obtained in the above.

5.2.3 Gumbel Random Variable

Let $\mathcal{G}(\mu)$ denote the unit scale Gumbel random variable with location parameter μ and probability density function

$$\mathcal{G}(x|\mu) = \exp(-(x - \mu) - \exp(x - \mu)). \tag{5.4}$$

We now review the properties of the Gumbel which we will exploit in section 5.3. Let $X \sim \mathcal{G}(\mu)$. The Gumbel distribution is closed under shifting, with

$$X + \text{const.} \sim \mathcal{G}(\mu + \text{const.}). \tag{5.5}$$

Let $X_i \sim \mathcal{G}(\mu_i)$ for all $i \in \{1, 2, \ldots, m\}$. The Gumbel is also closed under the max operation, with

$$\max(\{X_1, X_2, \ldots, X_m\}) \sim \mathcal{G}(\log \sum_{i=1}^{m} \exp(\mu_i)). \tag{5.6}$$

Finally, there is a closed-form expression for the index which obtains the maximum in the above expression:

$$p(k = \underset{i \in \{1,2,\ldots,m\}}{\text{argmax}}\, X_i) = \frac{\exp(\mu_k)}{\sum_{i=1}^{m} \exp(\mu_i)}. \tag{5.7}$$

5.3 Bayesian Dynamic Programming

In this section, we propose a stochastic approach to seek optimal paths. We denote a distribution family given a DAG with edge weights and give ingredients required for variational Bayesian inference by using DP with Gumbel propagation, namely, Bayesian dynamic programming.

5.3.1 Stochastic Optimal Paths

In the stochastic approach, every possible path $\mathbf{y} \in \mathcal{Y}$ on $\mathbf{W}$ follows a Gibbs distribution given a DAG $\mathcal{R}$, edge weights $\mathbf{W}$ and temperature parameter α defined by theorem 5.3.1.

Definition 5.3.1. Denote by

$$\mathcal{D}(\mathcal{R}, \mathbf{W}, \alpha) \tag{5.8}$$

the Gibbs distribution over $\mathbf{y} \in \mathcal{Y}(1, N)$ with probability mass function

$$\mathcal{D}(\mathbf{y}|\mathcal{R}, \mathbf{W}, \alpha) = \frac{\exp(\alpha \left\|\mathbf{y}\right\|_{\mathbf{W}})}{\sum_{\widehat{\mathbf{y}} \in \mathcal{Y}(1,N)} \exp(\alpha \left\|\widehat{\mathbf{y}}\right\|_{\mathbf{W}})}. \tag{5.9}$$

Despite the intractable form of the denominator in eq. (5.9), we provide the ingredients necessary for approximate Bayesian inference for latent distribution (unobserved) $\mathcal{D}$. In particular, we can efficiently compute the normalized likelihood (theorem 5.3.6), sample (theorem 5.3.7), and compute the KL divergence within $\mathcal{D}(\mathcal{R}, \cdot, \alpha)$ (theorem 5.3.10) in linear time (theorem 5.3.11).

5.3.2 Gumbel Propagation

Gumbel propagation offers an equivalent formulation of theorem 5.3.1 that lends itself to dynamic programming by the properties in eq. (5.7) and eq. (5.5) as per the following result.

Lemma 5.3.2. *Let*

$$Y = \operatorname*{argmax}_{\mathbf{y} \in \mathcal{Y}(1,N)} \left\{\Omega_{\mathbf{y}}\right\}, \tag{5.10}$$

where for all $\mathbf{y} \in \mathcal{Y}(1, N)$*,*

$$\Omega_{\mathbf{y}} = \alpha \left\|\mathbf{y}\right\|_{\mathbf{W}} + G_{\mathbf{y}}, \tag{5.11}$$

$$G_{\mathbf{y}} \sim \mathcal{G}(0). \tag{5.12}$$

Then the probability of $Y = \mathbf{y}$ *is given by* (5.9)*.*

Proof. The result follows directly from (5.6) and (5.5). □

Let the definitions of $\Omega_{\mathbf{y}}$ and $G_{\mathbf{y}}$ extend to all $\mathbf{y} \in \bigcup_{u=1}^{N} \mathcal{Y}(1, u)$, which is the set of all partial paths. We define for each node $v \in \mathcal{V}$ the real-valued random variable

$$Q_v = \max_{\mathbf{y} \in \mathcal{Y}(1,v)} \left\{\Omega_{\mathbf{y}}\right\}. \tag{5.13}$$

Lemma 5.3.3. *The* Q_v *are Gumbel distributed with*

$$Q_v \sim \mathcal{G}(\mu_v), \tag{5.14}$$

where

$$\mu_v = \log \sum_{\mathbf{y} \in \mathcal{Y}(1,v)} \exp(\alpha \left\|\mathbf{y}\right\|_{\mathbf{W}}). \tag{5.15}$$

Proof. The result follows directly from (5.6) and (5.5). □

The motivation for constructing Q_v is Q_v allows us to set up a recursion on the entire DAG $\mathcal{R}$. We now state the first main result in theorem 5.3.4.

Lemma 5.3.4. *The location parameters μ_v satisfy the recursion*

$$\mu_1 = 0 \tag{5.16}$$
$$\mu_v = \log \sum_{u \in \mathcal{P}(v)} \exp(\mu_u + \alpha\, w_{u,v}). \tag{5.17}$$

for all $v \in \{2, 3, \ldots, N\}$.

Proof. From the DAG structure of $\mathcal{R}$ we have

$$\mathcal{Y}(1, v) = \bigcup_{u \in \mathcal{P}(v)} \{\mathbf{y} \cdot v : \forall \mathbf{y} \in \mathcal{Y}(1, u)\}, \tag{5.18}$$

where $\mathbf{y} \cdot v = (y_1, y_2, \ldots, y_{|y|}, v)$ denotes concatenation.

Then we have

$$\begin{aligned}
\mu_v &= \log \sum_{\mathbf{y} \in \mathcal{Y}(1,v)} \exp(\alpha \left\|\mathbf{y}\right\|_{\mathbf{W}}) && (5.19)\\
&= \log \sum_{\mathbf{y} \in \bigcup_{u \in \mathcal{P}(v)} \{\widehat{\mathbf{y}} \cdot v : \forall \widehat{\mathbf{y}} \in \mathcal{Y}(1,u)\}} \exp(\alpha \left\|\mathbf{y}\right\|_{\mathbf{W}}) && (5.20)\\
&= \log \sum_{u \in \mathcal{P}(v)} \sum_{\widehat{\mathbf{y}} \in \mathcal{Y}(1,u)} \exp(\alpha \left\|\widehat{\mathbf{y}} \cdot v\right\|_{\mathbf{W}}) && (5.21)\\
&= \log \sum_{u \in \mathcal{P}(v)} \sum_{\widehat{\mathbf{y}} \in \mathcal{Y}(1,u)} \exp\left(\alpha \left\|\widehat{\mathbf{y}}\right\|_{\mathbf{W}} + \alpha\, w_{u,v}\right) && (5.22)\\
&= \log \sum_{u \in \mathcal{P}(v)} \exp\left(\log \sum_{\widehat{\mathbf{y}} \in \mathcal{Y}(1,u)} \exp\left(\alpha \left\|\widehat{\mathbf{y}}\right\|_{\mathbf{W}}\right) + \alpha\, w_{u,v}\right) && (5.23)\\
&= \log \sum_{u \in \mathcal{P}(v)} \exp\left(\mu_u + \alpha\, w_{u,v}\right), && (5.24)
\end{aligned}$$

where eq. (5.19) restates (5.15), eq. (5.20) follows from (5.18), eq. (5.21) expands the summation, eq. (5.22) follows from the definition of $\left\|\mathbf{y}\right\|_{\mathbf{W}}$, eq. (5.23) follows from the identity

$$\sum_i \exp(a + b_i) = \exp(a + \log \sum_i \exp(b_i)), \tag{5.25}$$

eq. (5.24) follows from

$$\mu_v = \log \sum_{\mathbf{y} \in \mathcal{Y}(1,v)} \exp(\alpha \left\|\mathbf{y}\right\|_{\mathbf{W}}) \tag{5.26}$$

yielding eq. (5.17) as required. □

5.3.3 Sampling and Likelihood

Then we state an alternative normalized likelihood of a sampled path $\mathbf{y}$ by theorem 5.3.7 as theorem 5.3.6 according to a transition matrix defined in theorem 5.3.5. The transition matrix in theorem 5.3.5 can be computed according to the location parameter μ defined in theorem 5.3.4 directly.

Lemma 5.3.5. *Let paths* $\mathbf{y} = (y_1, y_2, \ldots, y_{|\mathbf{y}|})$ *denote the component of the random variable* Y *defined in* (5.10)*, given that* $Y = (y_1, y_2, \ldots, y_{|\mathbf{y}|})$*. The probability of the transition* $v \to u$ *is*

$$\pi_{u,v} \equiv p(y_{i-1} = u | y_i = v, u \in \mathcal{P}(v)) \tag{5.27}$$

$$= \frac{\exp(\mu_u + \alpha\, w_{u,v})}{\exp(\mu_v)}, \tag{5.28}$$

for all $i \in \{2, 3, \ldots, N\}$.

Proof. Assuming without loss of generality that $v = N$,

$$\pi_{u,v} = p(y_{i-1} = u | y_i = v, u \in \mathcal{P}(v)) \tag{5.29}$$

$$= \frac{p(y_{i-1} = u, y_i = v | u \in \mathcal{P}(v))}{p(y_i = v)} \tag{5.30}$$

$$= \frac{1}{p(y_i = v)} \sum_{\widetilde{\mathbf{y}} \in \mathcal{Y}(1,u)} p(Y = \widetilde{\mathbf{y}} \cdot v) \tag{5.31}$$

$$= \frac{1}{p(y_i = v)} \sum_{\widetilde{\mathbf{y}} \in \mathcal{Y}(1,u)} \frac{\exp(\alpha \left\| \widetilde{\mathbf{y}} \cdot v \right\|_{\mathbf{W}})}{\sum_{\widehat{\mathbf{y}} \in \mathcal{Y}(1,N)} \exp(\alpha \left\| \widehat{\mathbf{y}} \right\|_{\mathbf{W}})} \tag{5.32}$$

$$\propto \sum_{\widetilde{\mathbf{y}} \in \mathcal{Y}(1,u)} \exp(\alpha \left\| \widetilde{\mathbf{y}} \right\|_{\mathbf{W}} + \alpha\, w_{u,v}) \tag{5.33}$$

$$\propto \exp\big(\log \sum_{\widetilde{\mathbf{y}} \in \mathcal{Y}(1,u)} \exp(\alpha \left\| \widetilde{\mathbf{y}} \right\|_{\mathbf{W}}) + \alpha\, w_{u,v} \big) \tag{5.34}$$

$$\propto \exp\left(\mu_u + \alpha\, w_{u,v}\right) \tag{5.35}$$

$$= \frac{\exp\left(\mu_u + \alpha\, w_{u,v}\right)}{\sum_{\widetilde{u} \in \mathcal{P}(v)} \exp\left(\mu_{\widetilde{u}} + \alpha\, w_{\widetilde{u},v}\right)} \tag{5.36}$$

$$= \frac{\exp(\mu_u + \alpha\, w_{u,v})}{\exp(\mu_v)}, \tag{5.37}$$

where eq. (5.30) rewrite (5.29) by the rule of conditional probability, eq. (5.31) marginalises over $\mathbf{y}_{<i-1}$, eq. (5.32) extend the probability by eq. (5.9), eq. (5.33) neglects factors that do not depend on u, eq. (5.34) uses the identity of (5.25) then get eq. (5.35) according to eq. (5.15). eq. (5.36) makes the normalization explicit and eq. (5.37) uses (5.17) to recover eq. (5.28) as required. □

Corollary 5.3.6. *The path probability may be written*

$$\mathcal{D}(\mathbf{y}|\mathcal{R},\mathbf{W},\alpha) = \prod_{(u,v)\in\mathbf{y}} \pi_{u,v}. \tag{5.38}$$

Corollary 5.3.7. *Paths* $\mathbf{y} \sim \mathcal{D}(\mathcal{R},\mathbf{W},\alpha)$ *may be sampled (in reverse) by*

1. *Initialising* $v = N$,
2. *sampling* $u \in \mathcal{P}(v)$ *with probability* $\pi_{u,v}$,
3. *setting* $v \leftarrow u$,
4. *if* $v = 1$ *then stop, otherwise return to step 2.*

5.3.4 KL Divergence

Given $\mathcal{D}(\mathcal{R},\mathbf{W},\alpha)$ and $\mathcal{D}(\mathcal{R},\mathbf{W}^{(r)},\alpha)$, where $\mathbf{W}^{(r)}$ are edge weights having different values to $\mathbf{W}$. We give a tractable closed-form of the KL divergence within the distribution family of $\mathcal{D}(\mathcal{R},\cdot,\alpha)$ in theorem 5.3.10.

Definition 5.3.8. We denote the total probability of paths that include a given edge $(u,v) \in \mathcal{E}$ by

$$\omega_{u,v} \equiv \sum_{\{\mathbf{y}\in\mathcal{Y}(1,N):(u,v)\in\mathbf{y}\}} \mathcal{D}(\mathbf{y}|\mathcal{R},\mathbf{W},\alpha). \tag{5.39}$$

The quantity in the above definition may be computed using two dynamic programming passes, one topologically ordered and the other reverse topologically ordered, by applying the following

Lemma 5.3.9. *For all* $(u,v) \in \mathcal{E}$,

$$\omega_{u,v} = \pi_{u,v}\,\lambda_u\,\rho_v, \tag{5.40}$$

where we have the recursions

$$\lambda_1 = 1 \tag{5.41}$$

$$\lambda_v = \sum_{u\in\mathcal{P}(v)} \lambda_u\,\pi_{u,v} \tag{5.42}$$

for all $v \in \{2,3,\ldots,N\}$ *(in topological order w.r.t.* $\mathcal{R}$*), and*

$$\rho_N = 1 \tag{5.43}$$

$$\rho_u = \sum_{v\in\mathcal{C}(u)} \rho_v \pi_{u,v} \tag{5.44}$$

for all $u \in \{N-1,N-2,\ldots,1\}$ *(in reverse topological order w.r.t.* $\mathcal{R}$*).*

Proof. From the DAG structure of $\mathcal{R}$ we have, for all edges $(u, v) \in \mathcal{E}$,

$$\mathcal{Y}(1, v) = \bigcup_{\mathbf{l} \in \mathcal{Y}(1,u)} \bigcup_{\mathbf{r} \in \mathcal{Y}(v,N)} \mathbf{l} \cdot \mathbf{r}, \tag{5.45}$$

where we recall $\mathbf{l} \cdot \mathbf{r}$ denotes concatenation. We therefore have

$$\begin{aligned}
\omega_{u,v} &= \sum_{\{\mathbf{y} \in \mathcal{Y}(1,N) : (u,v) \in \mathbf{y}\}} \mathcal{D}(\mathbf{y} | \mathcal{R}, \mathbf{W}, \alpha) && (5.46)\\
&= \sum_{\{\mathbf{y} \in \mathcal{Y}(1,N) : (u,v) \in \mathbf{y}\}} \prod_{(u',v') \in \mathbf{y}} \pi_{u',v'} && (5.47)\\
&= \sum_{\mathbf{l} \in \mathcal{Y}(1,u)} \sum_{\mathbf{r} \in \mathcal{Y}(v,N)} \prod_{(u',v') \in \mathbf{l} \cdot \mathbf{r}} \pi_{u',v'} && (5.48)\\
&= \sum_{\mathbf{l} \in \mathcal{Y}(1,u)} \sum_{\mathbf{r} \in \mathcal{Y}(v,N)} \pi_{u,v} \prod_{(u',v') \in \mathbf{l}} \pi_{u',v'} \prod_{(u',v') \in \mathbf{r}} \pi_{u'',v''} && (5.49)\\
&= \pi_{u,v} \underbrace{\sum_{\mathbf{l} \in \mathcal{Y}(1,u)} \prod_{(u',v') \in \mathbf{l}} \pi_{u',v'}}_{\equiv \lambda_u} \underbrace{\sum_{\mathbf{r} \in \mathcal{Y}(v,N)} \prod_{(u',v') \in \mathbf{r}} \pi_{u'',v''}}_{\equiv \rho_v}, && (5.50)
\end{aligned}$$

where (5.46) restates (5.39), (5.47) uses the chain rule of probability using eq. (5.38), (5.48) follows from (5.45), (5.49) re-factors the product and the final (5.50) rearranges sums and products.

Finally, it is straightforward to show by induction that the λ_u and ρ_v defined in (5.50) above obey the classic sum-of-product recursions given in the statement of the lemma. For example,

$$\begin{aligned}
\lambda_v &= \sum_{u \in \mathcal{P}(v)} \pi_{u,v} \lambda_u, && (5.51)\\
&= \sum_{u \in \mathcal{P}(v)} \pi_{u,v} \sum_{\mathbf{l} \in \mathcal{Y}(1,u)} \prod_{(u',v') \in \mathbf{l}} \pi_{u',v'} && (5.52)\\
&= \sum_{u \in \mathcal{P}(v)} \sum_{\mathbf{l} \in \mathcal{Y}(1,u)} \prod_{(u',v') \in (\mathbf{l} \cdot v)} \pi_{u',v'} && (5.53)\\
&= \sum_{\mathbf{l} \in \mathcal{Y}(1,v)} \prod_{(u,v) \in \mathbf{l}} \pi_{u,v}, && (5.54)
\end{aligned}$$

as required. □

Lemma 5.3.10. *The KL divergence within the family $\mathcal{D}(\mathcal{R}, \cdot, \alpha)$ is*

$$\mathcal{D}_{\mathrm{KL}}\left[\mathcal{D}(\mathcal{R}, \mathbf{W}, \alpha) \middle\| \mathcal{D}(\mathcal{R}, \mathbf{W}^{(r)}, \alpha)\right] = \mu_N^{(r)} - \mu_N + \alpha \sum_{(u,v) \in \mathcal{E}} \omega_{u,v} \left(w_{u,v} - w_{u,v}^{(r)}\right), \tag{5.55}$$

where $\omega_{u,v}$ is the marginal probability of edge (u, v) on $\mathcal{D}(\mathcal{R}, \mathbf{W}, \alpha)$ defined in theorem 5.3.8, μ_N is defined in eq. (5.15) *and $\mu_N^{(r)}$ is similar to μ_N but defined in terms of $\mathbf{W}^{(r)}$ rather than $\mathbf{W}$.*

Proof. We have

$$\mathcal{D}_{\mathrm{KL}}\left[\mathcal{D}(\mathcal{R},\mathbf{W},\alpha)\middle\|\mathcal{D}(\mathcal{R},\mathbf{W}^{(r)},\alpha)\right] \tag{5.56}$$

$$= \mathbb{E}_{\mathbf{y}\sim\mathcal{D}(\mathcal{R},\mathbf{W},\alpha)}\left[\log\mathcal{D}(\mathbf{y}|\mathcal{R},\mathbf{W},\alpha) - \log\mathcal{D}(\mathbf{y}|\mathcal{R},\mathbf{W}^{(r)},\alpha)\right] \tag{5.57}$$

$$= \mathbb{E}_{\mathbf{y}\sim\mathcal{D}(\mathcal{R},\mathbf{W},\alpha)}\Big[\log\frac{\exp(\alpha\,\|\mathbf{y}\|_{\mathbf{W}})}{\sum_{\widehat{\mathbf{y}}\in\mathcal{Y}(1,N)}\exp(\alpha\,\|\widehat{\mathbf{y}}\|_{\mathbf{W}})} - \log\frac{\exp(\alpha\,\|\mathbf{y}\|_{\mathbf{W}^{(r)}})}{\sum_{\widehat{\mathbf{y}}\in\mathcal{Y}(1,N)}\exp(\alpha\,\|\widehat{\mathbf{y}}\|_{\mathbf{W}^{(r)}})}\Big]$$

$$= \mathbb{E}_{\mathbf{y}\sim\mathcal{D}(\mathcal{R},\mathbf{W},\alpha)}\Big[\alpha\,\|\mathbf{y}\|_{\mathbf{W}} - \alpha\,\|\mathbf{y}\|_{\mathbf{W}^{(r)}} + \tag{5.58}$$

$$\log\sum_{\widehat{\mathbf{y}}\in\mathcal{Y}(1,N)}\exp(\alpha\,\|\widehat{\mathbf{y}}\|_{\mathbf{W}^{(r)}}) - \log\sum_{\widehat{\mathbf{y}}\in\mathcal{Y}(1,N)}\exp(\alpha\,\|\widehat{\mathbf{y}}\|_{\mathbf{W}})\Big]. \tag{5.59}$$

The third and fourth terms inside the final expectation (on line (5.59)) are easily handled; *e.g.* for the third term we have

$$\mathbb{E}_{\mathbf{y}\sim\mathcal{D}(\mathcal{R},\mathbf{W},\alpha)}\left[\log\sum_{\widehat{\mathbf{y}}\in\mathcal{Y}(1,N)}\exp(\alpha\,\|\widehat{\mathbf{y}}\|_{\mathbf{W}^{(r)}})\right]$$

$$= \log\sum_{\widehat{\mathbf{y}}\in\mathcal{Y}(1,N)}\exp(\alpha\,\|\widehat{\mathbf{y}}\|_{\mathbf{W}^{(r)}}) \tag{5.60}$$

$$= \mu_N^{(r)}. \tag{5.61}$$

by the definition (5.15).

The first two terms inside the final expectation mentioned above (on line (5.58)) may also be efficiently re-factored; *e.g.* for the second term,

$$\mathbb{E}_{\mathbf{y}\sim\mathcal{D}(\mathcal{R},\mathbf{W},\alpha)}\left[\|\mathbf{y}\|_{\mathbf{W}^{(r)}}\right]$$

$$= \sum_{\mathbf{y}\in\mathcal{Y}(1,N)}\mathcal{D}(\mathbf{y}|\mathcal{R},\mathbf{W},\alpha)\sum_{(u,v)\in\mathbf{y}}w_{u,v}^{(r)}$$

$$= \sum_{\mathbf{y}\in\mathcal{Y}(1,N)}\prod_{(u,v)\in\mathbf{y}}\pi_{u,v}\sum_{(u,v)\in\mathbf{y}}w_{u,v}^{(r)}$$

$$= \sum_{(u,v)\in\mathcal{E}}w_{u,v}^{(r)}\underbrace{\sum_{\{\mathbf{y}\in\mathcal{Y}(1,N):(u,v)\in\mathbf{y}\}}\prod_{(u',v')\in\mathbf{y}}\pi_{u',v'}}_{\equiv\omega_{u,v}}.$$

The expectation of (5.58)-(5.59) may therefore be rewritten as (5.55). □

Corollary 5.3.11. *The KL divergence* (5.55)*, the likelihood* (5.38)*, and the sampling algorithm (theorem 5.3.7) may be computed in* $O(|\mathcal{E}|)$ *time.*

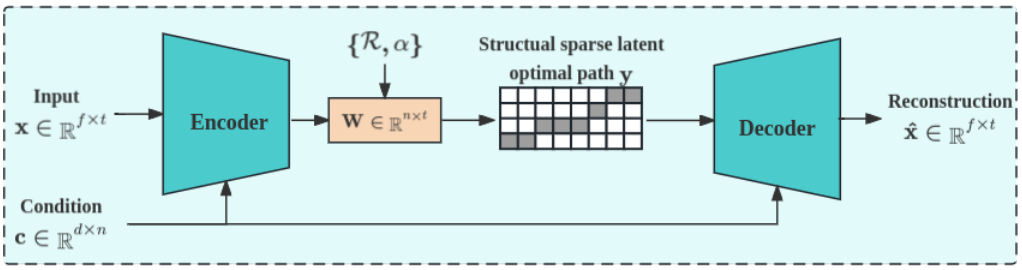


Figure 5.1: An overview pipeline of BDP-VAE. BDP-VAE captures the unobserved sparse structural dependency (i.e., optimal paths on a DAG) in the latent space in parallel training the model and allows gradient-based optimisation for learning the edge weights $\mathbf{W}$.

5.4 BDP-VAE

We now show how to apply the method in section 5.3 to a conditional VAE framework to obtain sparse latent optimal paths. An unconditional BDP-VAE framework can be directly applied based on theorem 5.3.6, theorem 5.3.7, and theorem 5.3.10, but a conditional BDP-VAE framework may be challenging in real applications. Given a sequential input $\mathbf{x}$ with length t and a sequential condition $\mathbf{c}$ with length n, where $t \neq n$ and $\{t, n\}$ are varying within the dataset, we wish to find an unobserved hard structural relationship between $\mathbf{x}$ and $\mathbf{c}$ in the latent space of VAEs denoted as $\mathbf{y}$. Conditional BDP-VAEs consist of three parts: an encoder models posterior distribution $q(\mathbf{y}|\mathbf{x}, \mathbf{c}; \phi)$, a decoder models the distribution of $p(\mathbf{x}|\mathbf{y}, \mathbf{c}; \theta)$, and a prior encoder models the prior distribution $p(\mathbf{y}|\mathbf{c}; \theta)$. An overview pipeline of BDP-VAE is in fig. 5.1, which captures structured sparse optimal paths in the latent space.

We assume the conditional input $\mathbf{c}$ is always observed and the conditional ELBO is defined as

$$\mathcal{L}(\phi, \theta, \mathbf{x}|\mathbf{c}) = \mathbb{E}_{\mathbf{y} \sim q(\cdot|\mathbf{x}, \mathbf{c}; \phi)} \left[\log p(\mathbf{x}|\mathbf{y}, \mathbf{c}; \theta)\right] - \mathcal{D}_{\mathrm{KL}} \left[q(\mathbf{y}|\mathbf{x}, \mathbf{c}; \phi) \middle\| p(\mathbf{y}|\mathbf{c}; \theta)\right]. \tag{5.62}$$

5.4.1 Posterior and Latent Optimal Paths

Given a DAG $\mathcal{R}$ with edge $\mathcal{E}$ and nodes $\mathcal{V}$, the distribution of the posterior encoder is denoted as

$$q(\mathbf{y}|\mathbf{x}, \mathbf{c}; \phi) = \mathcal{D}(\mathbf{y}|\mathcal{R}, \mathbf{W} = \mathrm{NN}_{\mathbf{W}}(\mathbf{x}, \mathbf{c}; \phi), \alpha), \tag{5.63}$$

where $\mathrm{NN}_{\mathbf{W}}(\cdot; \phi)$ is a neural network to learn the edge weights $\mathbf{W}$ of the DAG $\mathcal{R}$ and α is a hyper-parameter. The latent optimal path $\mathbf{y}$ with $\{0, 1\}$ can be sampled in reverse according to theorem 5.3.5 and theorem 5.3.7. As a result, $\mathbf{y}$ forms a *sparse matrix* with dimensions identical to those of the weight matrix $\mathbf{W}$. As shown in fig. 5.1, the sparsity and structure of $\mathbf{y}$ are inherently achieved through its construction using

the DAG $\mathcal{R}^2$.

5.4.2 Conditional Prior

Denote the distribution of the conditional prior as

$$p(\mathbf{y}|\mathbf{c};\theta) = \mathcal{D}(\mathbf{y}|\mathcal{R}, \mathbf{W}^{(0)} = \mathrm{NN}_{\mathbf{W}^{(0)}}(\mathbf{c};\theta), \alpha), \tag{5.64}$$

We have provided a closed-form KL divergence within the distribution family $\mathcal{D}(\mathcal{R}, \cdot, \alpha)$ in theorem 5.3.10 which may be convenient for unconditional generation by pre-setting the prior distribution statistics directly or has tractable $\mathbf{W}^{(0)}$.

In most conditional generation tasks, the non-accessible $\mathbf{x}$ during the inference phase in real applications leads to problems on the prior encoder when forming the edge weights $\mathbf{W}^{(0)}$ given information on $\mathbf{c}$ only, especially in the case that $\mathbf{x}$ has varying lengths t. To address this issue, we give a flexible solution for inferring feature information of $\mathbf{x}$ given $\mathbf{c}$ to form the edge weights $\mathbf{W}^{(0)}$ in the conditional prior. Inspired by Ma et al. [2019], we make use of a flow-based model[3] as the conditional prior to infer information about $\mathbf{x}$ condition on $\mathbf{c}$ and further obtain edge weights $\mathbf{W}^{(0)}$.

Assume there exists a series of invertible transformations of random variables $\mathbf{x}$, such that

$$\mathbf{x} \underset{g_1}{\overset{f_1}{\longleftrightarrow}} \cdots \underset{g_k}{\overset{f_k}{\longleftrightarrow}} \mathbf{c} \underset{g_{k+1}}{\overset{f_{k+1}}{\longleftrightarrow}} \cdots \underset{g_K}{\overset{f_K}{\longleftrightarrow}} v \tag{5.65}$$

where $f = f_1 \circ \cdots \circ f_K$ and $v \sim N(0,1)$ (θ is omitted for brevity), the KL divergence term in eq. (5.62) can be written as

$$\begin{aligned} &\mathcal{D}_{\mathrm{KL}}\left[q(\mathbf{y}|\mathbf{x},\mathbf{c};\phi)\middle\|p(\mathbf{y}|\mathbf{c};\theta)\right] &(5.66)\\ &= \mathcal{D}_{\mathrm{KL}}\left[q(\mathbf{y}|\mathbf{x},\mathbf{c};\phi)\middle\|p(\mathbf{y}|\mathbf{x},\mathbf{c};\theta)\right] - \log p(\mathbf{x}|\mathbf{c};\theta) \\ &= -\log p_{N(0,1)}(f_\theta(\mathbf{x}))|\det(\frac{\partial f_\theta(\mathbf{x})}{\partial \mathbf{x}})|. &(5.67)\end{aligned}$$

The backward pass is to infer the KL divergence during training. The forward pass is to infer feature information of $\mathbf{x}$ given $\mathbf{c}$ and form the edge weights $\mathbf{W}^{(0)}$ during inference.

[2]In the case where the structure of DAGs $\mathcal{R}$ is unknown, since we are learning the DAG weights $\mathbf{W}$ and zero weights are equivalent to removing an edge, in some sense BDP algorithm can in principle learn an approximate DAG structure by imposing small edge weight values. However, in this study, we target to verify our method on applications with a clear prior knowledge of DAG structure.

[3]The flow-based conditional prior strategy is proposed to solve the problem of forming edge weights $\mathbf{W}^{(0)}$ in case of $\mathbf{W}^{(0)}$ is intractable if $\mathbf{x}$ is non-accessible with varying lengths t. In practical applications, this strategy is not mandatory if the task has a known and fixed prior $\mathcal{D}(\mathcal{R}, \mathbf{W}^0, \alpha)$ or tractable $\mathbf{W}^{(0)}$ (i.e., theorem 5.3.10).

5.4.3 Learning

Based on the idea of Mohamed et al. [2020], the gradient of the ELBO (5.62) with respect to θ is straightforward, however, the gradient with respect to ϕ of the reconstruction error part in the ELBO is non-trivial. We make use of the REINFORCE estimator

$$\nabla_\phi \, \mathbb{E}_{\mathbf{y}\sim q(\cdot|\mathbf{x},\mathbf{c};\phi)} \left[\log p(\mathbf{x}|\mathbf{y},\mathbf{c};\theta)\right] = \log p(\mathbf{x}|\tilde{\mathbf{y}},\mathbf{c};\theta) \nabla_\phi \, \log q(\tilde{\mathbf{y}}|\mathbf{x},\mathbf{c};\phi), \tag{5.68}$$

where $\tilde{\mathbf{y}}$ is an exact sample via theorem 5.3.7 from the posterior $q(\cdot|\mathbf{x},\mathbf{c};\phi)$, and we recall that $\log q(\tilde{\mathbf{y}}|\mathbf{x},\mathbf{c};\phi)$ may be computed using the efficient and exact closed-form of eq. (5.38), and automatically differentiated.

Alternatively, we provide hints of the Gumbel softmax trick to avoid using the REINFORCE estimator in section 5.5 for interested readers. In this study, we focus on evaluating the closed form of latent sparse optimal paths. Compared to the reparameterization trick discussed in section 5.5, the log-derivative method (i.e., the REINFORCE estimator) has the advantage of being the most straight-forward, is formally unbiased, does not require the temperature parameter, and allows fast evaluation with the closed form expression of sparse paths $\mathbf{y}$ in eq. (5.38).

5.5 Gumbel Softmax Trick

In this section, we investigate a possible interpretation of achieving reparameterization trick (i.e., the Gumbel softmax trick) on BDP-VAE framework for interested readers. Compared to the log-derivative trick discussed in section 5.4.3, the Gumbel softmax trick samples soft latent optimal paths and has advantages on gradient-based optimisation. However, this method may require extra effort in designing its implementation and temperature parameter τ.

5.5.1 Node-wise Gumbel Softmax Propagation

Recall that the *Gumbel argmax* reparameterization for the categorical distribution employs parameter-independent Gumbels via

$$\forall i \in \{1,2,\ldots,m\}, \quad G_i \sim \mathcal{G}(0) \tag{5.69}$$

$$k = \underset{i\in\{1,2,\ldots,m\}}{\operatorname{argmax}} \; \log \mu_i + G_i, \tag{5.70}$$

which is equivalent to

$$k \sim \text{Categorical}(\beta_1, \beta_2, \ldots, \beta_m), \tag{5.71}$$

where the probability β_k is equal to the r.h.s. of (5.7).

The *Gumbel softmax* is a differentiable approximation to the above, where the

$k \in \{1, 2, \ldots, m\}$ of (5.70) is replaced by $\check{} \in \mathbb{R}^m$ given by

$$\kappa_k = \frac{\exp\left((\log \mu_k + G_k)/\tau\right)}{\sum_{i=1}^{m} \exp\left((\log \mu_i + G_i)/\tau\right)}, \tag{5.72}$$

where τ is a free parameter. This approximation is useful for gradient-based optimisation because it is both differentiable and parameterised.

Due to the generally intractable size $|\mathcal{Y}(1, N)|$ of the set of paths that make up the domain of $\mathcal{D}(\mathbf{y}|\mathcal{R}, \mathbf{W}, \alpha)$, there is no simple analogue of the above for our setting. Instead, we offer two alternative approaches, both of which are differentiable reparameterizations of a distribution of real values, one per node.

5.5.1.1 Marginal Node-wise Gumbel Softmax Distribution

Consider the following

Definition 5.5.1. Let $\mathbf{y} \sim \mathcal{D}(\mathcal{R}, \mathbf{W}, \alpha)$. The hitting probability for the node v is the probability that $\mathbf{y}$ includes v,

$$\zeta_v \equiv p(v \in \mathbf{y}). \tag{5.73}$$

The node-hitting probabilities may be efficiently obtained via

Lemma 5.5.2. *The ζ_u obey the recursion*

$$\zeta_N = 1 \tag{5.74}$$

$$\zeta_u = \sum_{v \in \mathcal{C}(u)} \zeta_v \, \pi_{u,v}, \tag{5.75}$$

for all $v \in \{N-1, N-2, \ldots, 1\}$ (in reverse topological order w.r.t. $\mathcal{R}$).

We may now simply associate with each node $u \in \mathcal{V}$ a Bernoulli random variable with probability parameter ζ_u, along with a Gumbel-softmax approximation of that Bernoulli — see section 5.5.3 for details.

5.5.1.2 Path-dependent Node-wise Gumbel Softmax Distribution

Alternatively, we can soften the path sampling algorithm of theorem 5.3.7 to obtain a distribution over $\{\gamma_u \in [0, 1]\}_{u \in \mathcal{V}}$ that better captures the dependence between the events $u \in \mathbf{y}$ for all $u \in \mathcal{V}$. That is, we let

$$\gamma_N = 1 \tag{5.76}$$

$$\gamma_u = \sum_{v \in \mathcal{C}(u)} \gamma_v \, \delta_{u,v}, \tag{5.77}$$

for all $u \in \{N-1, N-2, \ldots, 1\}$ (in reverse topological order w.r.t. $\mathcal{R}$), where $\delta_{u,v}$ is the Gumbel-softmax analog to step 2 of theorem 5.3.7, namely

$$\forall (u,v) \in \mathcal{E},\ G_{u,v} \sim \mathcal{G}(0) \tag{5.78}$$

$$\delta_{u,v} = \frac{\exp\left((\log \pi_{u,v} + G_{u,v})/\tau\right)}{\sum_{i \in \mathcal{C}(u)} \exp\left((\log \pi_{u,i} + G_{u,i})/\tau\right)}. \tag{5.79}$$

5.5.2 KL Divergence Between two Gumbels

Lemma 5.5.3. *The KL divergence between unit variance Gumbels is*

$$\mathcal{D}_{\mathrm{KL}}\left[\mathcal{G}(\alpha) \middle\| \mathcal{G}(\beta)\right] = \alpha - \beta + (1 - \exp(\alpha - \beta))\,\mathrm{Ei}(-\exp(-\alpha)), \tag{5.80}$$

in terms of the standard special "exponential integral" function

$$\mathrm{Ei}(z) \equiv -\int_{-z}^{\infty} \exp(-t)/t \,\mathrm{d}t. \tag{5.81}$$

Proof.

$$\begin{aligned}
&\mathcal{D}_{\mathrm{KL}}\left[\mathcal{G}(\alpha) \middle\| \mathcal{G}(\beta)\right] \\
&= \int_0^\infty G(x|\alpha) \log \frac{G(x|\alpha)}{G(x|\beta)} \,\mathrm{d}x \\
&= \int_0^\infty \exp(-(x-\alpha) - \exp(x-\alpha)) \log \frac{\exp(-(x-\alpha) - \exp(x-\alpha))}{\exp(-(x-\beta) - \exp(x-\beta))} \,\mathrm{d}x \\
&= \int_0^\infty \exp(-(x-\alpha) - \exp(x-\alpha))(\alpha - \beta + \exp(x-\beta) - \exp(x-\alpha)) \,\mathrm{d}x \\
&= \alpha - \beta + I_1 - I_2 \\
&= \alpha - \beta + (1 - \exp(\alpha - \beta))\,\mathrm{Ei}(-\exp(-\alpha)),
\end{aligned}$$

because

$$\begin{aligned}
I_2 &\equiv \int_0^\infty \exp(-\exp(x-\alpha)) \,\mathrm{d}x \\
&= -\,\mathrm{Ei}(-\exp(-\alpha))
\end{aligned}$$

and

$$\begin{aligned}
I_1 &\equiv \int_0^\infty \exp(\alpha - \beta - \exp(x-\alpha)) \,\mathrm{d}x \\
&= -\exp(\alpha - \beta)\,\mathrm{Ei}(-\exp(-\alpha)),
\end{aligned}$$

giving the desired result. □

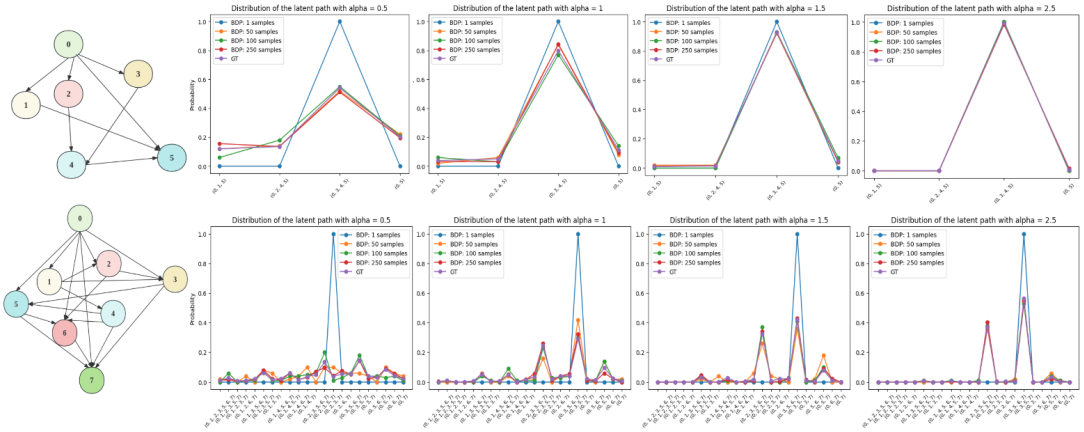


Figure 5.2: Toy experiments on BDP to find stochastic optimal paths under randomly generated DAGs. *The first row is a 5-node DAG and its density plots with different α value. The second row is an 8-node DAG and its density plots with different α value.*

5.5.3 Binary Gumbel (Soft) Max

For the binary case, we can sample just one logistic random variable, rather than two Gumbels, as per the traditional Gumbel max trick. Let $X \sim \text{Bernoulli}(\zeta)$. The Gumbel max parameterised, for $g_1, g_2 \sim \mathcal{G}(0)$

$$X = \underset{k\in\{1,2\}}{\text{argmax}} \left(\log\zeta + g_1, \log(1-\zeta) + g_2\right)_k \tag{5.82}$$

$$= \underset{k\in\{1,2\}}{\text{argmax}} \left(\log\zeta, \log(1-\zeta) + l\right)_g bk, \tag{5.83}$$

where we may show that $l = g_2 - g_1 \sim \text{Logistic}(0,1)$.[4] The soft-max analogue $X_\tau \in (0,1)$ of X, where τ is a temperature parameter, is therefore

$$X_\tau \equiv \frac{\exp(\tau^{-1}\log\zeta)}{\exp(\tau^{-1}\log\zeta) + \exp\left(\tau^{-1}\left(\log(1-\zeta)+l\right)\right)}. \tag{5.84}$$

5.6 Experiments and Results

In this section, we conduct four experiments to verify our methods from section 5.3 and section 5.4, and show its applicability on two speech synthesis applications. To show the generalization of the proposed method in section 5.3, we extend BDP to two common application examples of computational graphs: monotonic alignment (MA) [Kim et al., 2020] and dynamic time warping (DTW) [Sakoe and Chiba, 1978; Mensch and Blondel, 2018], which is commonly used in speech synthesis tasks.

[4] https://en.wikipedia.org/wiki/Logistic_distribution

Table 5.1: Mel Cepstral Distortion (MCD) and Real-Time Factor (RTF) compared with other TTS models.

Model	Training	Align. (Train)	Align. (Infer)	MCD	RTF
FastSpeech2 (Baseline)	Non end-to-end	Discrete	Continuous	9.96 ± 1.01	3.87×10^{-4}
Tacotron2	end-to-end	Continuous	Continuous	11.39 ± 1.95	6.07×10^{-4}
VAENAR-TTS	end-to-end	Continuous	Continuous	8.18 ± 0.87	1.10×10^{-4}
Glow-TTS	end-to-end	Discrete	Continuous	8.58 ± 0.89	2.87×10^{-4}
BDPVAE-TTS (ours)	end-to-end	Discrete	Discrete	8.49 ± 0.96	3.00×10^{-4}

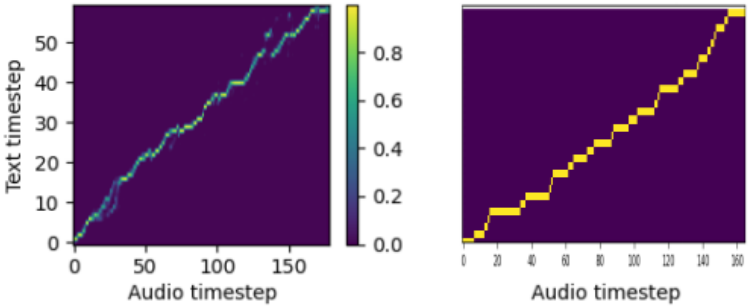


Figure 5.3: An example of continuous phoneme-duration alignment vs discrete phoneme-duration alignment. *Continuous alignment results in disconnected and imprecise phoneme boundaries, whereas discrete alignment provides clean and consistent phoneme durations.*

5.6.1 Stochastic Optimal Paths on Toy DAGs

We conduct an experiment to demonstrate how BDP in section 5.3 finds the optimal paths on toy DAGs, in which the DAG structures and edge values are randomly generated. In fig. 5.2, we plot approximated density plots of path distributions with 1, 50, 100, and 250 samples by BDP compared with the corresponding ground truth path density plot. As BDP samples increase, the distribution of path samples will eventually converge to the real path distribution.

We then studied how the value of the temperature parameter α affects the path distribution. The larger α is, the sharper the distribution is, leading to an accurate optimal path result with less sample time. Conversely, for too small α, BDP needs more samples to obtain the optimal path. We conduct another experiment to connect this finding with BDP-VAE in section 5.6.5.

5.6.2 Application 1: End-to-end Text-to-Speech

We apply the BDP-VAE framework with the computational graph of MA to perform an end-to-end TTS model on the RyanSpeech [Zandie et al., 2021] dataset. RyanSpeech contains 11279 audio clips (10 hours) of a professional male voice actor's speech recorded at 44.1kHz sampling rates. We randomly split 2000 clips for validation and 9297 clips for training.

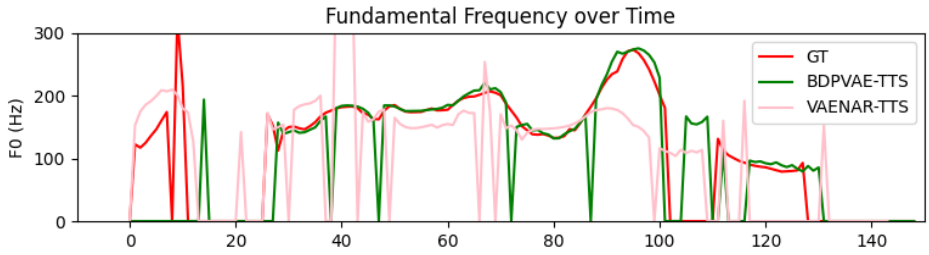


Figure 5.4: Inference F0 trajectory comparison with VAENAR-TTS of utterance "I suppose I have many thoughts.". *The intonation of BDPVAE-TTS is close to the GT indicating that sparse optimal paths help the decoder with a better understanding of how each phoneme contributes to the overall utterance with approximated durations.*

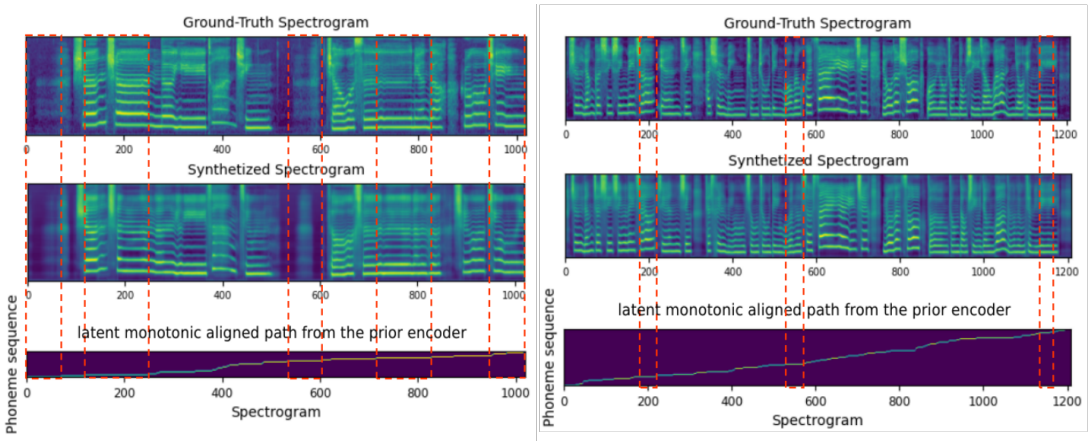


Figure 5.5: Visualization of GT, synthesized singing voice spectrogram, and latent optimal path from the prior encoder. *The GT and generated spectrogram are almost identical, and the generated spectrogram has a similar temporal structure to the inferred latent optimal path.*

The task of TTS involves an unobserved monotonic hard alignment that maps phonemes to time intervals, since the order of the phonemes should be preserved while the duration of each may vary in speech. Thus, TTS models generate speech according to corresponding phonemes in which the duration of each phoneme is discrete, structural, and unobserved Mehrish et al. [2023]. Existing end-to-end TTS methods use a soft approximation such as attention to obtain continuous alignment. As illustrated in fig. 5.3 compared to discrete phoneme-duration alignment, the softening solution may easily obtain disconnected alignment. To show the BDP-VAE framework can be easily adapted to down-stream tasks, we redesigned the popular non-end-to-end TTS model FastSpeech2 [Ren et al., 2020] into the BDP-VAE framework, namely BDPVAE-TTS, which can capture unobserved hard monotonic dependencies between phonemes and utterances jointly on both training and inference.

We verify model performance by an objective metric, the Mel cepstral distortion (MCD) [Kubichek, 1993; Chen et al., 2022a], between ground truths and synthesized outputs. We record inference speed by real-time factor (RTF) per generated spectrogram frame. We randomly pick 70 sentences from the test set, the numerical results

are shown in table 5.1. Our method outperforms the baseline on both MCD and RTF and achieves end-to-end training. This shows the success of BDP-VAE in adapting a non-end-to-end model that relies on external DP aligners into an end-to-end pipeline.

We make a comparison with other end-to-end TTS models Shen et al. [2018a]; Kim et al. [2020]; Lu et al. [2021] which target capturing phoneme dependencies. Shen et al. [2018a] models the unobserved temporal dependencies by an auto-regressive architecture with attention. Kim et al. [2020] integrate a monotonic alignment search in parallel in a Glow model Kingma and Dhariwal [2018] to obtain hard monotonic alignment. Lu et al. [2021] utilizes the latent Gaussian and captures soft monotonic alignment in the decoder by causality-masked self-attention. Among them, BDPVAE-TTS performs discrete monotonic alignment on both training and inference that ensures model consistency for training and inference. BDPVAE-TTS gets a better MCD and RTF than Shen et al. [2018a] and Kim et al. [2020], but gets higher MCD and RTF than Lu et al. [2021].

For the RTF, since BDPVAE-TTS involves a linear time consumption DP algorithm which causes higher inference time than VAENAR-TTS. We have discussed time complexity for capturing phoneme-to-frame dependencies during training for each end-to-end TTS model in this experiment below. However, the linear time consumption (theorem 5.3.11) is the best we can do for solving DP problem.

Time Complexity Analysis: Denote the maximum number of spectrogram frames in the computational graph of MA is T_{mel} and the maximum number of phoneme tokens in the computational graph of MA is T_{text}. In this experiment, BDPVAE-TTS obtains discrete latent monotonic paths by BDP with $\mathcal{O}(T_{mel})$ time complexity with diagonal updating strategy. Besides, the time complexity of monotonic alignment search (MAS) in Glow-TTS is $\mathcal{O}(T_{text} \times T_{mel})$ [Kim et al., 2020], the time complexity of self-attention in VAENAR-TTS is $\mathcal{O}(1)$ [Vaswani et al., 2017], and the time complexity of the auto-regressive TTS model (i.e., Tacotron2 [Shen et al., 2018a]) is $\mathcal{O}(T_{mel})$.

For the MCD, even though BDPVAE-TTS obtains a higher MCD than VAENAR-TTS, BDPVAE-TTS captures *discrete monotonic aligned paths* in its latent space on training and inference phase which ensures model's train and test consistency and improves the model's interpretability. VAENAR-TTS compresses the learned feature of spectrograms with conditions into Gaussian latent variables and uses these in its decoder for reconstruction. During the decoding process, these latent variables are used alongside phonemes to reconstruct the outputs. In contrast, the latent variables in BDPVAE-TTS are designed to capture phoneme-duration dependencies. This focus provides the decoder with a nuanced understanding of how the phoneme contributes to the overall speech utterance, which can be interpreted by the inference fundamental frequency (F0) in fig. 5.4.

5.6.3 Application 2: End-to-end Singing Voice Synthesis

We extend BDP-VAE with MA for end-to-end SVS on the popcs dataset [Liu et al., 2022]. We perform this experiment not to compare with other models as in section 5.6.2, but to demonstrate the utility of our method in a related task. In SVS, the

Table 5.2: Mean absolute error and standard deviation (MAE/MAE±STD) of phoneme duration on TIMIT.

Model	**Distribution**	**Train**	**Inference**
BDP-VAE	$\mathcal{D}(\mathcal{R}_{\text{DTW}}, \text{NN}_{\{\phi/\theta\}}, 1)$	2.92	3.93 ± 0.37
Baseline	$\mathcal{D}(\mathcal{R}_{\text{DTW}}, \mathbf{W}_{U(0,1)}, 1)$	5.67	5.69 ± 0.31

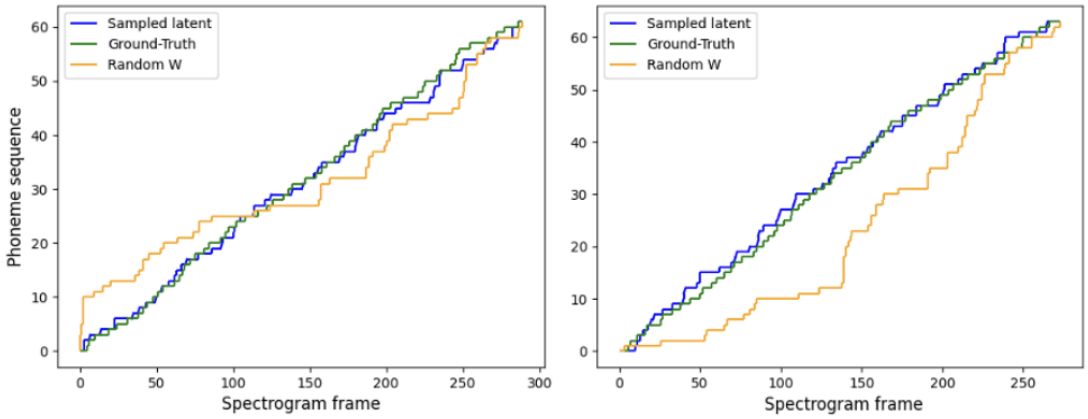


Figure 5.6: Visualization of GT alignment between phoneme tokens and spectrogram frames, latent optimal paths from the encoder, optimal paths from random latent space for two audio clips. *BDP-VAE achieves closer alignments with GT, indicating its effectiveness in finding latent optimal paths.*

longer phoneme duration also provides an opportunity to visualize the monotonically aligned optimal path clearly. The popcs contains 117 Chinese Mandarin pop songs (5 hours) collected from a qualified female vocalist. We randomly split 50 clips for inference and the rest for training. During the inference phase, the conditional inputs are the fundamental frequency and lyrics of the song clips.

Similar to TTS task, SVS task also involves unobserved structural dependencies. BDP-VAE framework could help the SVS task to capture the discrete structural dependencies in parallel, leading to an end to end framework and a better model interpretability.

Two inference results are visualized in fig. 5.5 where red rectangles indicate that the temporal structure between the generated mel-spectrogram and ground truth is almost identical. As expected, the decoder of BDP-VAE synthesizes spectrograms according to the extended conditions (i.e., the phonemes) mapping by the dependency of the latent monotonic path from the prior encoder. Therefore, the decoder synthesizes singing voice according to latent path-aligned phoneme conditions.

5.6.4 Verification of Behaviour of Latent Optimal Path in BDP-VAE

To verify the behaviour and generalization of the latent optimal paths in BDP-VAE, we obtain latent optimal paths under the computational graph of DTW on the TIMIT

Table 5.3: Mean absolute error and standard deviation (MAE / MAE $\pm$ STD) for the duration on TIMIT with different hyper-parameter α settings.

Model	**Alpha**	**Train**	**Inference**
BDP-VAE	0.5	2.99	4.40 ± 0.46
BDP-VAE	1	2.92	3.93 ± 0.37
BDP-VAE	1.5	2.94	3.89 ± 0.35
BDP-VAE	2	3.01	4.08 ± 0.27
BDP-VAE	3	3.25	4.53 ± 0.10

speech corpus dataset [Garofolo et al., 1992] which includes manually time-aligned phonetic and word transcriptions. TIMIT contains English speech of 630 speakers which utterances are recorded as 16-bit 16kHz speech waveform files. We randomly split 50 clips for testing and 580 clips for training.

To the best of our knowledge, there are no existing similar works that enable VAEs to obtain hard latent optimal paths given any defined DAG structures. Thus, inspired by the experimental designs in Mensch and Blondel [2018] and van den Oord et al. [2017b], we set a latent distribution $\mathcal{D}(\mathcal{R}_{\text{DTW}}, \mathbf{W} = \{w \in \mathbf{W} \sim U(0,1)\}, 1)$ as the baseline due to lack of direct comparison. The random baseline still follows the DTW constraint with uniform the edge weights, which can verify the behaviour of latent space in BPD-VAE.

We take a summation along the spectrogram dimension to obtain phoneme duration and compute the mean absolute error (MAE) between the ground truth and the phoneme duration. We input phoneme tokens and spectrogram lengths as conditions on inference to obtain latent optimal paths of the prior encoder. We repeat the inference 5 times and take the average and standard deviation of MAEs as our metric of evaluation. As shown in table 5.2, our MAE is lower than the baseline indicating that BDP-VAE captures meaningful information to obtain stochastic optimal paths in the latent space and is not merely guessing randomly. fig. 5.6 visualizes ground-truth alignments, latent optimal paths from BDP-VAE, and optimal paths from the baseline latent distribution on two audio clips, which clearly shows the latent optimal path from BDP-VAE under DTW gets a close alignment to the GT. This indicates BDP-VAE has the ability to learn informative edge weights $\mathbf{W}$ and capture sparse optimal paths in latent space.

5.6.5 Hyper-parameter Sensitivity in BDP-VAE

To study the sensitivity of the temperature parameter α, we extend the experiment in section 5.6.4. table 5.3 shows the MAE value on the training and inference phase with different α settings. As discussed in section 5.6.1, the α affects the sharpness of the path distribution. When α is smaller, the distribution is more stochastic. As α increases, the distribution becomes sharper. However, when the α value becomes large, the latent path alignment performance decreases, which is consistent with the contribution of temperature in Platt [2000]. In BDP-VAE, the model with too large α may not capture enough variety of data, conversely, for too small α the model becomes too

spread out and lacks meaningful structure, with the distribution approaching uniform as $\alpha \to 0$. In real applications, the value of α in BDP-VAE should be located in a reasonable range which could be found by tuning or setting a learnable parameter.

5.6.6 Limitations

As a discrete latent variable model, BDP-VAE uses the REINFORCE estimator, which may lead to high gradient variance and slow convergence during training. This limitation may be solved by involving variance reduction techniques during training.

5.7 Summary

In this chapter, we introduced a probabilistic softening solution to the classical optimal path problem on DAGs, as stochastic optimal paths, aiming to improve the training efficiency of multiple training strategies into an end-to-end training pipeline in the speech synthesis domain. To achieve variational Bayesian inference with latent paths, we give efficient and tractable algorithms for sampling, likelihood, and KL divergence within the family of path distributions in linear time with respect to edge numbers by dynamic programming with properties of the Gumbel distribution as message-passing, namely Bayesian dynamic programming with Gumbel propagation. We demonstrate the usage of stochastic optimal paths in the VAE framework and propose BDP-VAE. BDP-VAE captures sparse optimal paths as latent representations given a DAG and further achieves end-to-end training for downstream generative tasks that rely on unobserved structural relationships. We showed how BDP finds stochastic optimal paths under general small toy DAGs and demonstrated BDP-VAE with the computational graph of monotonic alignment on two real-world applications to achieve an end-to-end framework. We verified the behavior and generalization of the latent optimal paths under the computational graph of dynamic time warping. We also studied the sensitivity of the hyperparameter α and gave suggestions for real applications. Our experiments show the success of our approach on generative tasks where it achieves end-to-end training involving unobserved sparse structural optimal paths. Beyond VAEs, BDP can potentially be integrated into other probabilistic frameworks or planning in model-based reinforcement learning to obtain latent paths that leave room for future extensions.

Statement of Authorship

This chapter is based on the following publication:

Niu, X., Walder, C., Zhang, J., & Martin, C. P. (2024, July). Latent Optimal Paths by Gumbel Propagation for Variational Bayesian Dynamic Programming. In International Conference on Machine Learning (pp. 38316-38343). PMLR.

Contributions of Candidate

I conceived the research idea, designed the proposed BDP and BDP-VAE framework, implemented the model, conducted all experiments, analyzed the results, and prepared the manuscript.

Contributions of Co-authors

Dr. Christian Walder provided technical guidance on the theoretical development of the BDP framework, offered valuable feedback throughout the research, and contributed to revising the manuscript. Dr. Charles Patrick Martin and Dr. Jing Zhang supervised the research, provided technical discussions and feedback on the methodology, and contributed to the manuscript revision.

Chapter 6

HybridVC: Efficient Voice Style Conversion with Text and Audio Prompts

In chapter 5, we explored a theoretical perspective to improve the training efficiency and naturalness of text-to-speech (TTS) synthesis. While TTS has achieved remarkable progress in generating intelligible and natural speech from transcripts, it typically provides limited control over prosody and speaker characteristics. For example, generating speech with a specific emotional tone, timbre, or prosodic pattern cannot be easily achieved by conditioning only on text. This limitation motivates the task of voice conversion (VC), which can be viewed as a complementary direction within the speech synthesis domain. Unlike TTS, which maps text to speech, VC focuses on modifying an existing speech signal to change its vocal identity, prosody, or style while preserving the underlying linguistic content. In this way, VC provides the fine-grained control over prosody and voice style that TTS alone cannot easily achieve, making the two tasks closely related yet distinct components of the broader speech synthesis landscape.

In this chapter, we introduce HybridVC, an efficient voice conversion framework that combines the strengths of latent modeling and contrastive learning. Built upon a pre-trained conditional variational autoencoder (CVAE), HybridVC supports both text prompts and audio prompts, enabling flexible and precise voice style conversion. The framework models a latent distribution conditioned on speaker embeddings from a pretrained speaker encoder, while simultaneously optimizing style text embeddings through contrastive learning to align with speaker style information. This design allows HybridVC to achieve strong performance while maintaining training efficiency under limited computational resources. Our experiments demonstrate HybridVC's capability for advanced multi-modal voice style conversion, underscoring its potential for real-world applications such as personalized voice customization in social media and interactive platforms[1].

[1]Demonstration page:https://xinleiniu.github.io/HybridVC-demo/

6.1 Introduction

Voice conversion (VC) entails the process of modifying the vocal characteristics of a source speech to align with those of a target speaker. The key objective of VC is to alter the vocal identity, such as accent, tone, and timbre, to resemble the target speaker's voice style, while meticulously retaining the linguistic content of the source speech [Mohammadi and Kain, 2017; Zhou et al., 2021; Wang et al., 2018].

Traditional VC techniques rely on parallel training data [Toda et al., 2007; Desai et al., 2009; Sun et al., 2015; Zhang et al., 2019], which constrains their applicability for adapting to the voice styles of unseen speakers. To diversify the utilization of VC models, recent works focus on non-parallel any-to-any (A2A) VC models, so-called one-shot VC, that convert any speech to any voice style according to a few seconds of the target voice [Qian et al., 2019]. The key idea of A2A VC is learning disentangled feature representations of content and speaker style identity. To disentangle content and speaker information, existing works focus on methods including instance normalization [Park et al., 2023], vector quantization [Wu and Lee, 2020; Wu et al., 2020; Wang et al., 2021], transfer learning from ASR and TTS models [Zhang et al., 2023; Li et al., 2023b; Hussain et al., 2023; Rekimoto, 2023], and adversarial training [Park et al., 2023; Li et al., 2023a; Li and Wei, 2022].

With the development of large language models (LLMs), leveraging natural language prompts for generating images [Gu et al., 2022b], sound [Yang et al., 2023c; Liu et al., 2023a; Huang et al., 2023d; Niu et al., 2024d], and speech [Wang et al., 2023b; Yang et al., 2023a] has gained popularity. A notable development in VC is the introduction of voice style transfer using text prompts, as proposed by Gölge and Davis [2022]. Building on this concept, Kuan et al. [2023] introduced Text-guidedVC, diverging from traditional VC techniques using natural language instructions, rather than an audio reference, to guide the VC process. This shift enhances the framework's flexibility and allows for a wider range of conversion styles by incorporating descriptors in natural language. Concurrently, PromptVC [Yao et al., 2023] proposed an A2A text prompt-based VC model, following VITS [Kim et al., 2021] structure, that employs a latent diffusion model (LDM) [Rombach et al., 2022a] to generate style vectors by text prompts.

We observe two limitations in text-guided VC models [Kuan et al., 2023; Yao et al., 2023]. Firstly, it relies on parallel training data, where effectiveness is heavily dependent on the size and quality of the dataset. This requirement can pose challenges, as accumulating extensive parallel data is often resource-intensive. Secondly, unlike conventional VC models that use audio reference, text-guided VC may struggle with achieving precise voice conversions. In practical scenarios, text descriptions may fall short in accurately capturing and conveying specific voice styles, a gap that is readily bridged when using actual target voice samples. This limitation is crucial, as it affects the framework's ability to replicate exact voice styles based solely on text guidance. Therefore, an A2A VC model with hybrid prompts can be developed to combine the strengths of flexibility of text guidance and the precision of voice guidance.

VC models [Casanova et al., 2022; Yao et al., 2023; Li et al., 2023a] based on

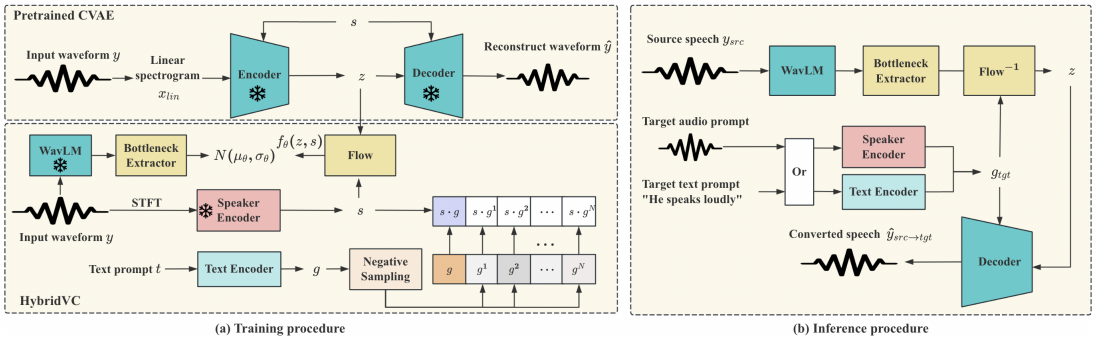


Figure 6.1: Overview of HybridVC, where z represents the latent variable obtained by CVAE backbone, g is text embeddings, s is speaker style embeddings, and $\{g^1, ..., g^N\}$ are N negative samples. The snowflake symbol represents frozen parameters during training.

VITS [Kim et al., 2021] framework may suffer from slow training due to decoding raw waveforms directly. A straightforward solution is to avoid the decoding step during training but maintain its advantages on its performance during inference. PromptVC [Yao et al., 2023] trains an LDM jointly with a VITS framework to learn textual information. Although Yao et al. [2023] doesn't provide experimental setup details, jointly training an LDM and a CVAE with adversarial training requires a large volume of training data, powerful computational resources, and long training time that potentially limits its extension to practical applications. In image editing tasks, Imagic [Kawar et al., 2023] proposes text embedding optimisation, which focuses on optimizing text embeddings under pre-trained language models. This method offers a more flexible solution that avoids retraining the entire model, thereby providing an easier and more efficient way to extend the model through few-shot learning.

In this work, we introduce HybridVC, an efficient A2A VC model that supports text prompts and audio prompts of target style voice to achieve better flexibility. HybridVC is a latent model with contrastive learning, which models the latent distribution conditioned on speaker style information and optimises the alignment of corresponding text embedding in parallel. Our experiments prove HybridVC achieves significant training efficiency under limited computational resources and achieves a flexible VC system that supports hybrid prompts. HybridVC supports small-scale training which can be easily adapted to applications such as user-defined personalized voice.

6.2 Method

In this section, we introduce our method, HybridVC, an A2A voice conversion model that supports text and audio prompts.

6.2.1 HybridVC

HybridVC, shown in fig. 6.1, is a conditional latent model that relies on a pre-trained CVAE backbone. We denote a text-speech pair (y, t), where y is an utterance and t is the corresponding style text description. Similarly to LDM [Rombach et al., 2022a], the first stage of HybridVC obtains a compact latent representation z from the pre-trained CVAE conditioned on speaker style information s by a pre-trained speaker encoder, as $z \sim q(z|x_{lin}, s)$. HybridVC models the distribution $p_\theta(z|y, s)$ conditioned on speaker embedding s and raw waveform y. Following Li et al. [2023a], the architecture of HybridVC is straightforward, using a pre-trained WavLM [Chen et al., 2022b] and a bottleneck extractor[2] to disentangle text-free content information c from waveform y, a pre-trained speaker encoder to obtain speaker style information s, and a normalizing flow conditioned on s to transform the distribution to a more complex distribution. This method enables HybridVC to encapsulate the content c and speaker style information s within a compact latent space, facilitating efficient manipulation and generation of speech.

To enable HybridVC to support audio and text prompts, the model incorporates a text encoder to optimise a text embedding g through contrastive learning, as detailed in section 6.2.2. The contrastive learning fine-tunes the text encoder, thereby, refining text embedding g to be well-aligned with speaker style embedding s obtained by the pre-trained speaker encoder.

6.2.1.1 CVAE backbone

The CVAE backbone compresses the input waveform y into a latent variable z conditioned on a speaker-style embedding s. In our experiments, we use the posterior encoder and decoder of a pre-trained FreeVC [Li et al., 2023a] as the CVAE backbone which is augmented with GAN training.

6.2.1.2 Speaker Encoder

The speaker encoder extracts the vocal identity (i.e., speaker style embedding s) from y. We follow FreeVC [Li et al., 2023a] that adopt a pre-trained verification model [Liu et al., 2021] as speaker encoder. The model is trained on a large number of speakers. The speaker encoder embeds mel-spectrogram of waveform y into a speaker style embedding s, as $s \in \mathbb{R}^{256}$.

6.2.1.3 Text Encoder

The text encoder extracts stylistic features from text prompt t. We fine-tune a pre-trained text encoder to refine the text embeddings g, ensuring they are closely aligned with the corresponding speaker style information s. The text encoder contains a

[2]The bottleneck extractor is a projection layer used to reduce feature dimension and further disentangle content information c from outputs of WavLM [Chen et al., 2022b]. We use the same setting as the one in Li et al. [2023a].

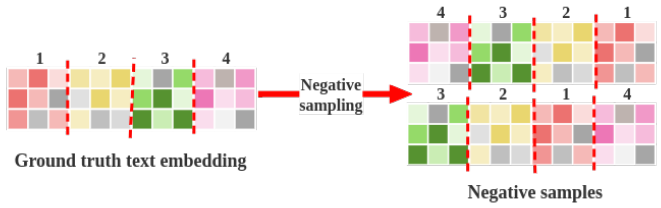


Figure 6.2: Illustration of negative sampling for text embedding.

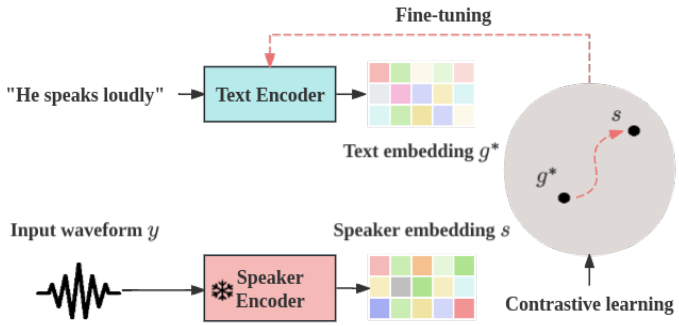


Figure 6.3: Illustration of optimizing the text embedding g^*.

RoBERTa [Liu et al., 2020] model from a pretrained CLAP [Wu et al., 2023a] that encodes a text prompt t into an embedding $g \in \mathbb{R}^{512}$, and a linear layer further compresses g to a lower feature dimension $g \in \mathbb{R}^{256}$ to discard irrelevant textual information.

6.2.2 PromptSpeech + Text Embedding optimisation

We use the PromptSpeech dataset [Guo et al., 2023], which pairs over 26,000 real speech samples from LibriTTS [Zen et al., 2019] with free-text style prompts involving categories of gender, pitch, energy, and speed. However, text-style prompts in PromptSpeech do not guarantee comprehensive coverage of information across these four categories. For instance, a prompt like "he said loudly" specifies gender and energy but not pitch or speed. This inconsistency presents a challenge to ensure comprehensive representations of all stylistic attributes, particularly when attempting to precisely map these free-form textual prompts to a predefined set of labels for speaker embedding s. To address this, we augment the data with N negative samples of text embeddings g extracted by the text encoder. Inspired by Zhu et al. [2022]; Lüddecke and Ecker [2022], we propose a negative sampling method for text embeddings g that randomly shuffle g into N negative samples $G' = \{g^1, ..., g^N\}$ as in fig. 6.2. Given the ground truth text embedding g, we construct G' by dividing g into K randomly ordered chunks. As illustrated in fig. 6.3, HybridVC fine-tunes the text encoder through contrastive learning to optimise the text embedding.

Table 6.1: Model comparison was measured on unseen source speeches with audio prompts from 9 unseen speakers, where h stands for hours and d stands for days.

Model	Training Config.	Dataset	WER ↓	CER ↓	F_0-PCC ↑	SSIM ↑	FAD↓	FD ↓	NISQA ↑
FreeVC	NVIDIA RTX3090	VCTK	13.56%	4.77%	0.719	0.781	1.15	0.71	4.51
FreeVC ⋆	2 NVIDIA RTX3090 (12d)	VCTK	**13.16**%	**4.60**%	**0.720**	0.771	0.89	0.71	4.34
YourTTS-VC	NVIDIA TESLA V100	VCTK	28.50%	12.45%	0.637	0.662	3.99	1.37	3.88
HybridVC ⋄	2 NVIDIA RTX3090 (15h)	VCTK	14.01%	4.85%	0.716	**0.783**	0.87	**0.70**	4.51
HybridVC ⋄	2 NVIDIA RTX3090 (17h)	PromptSpeech	18.18%	6.15%	0.701	0.771	**0.74**	0.81	4.51
HybridVC	2 NVIDIA RTX3090 (25h)	PromptSpeech	17.54%	6.17%	0.712	0.775	0.79	0.75	**4.52**

6.2.3 Training

As shown in fig. 6.1(a), the loss function of HybridVC contains two parts: a KL-divergence between the latent distributions of HybridVC and CVAE, and a contrastive loss between the text embedding with negative samples $\{g, G'\}$ and speaker embedding s. Following Kim et al. [2021], the KL loss $\mathcal{L}_{kl}$ between a Gaussian and normalising flow optimises the latent distribution of HybridVC,

$$\mathcal{L}_{kl} = \log q(z|x_{lin}, s) - \log p_\theta(z|y, s) \tag{6.1}$$

where f is the normalising flow, x_{lin} is the linear spectrogram,

$$p_\theta(z|y, s) = N(f_\theta(z, s); \mu_\theta, \sigma_\theta)|\frac{\partial f_\theta(z, s)}{\partial z}| \tag{6.2}$$

$$q(z|x_{lin}, s) = N(z; \mu_q, \sigma_q) \tag{6.3}$$

The contrastive loss $\mathcal{L}_{contrastive}$ is used to optimise the text encoder aligning the text embedding g with the speaker embedding s. Given a set of negative samples $G' = \{g^1, ..., g^N\}$, the contrastive loss is defined as

$$\mathcal{L}_{contrastive} = \log\frac{exp(cos(g, s)/\tau)}{\sum_{g_i \in \{g, G'\}} exp(cos(g_i, s)/\tau)} \tag{6.4}$$

where τ is a learnable temperature parameter.

The overall loss function $\mathcal{L}$ in HybridVC is defined as

$$\mathcal{L} = (1 - \alpha)\mathcal{L}_{kl} + \alpha\mathcal{L}_{contrastive} \tag{6.5}$$

where α is a weight-scaling hyper-parameter.

6.2.4 Inference

As shown in fig. 6.1(b), given a source speech y_{src}, and a prompt embedding g_{tgt} obtained from either the speaker encoder or the text encoder, the inference phase of

HybridVC is

$$\hat{y}_{src \to tgt} = Decoder(z, g_{tgt}) \tag{6.6}$$

where $z \sim p_\theta(z|y_{src}, g_{tgt})$.

6.3 Experiments and Results

6.3.1 Datasets

We conduct experiments on VCTK [Yamagishi et al., 2019] and PromptSpeech [Guo et al., 2023] datasets. PromptSpeech includes over 26000 real utterances from LibriTTS [Zen et al., 2019] with corresponding text style prompts, it is used for training HybridVC only. For VCTK, we follow the train/test separation as in Li et al. [2023a] which involves 107 speakers in total, 314 utterances for validation, 1070 utterances for testing and the rest for training[3]. VCTK is used to train HybridVC for a model comparison experiment. The VCTK testing set is used to verify models' performance as source speeches.

6.3.2 Experimental setups

In our experiment, all the audio samples are downsampled to 16kHz. The short-time Fourier transform (STFT) with a 1280 window size and 25% overlapping is performed to extract linear and 80-bin mel-spectrograms. Following Li et al. [2023a], we use SR-based data augmentation during training, where the resize ratio r ranges from 0.85 to 1.15. A HiFi-GAN v1 vocoder [Kong et al., 2020a] is used for the data augmentation. The negative sample size N is 10 and the number of shuffle chunks K is 8. We use the posterior encoder and decoder of a FreeVC [Li et al., 2023a] trained on the VCTK training set as the CVAE backbone. All experiments were performed on 2 Nvidia RTX3090 GPUs. The model is trained with up to 80k training steps, where the batch size is 256 and the initial learning rate is 2×10^{-4}. The RoBERTa model is initialized by weights from a CLAP checkpoint[4].

6.3.3 Models for comparison

Since HybridVC is proposed to improve the flexibility and training efficiency compared to the VITS-like VC models, we select the following models for comparison[5]: 1) FreeVC [Li et al., 2023a], an official checkpoint trained with SR data augmentation and a pre-trained speaker encoder on the VCTK training set, 2) FreeVC$\star$: a retrained 900k steps FreeVC with SR data augmentation and a pre-trained speaker encoder

[3] https://github.com/OlaWod/FreeVC

[4] https://github.com/LAION-AI/CLAP

[5] Since the closest methods mentioned previously, Text-guidedVC [Kuan et al., 2023] and PromptVC [Yao et al., 2023], did not release implementation code and training datasets, we cannot use them for a direct and fair comparison.

on the VCTK training set under the official training configuration[1], 3) YourTTS-VC [Casanova et al., 2022][6], officially released checkpoint on VCTK trained with pre-trained speaker encoder.

Two versions of the proposed models are tested: 1) HybridVC$\diamond$, the proposed model trained with KL loss $\mathcal{L}_{kl}$, and 2) HybridVC, the proposed model trained with the total loss $\mathcal{L}$.

6.3.4 Evaluation metrics

We use nine objective evaluation metrics to verify the proposed method in diverse aspects including voice similarity, content error rates, speech naturalness, audio quality, and style prompt consistency. We calculate *word error rate* (WER) and *character error rate* (CER) between source and converted voice by an ASR model[7]. We verify the voice similarity by *Pearson correlation coefficient of fundamental frequency* (F_0-PCC) between source and converted voice, *structural similarity index measure* (SSIM) between target and converted voice. F_0-PCC measures the similarity of pitch contours between converted voices and source voices. We then make use of *speech quality and naturalness assessment* (NISQA) [Mittag et al., 2021], *Fréchet audio distance* (FAD) [Kilgour et al., 2019] computed by a pretrained VGGish [Chen et al., 2020a] and *Fréchet distance* (FD) [Liu et al., 2023a] computed by a pretrained PANNs [Kong et al., 2020b] to verify the naturalness of converted voice and audio quality. The NISQA is a deep learning framework to predict speech quality and naturalness [Mittag et al., 2021], and FAD/FD measures audio quality. We calculate the cosine similarity (COS) between the text embeddings extracted by the text encoder and audio embeddings of converted speeches extracted by the speaker encoder to verify the voice style consistency. Referring to Kuan et al. [2023], we calculate the classification accuracy between converted voice and source voice given single-factor text prompts. For details on the calculation of these metrics, please refer to section 2.7.

6.3.5 Speech Intelligibility, Naturalness, and Audio Quality

Following the experimental design in Li et al. [2023a], we randomly select 9 unseen speakers as target voices and the test set of VCTK as source speeches. We aim to verify whether HybridVC improves training efficiency and maintains similar performance on audio prompt voice style conversion compared to baseline models, which only support audio prompt voice conversion. We train HybridVC$\diamond$ and FreeVC$\star$ on the VCTK training set under the same configuration. table 6.1 shows that HybridVC can achieve competitive performance on speech intelligibility, naturalness, and audio quality with only 15 hours of training on limited computational resources. Extending the experiment to include training on the PromptSpeech dataset, HybridVC maintains overall performance without noticeable degradation, despite the backbone

[6] https://github.com/Edresson/YourTTS
[7] https://huggingface.co/facebook/hubert-large-ls960-ft

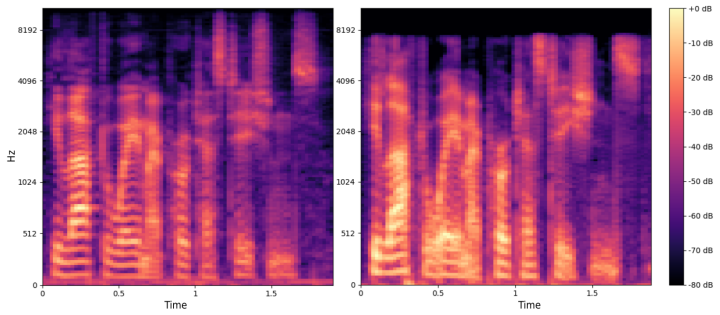


Figure 6.4: Visualization of the source speech (left) and the converted speech at 16k Hz sampling rate with text prompt “he speaks loudly" (right).

Table 6.2: Consistency test for audio prompts and text prompts

	F_0-PCC ↑	FAD ↓	SSIM ↑	COS ↑
Text prompts	0.69	0.79	-	0.68
Audio prompts	0.71	0.76	0.78	-

CVAE only being pre-trained on the VCTK training set. This underscores the robustness of HybridVC and highlights its capability to adapt to new datasets with minimal training resources and time, further explaining the advantage of HybridVC on extensions of practical application.

6.3.6 Consistency Test

We then conduct a consistency test between converted voice and style prompts for both audio and text prompts. We randomly select 9 unseen speakers as audio prompts and the VCTK testing set as source speeches to test consistency between converted voices and style audio prompts. To test the text-style voice conversion consistency, we randomly selected 10 speakers (5 female and 5 male) including 5 utterances for each speaker as source speech and 5 style text prompts from Prompt-Speech. As shown in table 6.2, HybridVC effectively maintains the prosody of source speech and audio quality but accurately converts the voice characteristics given audio and text prompts. Text prompts typically offer less information than audio prompts, resulting in a slight drop in F_0-PCC and FAD values. The COS between converted voice and text prompt, and SSIM between converted voice and audio prompt indicate the style of the converted voice is consistent with the provided prompts.

We then explore the congruence between source and converted voices using single-factor text prompts. fig. 6.4 provides an example of a visual comparison of a converted voice by a single-factor text prompt. In table 6.3, we calculate the accuracy percentage between the absolute average value of the converted voices and the source voices according to the text prompts. For instance, we count the number of converted voices with higher pitch under the “higher pitch” text prompt by comparing the average pitch value between converted voices and source voices. Despite these prompts containing only single style factors, HybridVC successfully adapts voices to match the specified style text prompts. However, we observe that HybridVC demonstrates

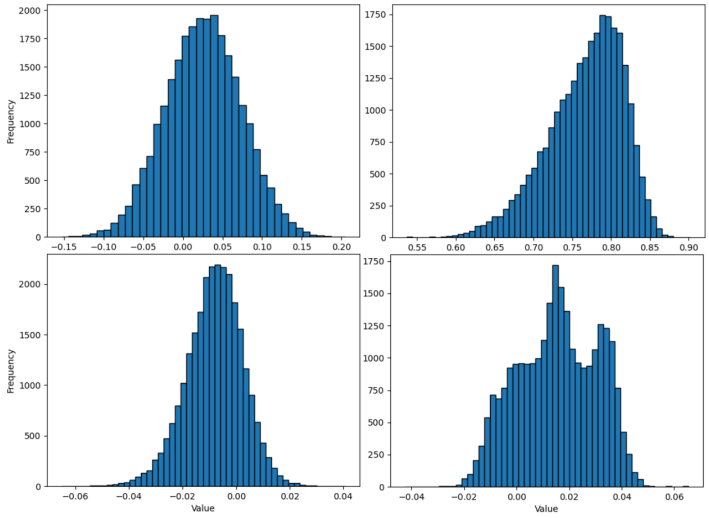


Figure 6.5: Cosine similarity distribution before text embedding optimisation (left-top); after fine-tuning with the proposed negative sampling method (right-top); after fine-tuning with negative samples within the batch (left-bottom); after fine-tuning with negative samples with manually classified labels (right-bottom).

Table 6.3: Accuracy of converted speech by given text prompts

Text prompt	Higher pitch	Higher volume	Lower pitch	Lower volume
Accuracy	89.8 %	91.1%	60.5%	58.9%

less sensitivity to text prompts that include words such as "lower", leaving room for future improvement.

6.3.7 Ablation Study

Latent model. We conduct an ablation study to verify whether the proposed latent model improves training efficiency while maintaining a similar performance as the baseline [Li et al., 2023a]. As shown in table 6.1, HybridVC$\diamond$ has great training efficiency, achieving competitive performance in only 15 hours of training. Adding the negative sampling method to HybridVC increases the training time while maintaining HybridVC's performance. This presents a reasonable trade-off between reducing training efficiency and enabling support for flexible hybrid prompts.

Negative sampling method. To verify the proposed negative sampling method, we train HybridVC augmented with: 1) construct negative samples within a training batch as in Wu et al. [2023a]; 2) classify text embeddings into 54 labels across four categories in PromptSpeech. fig. 6.5 shows the cosine similarity between the text embedding and speaker embedding before and after optimisation with different negative samples. Only HybridVC trained with the proposed negative sampling method effectively refines text embeddings well-aligned to speaker embeddings.

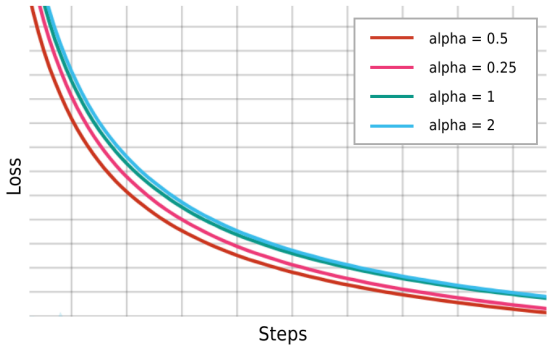


Figure 6.6: Convergence analysis plot against α.

Hyper-parameter α. We conduct convergence analysis against the α values. fig. 6.6 shows that the loss convergence is not sensitive to α values although smaller α values (greater weighting on $\mathcal{L}_{kl}$) lead to slightly faster convergence.

6.3.8 Limitation and Discussion

Although text prompts in HybridVC enable more flexible and diverse editing without the need of reference audio; however, textual descriptions often provide only high-level semantic guidance (e.g., gender). The consistency test indicates that HybridVC may limit supporting fine-gained text guided VC, such as fine speaker attribute control. Consequently, the generated voice depends heavily on the prior knowledge learned by the underlying pre-trained speech generation model, limiting the precision and controllability of text-guided voice conversion. Besides, HybridVC is designed for single-speaker voice conversion under offline settings. Extending the framework to support multi-speaker scenarios, streaming voice conversion, and more diverse speaking styles would further improve its applicability in real-world conversational and interactive speech generation systems.

6.4 Summary

In this chapter, we propose HybridVC, which is a flexible voice conversion model. To address limitations of existing voice conversion frameworks, HybridVC supports both audio and text prompts for voice conversion while enabling efficient training under limited computational resources. HybridVC is a latent model that models the latent distribution of a pre-trained CVAE backbone conditioned on speaker embeddings, and refines the text embedding to better align with information on speaker style in parallel. Our experiments indicate that HybridVC can achieve great training efficiency while maintaining robustness, flexibility, and performance compared to baseline models. Our model could be extended easily to applications, such as personalised voice on social media. However, as the first VC model supporting hybrid text/audio prompts, one limitation is that the optimised text embedding appears to be less sensitive to text prompts including words such as “lower”. This limitation motivates the development of fine-grained text-guided voice conversion models that

support independent control over diverse vocal attributes, such as speaker's age, region, speaking style, and emotion.

Statement of Authorship

This chapter is based on the following publication:

Niu, X., Zhang, J., & Martin, C. P. (2024). HybridVC: Efficient Voice Style Conversion with Text and Audio Prompts. In Proc. Interspeech 2024 (pp. 4368-4372).

Contributions of Candidate

I conceived the research idea, designed the proposed HybridVC framework, implemented the model, conducted all experiments, analyzed the results, and prepared the manuscript.

Contributions of Co-authors

Dr. Charles Patrick Martin and Dr. Jing Zhang supervised the research, provided technical discussions and feedback on the methodology, and contributed to the manuscript revision.

Chapter 7

SoundMorpher: Smooth and Consistent Sound Morphing with Diffusion Model

In previous chapters, we introduced a range of methods across the sound and speech domains. Audio also plays an important role in the music and creativity domain [McAdams, 2013, 2019; Schafer, 1993]. In practical applications, there is a growing demand for methods that extend beyond faithful audio reproduction, requiring flexible control, high perceptual fidelity, and the capacity to perform expressive and stylistic transformations [Burnard, 2012; Kamath et al., 2024; Caetano and Rodet, 2013]. Motivated by these challenges, this chapter focuses on sound morphing, the task of transforming one sound into another in a smooth and semantically consistent way. Sound morphing could potentially be applied to domains including music production, film sound design, and video-audio game development. By facilitating the creation of novel timbral variations, such as hybridized vocal or instrumental sounds [Caetano and Rodet, 2013; Kazazis et al., 2016; Krishnamurthy and Rattani, 2026; Chen et al., 2025], sound morphing provides a flexible and complementary alternative to conventional sound synthesis approaches.

In this chapter, we present SoundMorpher, a flexible sound morphing method, that is designed to generate smooth and consistent morphing trajectories. Traditional techniques typically assume a linear relationship between the morphing factor and sound perception, achieving smooth transitions by linearly interpolating the semantic features of source and target sounds while gradually adjusting the morphing factor. However, these methods oversimplify the complexities of sound perception, resulting in limitations in morphing quality. In contrast, SoundMorpher refines morphing trajectories using an explicit connection between the morph factor and the morphed sounds. This approach further refines the morphing sequence by ensuring a constant target perceptual difference for each transition and determining the corresponding morphing factors using binary search. To address the lack of a formal quantitative evaluation framework for sound morphing, we design a set of metrics based on three established objective criteria. These metrics enable a comprehensive assessment of morphed results and facilitate direct comparisons between methods,

fostering advances in sound morphing research. Extensive experiments demonstrate the effectiveness and flexibility of SoundMorpher in real-world scenarios, showcasing its potential in applications such as creative music composition, film post-production, and interactive audio technologies[1].

7.1 Introduction

Sound morphing is a technique to create a seamless transformation between multiple sound recordings. The goal is to produce intermediate sounds that might exist between two sounds and contrasts with simpler mixing or cross-fading between sounds. Sound morphing has a wide range of applications, including music compositions, synthesizers, psychoacoustic experiments to study timbre spaces [Caetano and Rodet, 2011; Hyrkas, 2021], film post-production, AR/VR, and adaptive audio content in video games [Qamar et al., 2020; Siddiq, 2015]. Traditionally, sound morphing relied on interpolating the parameters of a sinusoidal synthesis model [Tellman et al., 1995; Osaka, 1995; Williams et al., 2014]. High-level audio features in the time-frequency domain can also be controlled to achieve more effective and continuous morphing [Williams et al., 2014; Brookes and Williams, 2010; Caetano and Rodet, 2010, 2011; Roma et al., 2020; Caetano, 2019]. These techniques have quality limitations that rule out applications in inharmonic and noisy sounds such as environmental sounds [Gupta et al., 2023; Kamath et al., 2025]. Increasing interest in sound generation using machine learning (ML) has delivered techniques for ML-based sound morphing [Zou et al., 2021; Gupta et al., 2023; Kim et al., 2019b; Kamath et al., 2025] that outperform traditional methods in scenarios such as timbre morphing, audio texture morphing and environmental sound morphing.

We observed critical limitations of these existing sound morphing methods. Firstly, they are primarily designed for specific morphing tasks or tailored to particular application scenarios on task-specific datasets, which limits their ability to broader applications and different scenarios. Secondly, a lack of sufficient and formal quantitative evaluation limits further analysis of their effectiveness [Gupta et al., 2023] and impedes fair comparisons with other sound morphing techniques. Most importantly, these methods typically assume a linear relationship between morphing factors and morphed sound perception, as proposed by Caetano and Osaka [2012]. They achieve smooth morphing by linearly interpolating semantic features between the source and target sounds and gradually varying the morphing factor. This assumption oversimplifies the complex nature of sound perception, as gradually changing morph factors does not inherently result in smooth perceptual transitions.

We propose SoundMorpher, a flexible diffusion-based framework for smooth sound morphing that supports diverse tasks, including static, dynamic, and cyclostationary morphing, without task-specific retraining on the dataset, thereby enabling open-world applicability. In this chapter, our contributions include

[1]Demonstration page: https://xinleiniu.github.io/SoundMorpher-demo/.

- A heuristic open-world sound morphing pipeline that enables broad applicability and avoids costly retraining and specific datasets.
- The Sound Perceptual Distance Proportion (SPDP) binary search further refines morph factors with perceptual change, ensuring smoother and more natural transitions.
- Extensive experiments demonstrate the effectiveness of SoundMorpher in open-world applications, including timbre, music, and environmental sound morphing.

7.2 Preliminary

This section provides the background knowledge of the sound morphing task and latent diffusion model for audio generation.

7.2.1 Sound Morphing

Sound morphing aims to produce intermediate sounds as different combinations of the model of the source sound $\hat{S}_1$ and the target sound $\hat{S}_2$ [Caetano and Rodet, 2010, 2011], which can be formulated as

$$M(\alpha, k) = (1 - \alpha(k))\hat{S}_1 + \alpha(k)\hat{S}_2 \tag{7.1}$$

Each step k is characterized by one value of a single parameter α, the so-called morph factor, which ranges between 0 and 1. $\alpha = 0$ and $\alpha = 1$ produce resynthesized source and target sounds, respectively. Due to the temporal nature of sounds, sound morphing has three main types: *dynamic morphing*, where α gradually transfers from 0 to 1 over time [Kazazis et al., 2016], *static morphing*, where a single morph factor α leads to an intermediate sound between source and target [Sethares and Bucklew, 2015], and *cyclostationary morphing* where several hybrid sounds are produced in different intermediate points [Slaney et al., 1996].

To solve the limitation on previous works that target on expensive perceptual evaluation only, [Caetano and Osaka, 2012] provided three objective criteria to formalize the evaluation of sound morphing methods: (1) *Correspondence.* The morph is achieved by a description whose elements are intermediate between source and target sounds, highlighting semantic level transition; (2) *Intermediateness.* The morphed objects should be perceived as intermediate between source and target sounds, evaluating perceptual level correlation; (3) *Smoothness.* The morphed sounds should change gradually from source to target sounds, by the same amount of perception increment. Specifically, the linear assumption proposed by [Caetano and Osaka, 2012] is that adding the same amount of morph factor should increase the same amount of perceptual difference. In this study, we evaluate SoundMorpher by the three criteria proposed by [Caetano and Osaka, 2012] with a series of objective quantitative metrics.

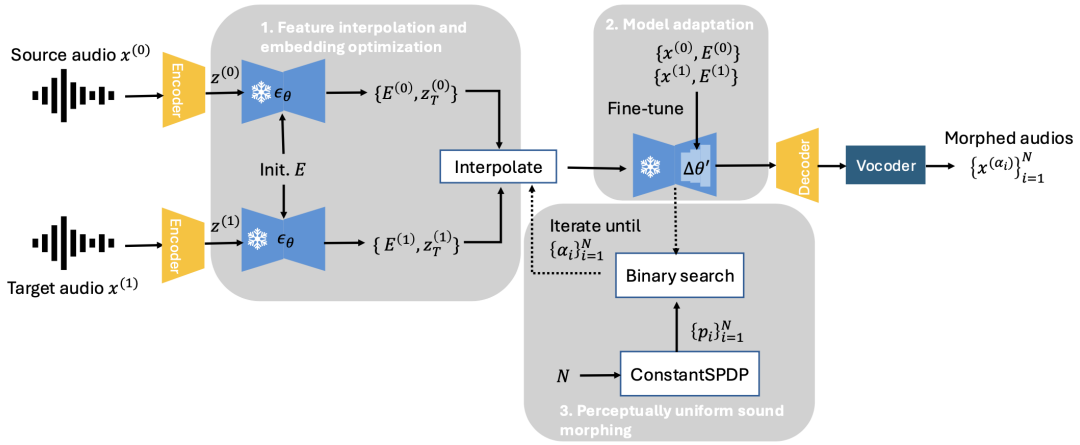


Figure 7.1: Overview of SoundMorpher pipeline, where the snowflake represents the model parameters are frozen.

7.2.2 Latent Diffusion Model on Audio Generation

SoundMorpher uses a pretrained text-to-audio (TTA) latent diffusion model [Rombach et al., 2022b]. This approach offers the advantage of performing various types of sound morphing without the need to train the entire model or use additional datasets. Specifically, we use AudioLDM2 [Liu et al., 2024c], a multi-modal conditioned audio generation model. It uses a pre-trained VAE [Kingma and Welling, 2013] to compress audio x into a low-dimensional latent space as VAE representations z. AudioLDM2 generates latent variables z_0 from Gaussian noise z_T given the condition C and further reconstructs audio $\hat{x}$ from z_0 using a VAE decoder and a vocoder [Kong et al., 2020a]. AudioLDM2 uses an intermediate feature Y as an abstraction of audio data x to bridge the gap between conditions C and audio x, named language of audio (LOA). The LOA feature is obtained by an AudioMAE [Huang et al., 2022; Tan et al., 2024] and a series of post-processing formulated as $Y = \mathcal{A}(x)$. The generation function $\mathcal{G}(\cdot)$ is achieved by an LDM. In the inference phase, AudioLDM2 predicts the LOA feature from the condition using a fine-tuned GPT-2 model [Radford et al., 2019]. Then generates audios conditioned on the estimated LOA feature $\hat{Y}$ and an extra text embedding E_{T5} from a FLAN-T5 [Chung et al., 2024] with an LDM as $\hat{x} = \mathcal{G}(\hat{Y}, E_{T5})$. We denote the conditional embeddings in AudioLDM2 as $E = \{\hat{Y}, E_{T5}\}$, the generative process is $\hat{x} = \mathcal{G}(E)$. To reduce computational demands on inference, AudioLDM2 uses DDIM, which provides an alternative solution and enables significantly reduced sampling steps with high generation quality.

7.3 Method

Given a source and target audio pair $\{x^{(0)}, x^{(1)}\}$, sound morphing aims to generate intermediate sounds $x^{(\alpha(t))}$ between the audio pair given morph factors $\alpha \in [0,1]$. To account for the variation of α over time in eq. (7.1), we discretize the function $\alpha(t)$ $t \in [0,T]$ into N elements, resulting in a morphed sequence of sounds $\{x^{(\alpha_i)}\}_{i=1}^{N}$ based on $\{\alpha_i\}_{i=1}^{N}$ where $\{\alpha_i\}_{i=1}^{N}$ is defined as α_i for i in $1, ..., N$.

Following the smoothness criterion proposed by Caetano and Osaka [2012], the desired morphing trajectory should exhibit approximately uniform perceptual progression, which means that listeners perceive successive intermediate sounds to change by similar amounts along the transition from the source to the target sound. To formalize this concept, we introduce a latent perceptual progression variable p_i, which denotes the perceived morphing position of the intermediate sound $x^{(\alpha_i)}$ on the source-to-target trajectory. Here, p_i does not represent the complete human perception of an audio signal, but rather its relative perceptual location along the morphing path. The mapping $\mathcal{P}(\cdot)$ from a generated audio sample to this perceptual progression is generally unknown and difficult to measure directly. Therefore, our objective is to determine a discrete sequence of morph factors $\{\alpha_i\}_{i=1}^{N}$ such that the perceptual progression between consecutive intermediate sounds is approximately uniform, i.e., Δp remains approximately constant throughout the morphing trajectory. We formulate the problem as follows.

$$p_{i+1} - p_i \equiv \mathcal{P}(x^{(\alpha_{i+1})}) - \mathcal{P}(x^{(\alpha_i)}) = \Delta p \tag{7.2}$$

where $i \in [1, ..., N-1]$.

This formulation is a refined sound morphing problem where, rather than controlling morph factor α, we control the constant perceptual difference Δp to find the optimal trajectory with morph factors $\{\alpha_i\}_{i=1}^{N}$ that will achieve *perceptually smooth sound morphing*. The overall pipeline of SoundMorpher is presented in fig. 7.1. In the following sections, we introduce the details of the SoundMorpher pipeline and provide extensions of our method on the different morphing methods discussed in section 7.2.

7.3.1 Feature Interpolation

Similar to other sound morphing methods, SoundMorpher achieves sound morphing by interpolating features between source and target audios. We introduce a method to bridge the source and target audios based on semantic conditional embeddings in a pre-trained AudioLDM2, leading to a controllable sound morphing method by a morph factor α.

Interpolating optimized conditional embeddings. Following [Yang et al., 2024b], we introduce a text-guided conditional embedding optimisation strategy under a pre-trained AudioLDM2, which retrieves the corresponding conditional embeddings E of the given audio data. This optimisation strategy enables the sound morphing method to handle diverse types of sounds without requiring the model to be retrained on

specific datasets. Following the definition of TTA model section 7.2.2, we denote E as the overall conditional embedding inputs for AudioLDM2 and optimize E below

$$E^{(0)} = \min_{E} \mathcal{L}_{\text{DPM}}(z_0^{(0)}, E; \theta) \text{ and } E^{(1)} = \min_{E} \mathcal{L}_{\text{DPM}}(z_0^{(1)}, E; \theta)$$

The optimized conditional embeddings $E^{(0)}$ and $E^{(1)}$ fully encapsulate the abstract details of $x^{(0)}$ and $x^{(1)}$. Due to the semantically structured nature of the conditional embeddings, the conditional distributions $p_\theta(z|E^{(0)})$ and $p_\theta(z|E^{(1)})$ closely mirror the degree of audio variation between the audio pair. To explore the intermediate data distribution between $z^{(0)}$ and $z^{(1)}$, we bridge these two distributions through linear interpolation. Specifically, we define the interpolated conditional distribution as $p_\theta(z|E^{(\alpha)}) := p_\theta(z|(1-\alpha)E^{(0)} + \alpha E^{(1)})$, where $\alpha \in [0, 1]$.

Interpolating latent state. The conditional embedding represents the conceptual abstraction of audio. We also wish to smoothly morph the content of the audio pair. We follow [Song et al., 2021a] and smoothly interpolate between $z_0^{(0)}$ and $z_0^{(1)}$ by spherical linear interpolation (slerp) between their starting noise $z_T^{(0)}$ and $z_T^{(1)}$ and further obtained the interpolated latent state $z_T^{(\alpha)} := \frac{\sin(1-\alpha)\omega}{\sin\omega} z_T^{(0)} + \frac{\sin\alpha\omega}{\sin\omega} z_T^{(1)}$, where $\omega = \arccos(\frac{z_T^{(0)\intercal} z_T^{(1)}}{||z_T^{(0)}||||z_T^{(1)}||})$. The denoised latent variable $z_0^{(\alpha)}$ is obtained by applying a diffusion the denoising process on the interpolated starting noise $z_T^{(\alpha)}$ and conditioning on the interpolated conditional embedding $E^{(\alpha)}$. The final morphed audio result $x^{(\alpha)}$ is obtained from $z_0^{(\alpha)}$ by the VAE decoder and a vocoder.

7.3.2 Model Adaptation

Model adaptation helps to limit the degree of morphed variation by suppressing high-density regions that are not related to the given inputs. To enhance the output quality of source and target audios, we follow [Yang et al., 2024b] and use LoRA [Hu et al., 2021] to inject a small number of trainable parameters for efficient model adaptation. We fine-tune AudioLDM2 with LoRA trainable parameters using $z^{(0)}$ and $z^{(1)}$. Compared to vanilla fine-tuning approaches, LoRA has advantages in training efficiency by injecting small number of trainable parameters. The model adaptation can be defined as

$$\min_{\Delta\theta'} \mathcal{L}_{\text{DPM}}(z_0^{(0)}, E^{(0)}; \theta + \Delta\theta') + \min_{\Delta\theta'} \mathcal{L}_{\text{DPM}}(z_0^{(1)}, E^{(1)}; \theta + \Delta\theta') \text{ s.t. } \text{rank}(\Delta\theta') = r \quad (7.3)$$

where r represents LoRA rank. We also use other LoRA parameters $\Delta\theta_0$ with rank r_0 to perform bias correction to achieve high text alignment on inference. During inference, with $\theta' = \theta + \Delta\theta'$ and $\theta_0 = \theta + \Delta\theta_0$, the noise prediction becomes $\hat{\epsilon}_\theta(z_t, t, E) := w\epsilon_{\theta'}(z_t, t, E) + (1-w)\epsilon_{\theta_0}(z_t, t, \varnothing)$. Although [Yang et al., 2024b] provide a heuristic suggestion for setting the LoRA rank value in the image morphing task; however, we further investigate the relationship between LoRA rank r and method performance in our empirical experiment.

7.3.3 Consistent and Smooth Sound Morphing

Although interpolating semantic features between the source and target audios enables a smooth transition by leveraging the inherent structure of the semantic space, our ultimate goal is to explore the complex relationship between morphed sound and the morph factor to further enhance the smoothness of the morphing sequence. In this section, we propose a method to further refine the morphed results by fixing the perceptual difference for each transition as constant as in eq. (7.2).

Sound perceptual distance proportion (SPDP). The relationship between morph factor α and perceptual stimuli p is intractable. Our goal is to establish an objective quantitative metric that links p_i and $x^{(\alpha_i)}$ as in eq. (7.2). This metric should satisfy two key conditions: (1) the output p should increase monotonically as α increases; (2) it should accurately represent sound perception and audio descriptive difference between $x^{(\alpha)}$ and $\{x^{(0)}, x^{(1)}\}$, ensuring a smooth transition through intermediate states.

To this end, we propose the *sound perceptual distance proportion* between $x^{(\alpha)}$ and $\{x^{(0)}, x^{(1)}\}$. We define $p_i \in \mathbb{R}^2$ as a 2D vector to represent the perceptual proximity of $x^{(\alpha_i)}$ to both $x^{(0)}$ and $x^{(1)}$. A key motivation is that linearly interpolating optimized conditional embeddings does not guarantee linear interpolation in the output space (e.g., mel-spectrograms), often leading to perceptual artifacts or uneven transitions. Since our base model, AudioLDM2, generates audio directly in the mel-spectrogram domain, we refine interpolation at this level to ensure that the final outputs are perceptually smooth and consistent. In addition, the Mel-scaled logarithmic spectrogram jointly captures perceptual information and descriptive audio structure while aligning with the three objective criteria in section 7.1.

Denoting $x_{mel}^{(\alpha_i)}$ as the log mel-spectrogram of audio $x^{(\alpha_i)}$, the SPDP point p_i of $x^{(\alpha_i)}$ between $x^{(0)}$ and $x^{(1)}$ given α_i is defined as

$$p_i = [\frac{||x_{mel}^{(\alpha_i)} - x_{mel}^{(0)}||_2}{\mathcal{D}}, \frac{||x_{mel}^{(\alpha_i)} - x_{mel}^{(1)}||_2}{\mathcal{D}}] \tag{7.4}$$

where $\mathcal{D} = ||x_{mel}^{(\alpha_i)} - x_{mel}^{(0)}||_2 + ||x_{mel}^{(\alpha_i)} - x_{mel}^{(1)}||_2$ represents the overall sound distance. eq. (7.4) calculates the distance proportion of how close the morphed sample $x^{(\alpha_i)}$ closes between source $x^{(0)}$ and target $x^{(1)}$, meanwhile the L2 norm measures the distance between two log Mel spectrograms. Therefore, the summation of the two elements in p is always equals to 1. Furthermore, the first element of p_i is expected to increase monotonically with α while the second element is monotonically decreasing, providing a clear and interpretable relationship between the parameter α and the perceptual proximity of $x^{(\alpha_i)}$ to $x^{(0)}$ and $x^{(1)}$. In summary, the mel-spectrogram representation effectively captures both perceptual and descriptive audio information, offering an elegant solution to satisfy the criteria of smoothness, correspondence, and intermediateness that proposed by Caetano and Osaka [2012].

Binary search with constant SPDP increment. To produce a smooth morphing trajectory as indicated in eq. (7.2), we use binary search to refine the corresponding $\{\alpha_i\}_{i=1}^N$ based on a constant Δp as in Algorithm 1. The target SPDP sequence $\{p_i\}_{i=1}^N$

is obtained by interpolation $p_i = (1 - \frac{i-1}{N-1})p^{(0)} + \frac{i-1}{N-1}p^{(1)}$, where the two endpoints are $p^{(0)} = [0, 1]^T$ and $p^{(1)} = [1, 0]^T$.

Algorithm 1 Binary search with constant Δp

Require: α_{start}: start alpha; α_{end}: end alpha; N: interpolate number; source audio $x^{(0)}$; target audio $x^{(1)}$;
Ensure: $p_{list} = []$, $p_{start} = [0, 1]^T$, $p_{end} = [1, 0]^T$; $\alpha_{list} = []$, $\alpha_{start} = 0$, $\alpha_{end} = 1$;
Step 1: Obtain target SPDP points with constant Δp.
Procedure ConstantSPDP(N, p_{start}, p_{end})
for i **from** 1 **to** $N - 1$ **do**
  $t \leftarrow \frac{i}{N-1}$; $p_i \leftarrow (1 - t) \times p_{start} + t \times p_{end}$
  $p_{list} \leftarrow p_{list} \cup [p_i]$
end for
# Step 2: Perform binary search given target SPDP points.
Procedure BinarySearch($\alpha_{start}, \alpha_{end}, x^{(0)}, x^{(1)}, p_{list}, \epsilon_{tol}$)
$\alpha_{list} \leftarrow [\alpha_{start}]$, $\alpha_{cur} \leftarrow \alpha_{start}$;
for p_i **from** p_1 **to** p_{N-2} **do**
  $p_{target} \leftarrow p_i$
  $\alpha_{t1} \leftarrow \alpha_{cur}$, $\alpha_{t2} \leftarrow \alpha_{end}$; $\alpha_{mid} \leftarrow \frac{\alpha_{t1}+\alpha_{t2}}{2}$
  $p_{mid} \leftarrow SPDP(x^{\alpha_{mid}}, x^{(0)}, x^{(1)})$
  while $|p_{mid} - p_i| > \epsilon_{tol}$ **do**
    if $p_{mid} > p_{target}$ **then**
      $\alpha_{t2} \leftarrow \frac{\alpha_{t1}+\alpha_{t2}}{2}$
    else
      $\alpha_{t1} \leftarrow \frac{\alpha_{t1}+\alpha_{t2}}{2}$
    end if
    $\alpha_{mid} \leftarrow \frac{\alpha_{t1}+\alpha_{t2}}{2}$
    $p_{mid} \leftarrow SPDP(x^{\alpha_{mid}}, x^{(0)}, x^{(1)})$
  end while
  $\alpha_{list} \leftarrow \alpha_{list} \cup [\alpha_{mid}]$
end for
return α_{list}

7.3.4 Controllable Sound Morphing with Discrete Morphing Series

Given a perceptually uniform sequence of morph factors $\alpha_i{}_{i=1}^{N}$, SoundMorpher supports three types of controllable sound morphing: **Static morphing.** To generate an intermediate sound at a target SPDP point p, we use binary search to find the corresponding α and generate the morphed result $x^{(\alpha)}$. **Cyclostationary morphing.** We sample N uniformly spaced SPDP points $p_i i = 1^N$, find the corresponding α_i, and generate the hybrid sounds $x^{(\alpha_i)} i = 1^N$. **Dynamic morphing.** Dynamic morphing requires perceptual smoothness over time. We obtain N uniform SPDP points p_i and their corresponding α_i by binary search. Each latent segment $\tilde{z}^{(\alpha_i)}$ spans $\frac{K}{N}$ of the total latent length K, and the final output is $x_{\text{morph}} = \text{vocoder}(\text{decoder}(\text{concat}(\tilde{z}^{(\alpha_1)}, ..., \tilde{z}^{(\alpha_N)})))$.

Table 7.1: Timbral morphing for musical instruments compared to the baseline on different instruments.

Group	Method	FAD ↓	FD ↓	CDPAM$_T$ ↓	CDPAM$_S$ ↓	$\mathcal{L}_2^{\text{timbre}}$ ↓	CDPAM$_E$ ↓
Piano ↔ Guitar	SMT	24.73	102.57	1.170	0.116 ± 0.074	1.263	0.122
	Ours	5.21	41.11	0.404	0.044 ± 0.020	0.466	0.132
Harp ↔ Kalimaba	SMT	13.46	88.89	1.495	0.150 ± 0.117	1.355	0.182
	Ours	4.67	37.92	0.768	0.076 ± 0.089	0.462	0.159
Taiko ↔ Hihat	SMT	8.51	131.57	2.339	0.234 ± 0.332	1.584	0.732
	Ours	3.32	47.59	1.314	0.131 ± 0.058	0.359	0.102
Piano ↔ Violin	SMT	21.38	90.63	1.902	0.190 ± 0.069	0.558	0.217
	Ours	3.42	20.14	0.782	0.078 ± 0.020	0.415	0.085
Piano ↔ Organ	SMT	21.36	63.26	1.291	0.129 ± 0.074	1.106	0.097
	Ours	3.29	19.73	0.233	0.023 ± 0.010	0.423	0.097

7.4 Experiments and Results

In this section, we provide systematic experiments on verifying our method. We showcase three applications of SoundMorpher in real-world scenarios: *Timbral morphing for musical instruments*, *Environmental sound morphing*, and *Music morphing*.

7.4.1 Evaluation Metric

We systematically verify SoundMorpher according to the criteria mentioned before, with a series of adapted quantitative objective metrics as

Correspondence. We design a metric that computes absolute error for the midpoint MFCCs proportion, namely MFCCs$_\mathcal{E}$, for descriptive correspondence. Let N be an odd integer. We define a series of perceptually uniform morphing results $\{x^{(\alpha_i)}\}_{i=1}^{N}$ with source and target audio $x^{(0)}$ and $x^{(1)}$, where i in the range of 1 to N. The MFCCs$_\mathcal{E}$ is computed by $\text{MFCCs}_\mathcal{E} = \text{abs}(\frac{||m^{(\frac{N+1}{2})} - m^{(0)}||_2}{||m^{(\frac{N+1}{2})} - m^{(0)}||_2 + ||m^{(\frac{N+1}{2})} - m^{(1)}||_2} - 0.5)$ where $m^{(i)}$ represents the extracted MFCC features of the i^{th} morphed results in the series $x^{(\alpha_i)}$, $m^{(0)}$ and $m^{(1)}$ represents MFCC features of $x^{(0)}$ and $x^{(1)}$. This metric aims to evaluate the MFCC coherence of the midpoint result $x^{(\frac{N+1}{2})}$ between two end points $x^{(0)}$ and $x^{(1)}$. The MFCCs capture *low-level* acoustic characteristics [Davis and Mermelstein, 1980; Logan et al., 2000]. Ideally, we wish the midpoint morphed result contains equal contributions from the source and target. A larger MFCCs$_\mathcal{E}$ indicates that the generated intermediate sample deviates further from the ideal midpoint (i.e., 0.5), reflecting lower morphing consistency. We extract MFCC features with 13 coefficients to compute MFCCs$_\mathcal{E}$. We use *Fréchet audio distance* (FAD) [Kilgour et al., 2019] using a pretrained VGGish [Chen et al., 2020a] and *Fréchet inception distance* (FD) [Eiter and Mannila, 1994] using a pretrained PANNs [Kong et al., 2020b] between morphed audios and source audio to verify semantic distribution similarity between morphed samples and source audio. Both metrics capture a *high-level* of

semantic audio descriptive information by extracting audio features from pre-trained audio classification models. Given two sets of audio X_{mrp} and X_{src}, where X_{mrp} contains consecutive morphed samples as $X_{mrp} = \{x^{(\alpha_i)}\}_{i=1}^{N}$, and X_{src} contains source audio samples. We calculate FAD and FD values based on audio features extracted by pre-trained VGGish [Chen et al., 2020a] and PANNs [Kong et al., 2020b], respectively, between X_{src} and X_{mrp}.

Intermediateness. CDPAM [Manocha et al., 2021] is a metric designed to evaluate the perceptual similarity between audio signals, which is commonly used in domains such as speech [Manocha et al., 2021], music [Jacobellis et al., 2024] and environmental sound [Hai et al., 2024]. We use total CDPAM by $\text{CDPAM}_T = \sum_{i=1}^{N-1} \text{CDPAM}(x^{(\alpha_i)}), x^{(\alpha_{i+1})})$ for morph sequence to reflect direct perceptual intermediateness. A smaller CDPAM_T indicates the morph sequence exhibits higher perceptual intermediate similarity between consecutive sounds, suggesting intermediate consistency.

Smoothness. We calculate the mean and standard deviation of CDPAM along with the morphing path to validate smoothness, as $\text{CDPAM}_S = \text{CDPAM}_{mean} \pm \text{CDPAM}_{std}$, where $\text{CDPAM}_{mean} = \frac{1}{N-1} \sum_{i=1}^{N-1} \text{CDPAM}(x^{(\alpha_i)}, x^{(\alpha_{i+1})})$, and $\text{CDPAM}_{std} = \sqrt{\frac{1}{N-1} \sum_{i=1}^{N-1} (\text{CDPAM}(x^{(\alpha_i)}, x^{(\alpha_{i+1})}) - \text{CDPAM}_{mean})^2}$. To study timbre, we define *timbral distance* as $\mathcal{L}_2^{timbre}$ by $\mathcal{L}_2^{\text{timbre}} = \frac{1}{N-1} \sum_{i=1}^{N-1} ||q^{(\alpha_{i+1})} - q^{(\alpha_i)}||_2$, where $q^{(\alpha_i)}$ represents the corresponding timbre point of $x^{(\alpha_i)}$ in timbre space [McAdams et al., 1995].

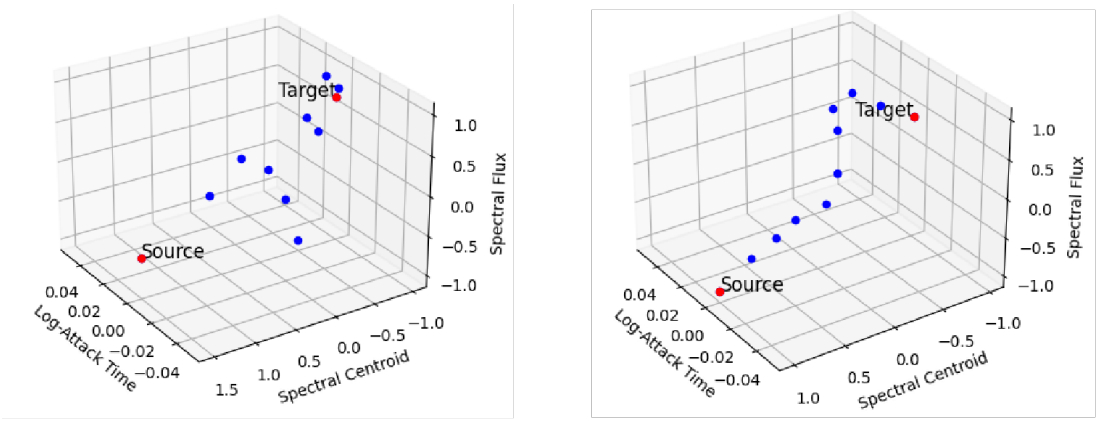
Reconstruction of perceptual correspondence. Lastly, we denote CDPAM_E that calculate CDPAM between $\{x^{(0)}, x^{(1)}\}$ and $\{\hat{x}^{(0)}, \hat{x}^{(1)}\}$, where $\hat{x}$ represents resynthesized end points when $\alpha = 0$ and $\alpha = 1$.

7.4.2 Timbral Morphing for Musical Instruments

Sound morphing can allow timbral morphing between the sound of two known musical instruments, creating sounds from unknown parts of the timbre space [McAdams, 2013; McAdams and Goodchild, 2017]. Timbral morphing for musical instruments involves transitioning between timbres of two different musical instruments to create a new sound. This new sound could possess characteristics of both original timbres as well as new timbral qualities between them, which usually applied to creative arts. In this experiment, we perform timbral morphing for isolated musical instruments given two recordings of the same musical composition played by different music instruments.

Dataset. To ensure high quality of paired compositions on timbral study, we selected 22 paired polyphonic instrument composition samples from demonstration pages of musical timbre transform projects, MusicMagus [Zhang et al., 2024c] and Timbrer [Kemppinen P., 2020]. The paired samples have durations varying from 5s to 10s, with 16.0kHz and 44.1kHz, which involve 5 groups of instrument pairs: 2 samples of piano-violin; 10 samples of piano-guitar; 1 sample of taiko-hihat; 1 sample of piano-organ, and 8 samples of harp-kalimaba.

Baseline. We compare with Sound Morphing Toolbox (SMT) [Caetano, 2019], a set of



SMT **SoundMorpher**

Figure 7.2: Timbre space visualization of morph trajectories for piano-organ timbre morphing. SoundMorpher obtains a smoother morph trajectory in the timbre space.

Matlab functions targeting musical instrument morphing. SMT implements a sound morphing algorithm for isolated musical instrument sounds. Since SMT performs sound morphing based on the linear assumption [Caetano and Osaka, 2012], we uniformly interpolate 11 morph factors in $[0, 1]$.

Results and analysis. The comparison of our method and the baseline on timbral morphing is in table 7.1. Overall, SoundMorpher demonstrates superior morphing quality compared to SMT across various metrics, including audio quality, intermediateness, smoothness, and resynthesis quality, when applied to different types of musical instrument timbre morphing. Notably, SMT fails in Taiko-Hihat timbral morphing due to high reconstruction perceptual error. In contrast, SoundMorpher maintains robustness across different types of musical instruments, making it a more flexible and efficient solution for timbral morphing applications on different types of musical instruments. It is not surprising that SMT has poor performance in this experiment, since the traditional methods are limited to complex or inharmonic sounds [Gupta et al., 2023]. fig. 7.2 visualizes a normalized timbre space, illustrating morphing trajectories generated by SMT and SoundMorpher. The timbre space is defined by three important timbral features: Log-Attack Time, Spectral Centroid, and Spectral Flux [McAdams et al., 1995; McAdams, 2013]. The SMT trajectory shows distinct steps, indicating that the transitions between each intermediate sound are relatively abrupt. The spacing between the blue points suggests that each step represents a significant change in timbre, which may result in a less smooth transition between two musical instruments. In contrast, the trajectory produced by SoundMorpher demonstrates a smoother curve. The points are more closely spaced, indicating more gradual changes between each intermediate timbre. This suggests that SoundMorpher achieves a more continuous and natural-sounding morphing process, with each step being a smaller, more refined adjustment compared to SMT.

Table 7.2: Environmental sound morphing with different types of environmental sounds.

Category	FAD_{cate}	$MFCCs_{\mathcal{E}}\downarrow$	FAD↓	FD↓	$CDPAM_T\downarrow$	$CDPAM_S\downarrow$	$CDPAM_E\downarrow$
Dog ↔ Cat	26.08	0.081	17.77	73.92	1.293	0.323 ± 0.160	0.236
Laughing ↔ Crying baby	10.39	0.044	9.35	65.98	0.855	0.214 ± 0.077	0.289
Church bell ↔ Clock alarm	68.29	0.058	22.89	75.77	2.205	0.551 ± 0.299	0.312
Door knock ↔ Clapping	21.36	0.083	10.85	76.35	1.594	0.428 ± 0.220	0.321

7.4.3 Environmental Sound Morphing

Environmental sounds are used in video game production to provide a sense of presence within a scene. For example, in video games, sound morphing could enhance user immersion by adapting audio cues to specific visual and interactive contexts. This means that it could be useful to morph between sonic locations, e.g., a city and a park, or between sound effects, e.g., different animal sounds to represent fantasy creatures. In this experiment, we perform cyclostationary morphing with $N = 5$ by SoundMorpher across various types of environmental sounds.

Dataset. We use ESC50 [Piczak, 2015] to study the robustness of SoundMorpher across different categories of environmental sounds. Then we make a direct comparison with MorphFader [Kamath et al., 2025], based on environmental sound morphing samples in its demonstration page. ESC50 consists of 5-second recordings organized into 50 semantic classes, which are loosely arranged into 5 major categories. We randomly select four major categories of scenarios to verify our method, including Dog-Cat (animal sounds), Laughing-Crying baby (human sounds), Church bells-Clock alarm (urban noise-interior sound), Door knock-clapping (interior sounds-human sounds). Each category of scenarios contains 25 randomly selected audio pairs, thereby 100 randomly paired samples.

Table 7.3: Model comparison with MorphFader on demo samples

Method	$CDPAM_T\downarrow$	$CDPAM_{mean\pm std}\downarrow$	$MFCCs_{\mathcal{E}}\downarrow$
MorphFader	0.972	0.243 ± 0.139	**0.065**
SoundMorpher	**0.935**	$\mathbf{0.226 \pm 0.162}$	**0.065**

Results and analysis. We evaluate SoundMorpher across different categories of environmental sounds (table 7.2). To capture the semantic gap between categories, we compute the class-level FAD, denoted as FAD_{cate}. Results show that SoundMorpher can effectively morph a wide range of environmental sounds; however, morphing quality decreases when the semantic gap between categories is large. We also observe that both the morphing quality metrics and perceptual reconstruction errors are higher for environmental sounds than for timbre morphing. This stems from the greater complexity of environmental sound, which often involves diverse physical events, strong temporal structure differences, and background noise, making morphing more difficult than in musical data. Spectrogram visualizations in fig. 7.3 illustrate these challenges. Finally, we compare with MorphFader [Kamath et al., 2025]

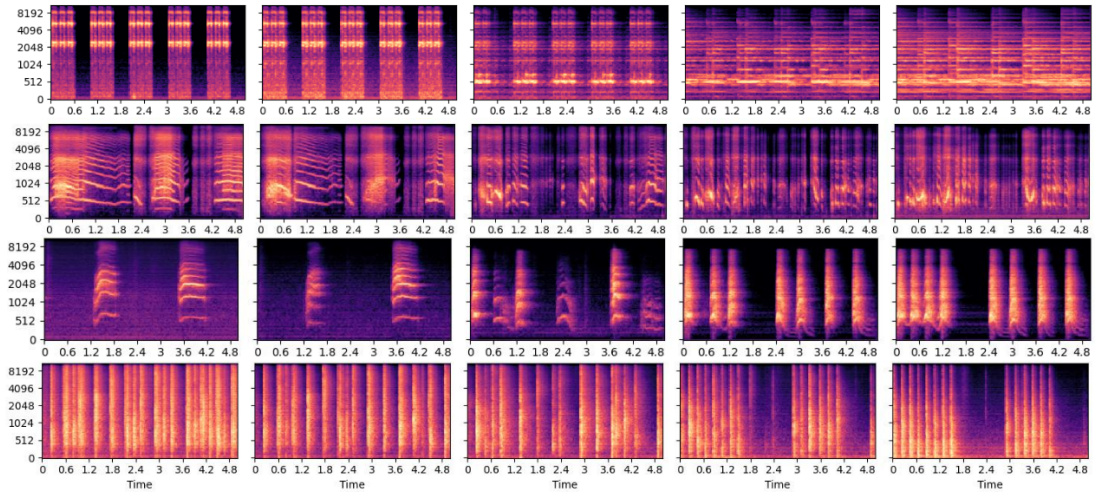


Figure 7.3: Visualization of environmental sound morphing with $N = 5$, from top to bottom: (1) church bells ↔ clock alarm (2) crying baby ↔ laughing (3) cat ↔ dog (4) clapping ↔ wood door knocking

(table 7.3), which performs morphing between text prompts rather than directly between audios. The comparison indicates that SoundMorpher generates smoother intermediates and more natural morphing.

7.4.4 Music Morphing

Film or game post-production often requires blending or fading between music tracks to seamlessly transition background music in between scenes. Music morphing transitions between two music compositions without cross fading, that is, each moment of the morphed music would be a single composition with elements that are perceptually in between both source and target music, rather than simply blending the two together. Different from timbral morphing, music morphing could ideally be accomplished with compositions from different genres and mixed musical instruments. In this experiment, we use SoundMorpher to perform dynamic morphing on music with $N = 15$.

Dataset. Motivated by Zhang et al. [2024c], we randomly selected 50 sample pairs from 20 high-quality musical samples available on AudioLDM2 [Liu et al., 2024c] demonstration page. These 10-second music compositions span different genres and feature both single or mixed musical instrument arrangements.

Baseline. Due to the lack of direct comparisons for dynamic music morphing, we use SoundMorpher without SPDP binary search as the baseline. It performs morphing by smoothly interpolating condition embeddings and latent states, with 15 uniformly sampled morph factors $\alpha \in [0, 1]$.

Results and analysis. Although this experiment contains complex music compositions with different genres and instruments, table 7.4 shows our method is superior on perceptual smoothly transiting source music to the target music and ensures correspondence, intermediateness and smoothness compared to the baseline, which

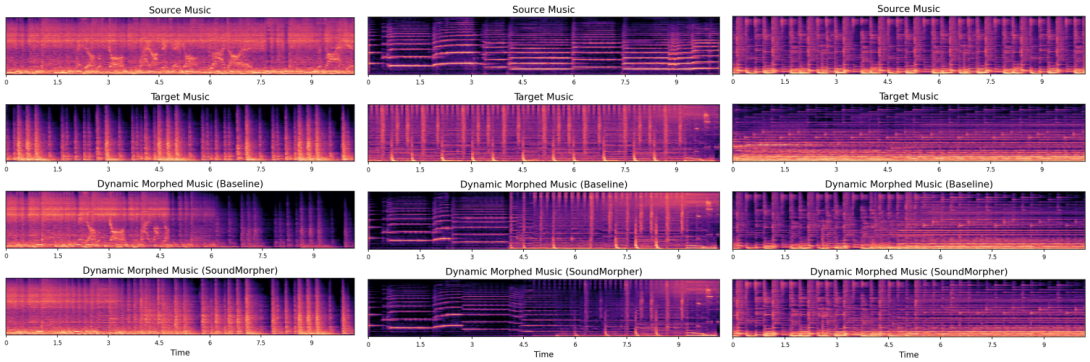


Figure 7.4: Visualization of dynamic morphed music for SoundMorpher and baseline, source and target music.

Table 7.4: Music morphing experimental results, baseline comparison & ablation study for sound perceptual features

Method	MFCCs$_{\mathcal{E}}$ ↓	FAD ↓	FD ↓	CDPAM$_T$ ↓	CDPAM$_S$ ↓	CDPAM$_E$ ↓
Baseline (Mel-Spec.)	0.099	10.82	58.20	0.938	0.067 ± 0.065	0.158
SoundMorpher (Mel-Spec.)	**0.047**	**9.85**	**56.25**	0.847	0.068 ± **0.045**	0.178
Reduced Mel-Spec. (n=2)	0.187	10.31	58.57	0.793	0.056 ± 0.075	0.182
Reduced Mel-Spec. (n=3)	0.151	10.76	59.39	**0.779**	**0.055** ± 0.079	0.151
MFCCs	0.053	10.11	57.38	0.987	0.071 ± 0.050	0.156
Spectral Contrast	0.066	10.54	58.44	0.863	0.061 ± 0.071	0.155

$\star$ where n represents the number of components of PCA for reducing dimension of mel-spectrogram

illustrates the advantage of the proposed SPDP with binary search. fig. 7.4 visually demonstrates the effectiveness of the proposed dynamic morphing method in transitioning smoothly from source to target music while ensuring perceptual consistency and intermediate transformations. Compared to the baseline results, which exhibit noticeable abrupt changes and spectral discontinuities, SoundMorpher produces gradual and smooth spectral transitions over time. This highlights that simply linearly interpolating semantic features, as done in the baseline method, fails to achieve smoother and more consistent sound morphing.

7.4.5 Discussion and Ablation Study

7.4.5.1 Ablation study on SPDP

We verified the perceptual feature in SPDP in music morphing task. We select alternative music information retrieval features including MFCCs with 13 coefficients [Logan et al., 2000], and Spectral contrast [Jiang et al., 2002]. We use principal component analysis (PCA) to reduce the dimensionality of mel-spectrogram to further capture variation of spectral content over time, which is referred to as reduced Mel-Spec. [Stevens et al., 1937; Casey et al., 2008; Jiang et al., 2002]. table 7.4 shows perfor-

Table 7.5: Experiment results on dynamic music morphing.

Init. text	T = 20	T =100	MFCCs$_{\mathcal{E}}$ ↓	FAD ↓	FD ↓	CDPAM$_T$ ↓	CDPAM$_S$ ↓	CDPAM$_E$ ↓
Info.	✓		0.047	10.21	56.62	1.213	0.086 ± 0.069	0.166
Info.		✓	0.044	10.21	56.13	1.077	0.084 ± 0.066	0.155
Uninfo.	✓		0.057	10.37	55.89	1.036	0.074 ± 0.049	0.211
Uninfo.		✓	0.047	9.85	56.25	0.847	0.068 ± 0.045	0.178

mance comparisons of SoundMorpher with different features, SoundMorpher with mel-spectrogram achieves better morphing quality in terms of correspondence and smoothness variation with smaller FAD, FD and CDPAM$_{std}$. While mel-spectrogram yields higher CDPAM$_{mean}$, CDPAM$_T$ and MFCCs$_{\mathcal{E}}$ compared to reduced Mel-Spec. and MFCCs, the differences in metric values are not obvious. However, the overall morph quality with mel-spectrogram is consistently better than other features. This suggests mel-spectrogram, as a pseudo-3D representation, provides more perceptual and semantic information, which contributes to improve morph quality compared to higher-level features.

7.4.5.2 Uninformative v.s. informative initial prompt

Complex audio is often difficult for non-experts to interpret—for example, identifying music genres can be challenging. To evaluate the impact of text-guided conditional embedding optimisation in music morphing, we conduct an ablation study on the initial text prompt. We compare a general, uninformative prompt ("a sound clip of music composition") with informative prompts provided by AudioLDM2. As shown in table 7.5, informative prompts improve resynthesis quality and morph correspondence. However, they slightly reduce intermediateness and smoothness, possibly due to a larger semantic gap between clearer endpoints. Despite this trade-off, the differences are minor, confirming the effectiveness of our conditional embedding optimisation.

7.4.5.3 Inference steps

To verify whether the number of DDIM steps affects SoundMorpher performance, we compare with 20 DDIM steps in table 7.5. A larger number of inference steps seems to help reconstruction quality and slightly improves morph quality, however, performance differences between inference steps are not significant. This indicates that SoundMorpher is robust for inference steps; therefore, we suggest selecting a suitable DDIM step to trade off overall time consumption and morph quality.

7.4.5.4 Ablation study on LoRA rank

We conduct an ablation study on LoRA model adaptation on the music morphing task. We test SoundMorpher with different LoRA ranks. We train LoRA parameters for 150 steps with a learning rate of 10^{-3} and set unconditional bias correction with $r_0 = 2$ for 15 steps with a learning rate of 10^{-3}. In table 7.6, SoundMorpher

Table 7.6: Ablation study on model adaptation with different LoRA rank r on music morphing. Rank with $-$ represents results without LoRA model adaptation.

Rank	MFCCs$_{\mathcal{E}}\downarrow$	FAD$\downarrow$	FD$\downarrow$	CDPAM$_T\downarrow$	CDPAM$_S\downarrow$	CDPAM$_E\downarrow$
-	0.073	10.38	56.02	1.052	0.085 ± 0.054	0.198
4	0.047	9.85	56.25	0.847	0.068 ± 0.045	0.178
8	0.059	10.01	56.35	1.035	0.073 ± 0.051	0.180
16	0.059	9.95	56.14	1.058	0.075 ± 0.052	0.169
32	0.130	10.77	59.06	0.818	0.058 ± 0.082	0.158

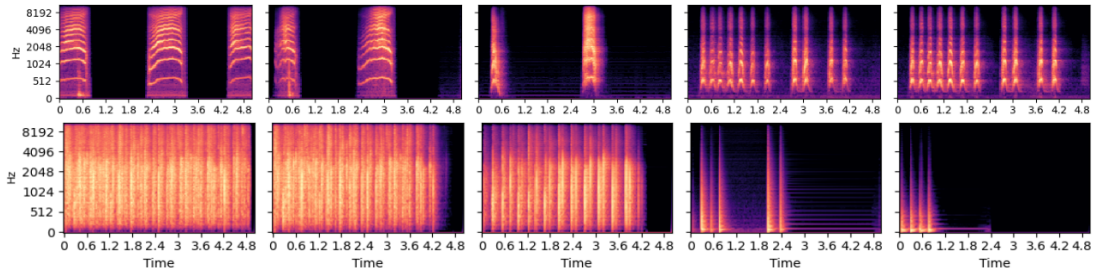


Figure 7.5: Examples of failure cases produced by SoundMorpher.

without model adaptation has an obvious performance drop in morphing correspondence compared to the results with model adaptation. Although a higher LoRA rank slightly improves perceptual reconstruction quality, SoundMorpher with $r = 32$ indicates poor correspondence with large MFCCs$_{\mathcal{E}}$ and large smoothness variance CDPAM$_{std}$. This result indicates that SoundMorpher with a higher rank does not lead to better morphing quality. When $r = 4$, SoundMorpher achieves the best smoothness and correspondence performance compared to $r = 8$, $r = 16$ and $r = 32$. Therefore, we suggest the LoRA rank in SoundMorpher should not be too large (e.g., 32).

7.4.5.5 Limitations

Despite its promising performance, SoundMorpher has several limitations. First, as SoundMorpher is built upon AudioLDM2 operating at a 16 kHz sampling rate, the perceptual quality of the generated audio is bounded by the fidelity and generative capability of the underlying foundation model. Consequently, sounds that are difficult for AudioLDM2 to model, such as white noise (e.g., wind), are often reconstructed with lower fidelity. Second, the proposed conditional embedding optimization assumes that a smooth morphing trajectory exists between the source and target sounds. While this assumption generally holds for sounds with gradual spectral or harmonic variations (e.g., musical instruments), it becomes less valid for sounds with substantially different temporal structures or discrete acoustic events (e.g., ambient noise and dog barks or a clap and two claps). In such cases, the transition may exhibit perceptual discontinuities or abrupt changes, making it inherently more difficult to

achieve smooth morphing. fig. 7.5 presents representative failure cases illustrating these challenging scenarios.

7.5 Summary

In this chapter, we propose SoundMorpher, an open-world sound morphing method based on a pre-trained diffusion model that generates smooth and consistent morph trajectories. Unlike prior approaches, we refine the morphing problem by modeling the complex relationship between morph factor and perception, enabling greater flexibility and higher-quality results across diverse scenarios and methods. We validate SoundMorpher using adapted objective metrics aligned with the three sound morphing criteria from Caetano and Osaka [2012], which may support future standardized evaluations. Experiments demonstrate its effectiveness in various real-world applications, with in-depth analysis provided. While SoundMorpher is also applicable to voice morphing via the speech generation capabilities of generative foundation models, we leave this for future work.

Statement of Authorship

This chapter is based on the following preprint paper:

Niu, X., Zhang, J., & Martin, C. P. (2024). SoundMorpher: Perceptually-Uniform Sound Morphing with Diffusion Model. arXiv preprint arXiv:2410.02144.

Contributions of Candidate

I conceived the research idea, designed the proposed SoundMorpher framework, implemented the model, conducted all experiments, analyzed the results, and prepared the manuscript.

Contributions of Co-authors

Dr. Charles Patrick Martin and Dr. Jing Zhang supervised the research, provided technical discussions and feedback on the methodology, and contributed to the manuscript revision.

Chapter 8

SteerMusic: Enhanced Musical Consistency for Zero-shot Text-guided and Personalized Music Editing

In chapter 7, we introduced a sound morphing framework that leverages a pretrained large audio generative model to perform flexible sound morphing without requiring intensive retraining on task-specific datasets. While sound morphing is widely used in music and sound design, some creative workflows in music production [Zhang et al., 2024c,b; Yang et al., 2025; Cervera et al., 2025] emphasize the controlled modification of existing audio content rather than the synthesis of entirely novel compositions. Motivated by this practical need, this chapter focuses on solving a challenge in the music editing task.

Music editing is an important step in music production, which has broad applications, including game development and film production. Most existing zero-shot text-guided editing methods rely on pretrained diffusion models by involving forward-backward diffusion processes. However, these methods often struggle to preserve the musical content. Additionally, text instructions alone usually fail to accurately describe the desired music. In this chapter, we propose two music editing methods that improve the consistency between the original and edited music by leveraging score distillation. The first method, *SteerMusic*, is a coarse-grained zero-shot editing approach using delta denoising score. The second method, *SteerMusic+*, enables fine-grained personalized music editing by manipulating a concept token that represents a user-defined musical style. *SteerMusic+* allows for the editing of music into user-defined musical styles that cannot be achieved by the text instructions alone. Experimental results show that our methods outperform existing approaches in preserving both music content consistency and editing fidelity[1].

[1]Demonstration page: https://steermusic.pages.dev/.

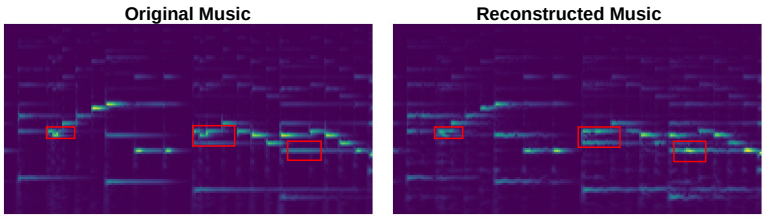


Figure 8.1: The distortion of the reconstructed melody (CQT1-PCC=0.721) after only 20 DDIM inversion steps.

8.1 Introduction

Text-guided diffusion probabilistic models (DPMs) [Ho et al., 2020a; Song et al., 2021a; Lai et al., 2025] have shown impressive performance in generating diverse high-quality audio samples, including music, speech, and sounds [Mariani et al., 2023; Zhang et al., 2024a; Niu et al., 2024d]. These text-to-audio (TTA) diffusion models are trained with large-scale datasets that can generate diverse samples conditioned on the natural language prompts specified. Consequently, the text-guided the music editing task was proposed, which edits music by modifying the corresponding text prompts of the source music. Unlike controllable music generation, the music editing task modifies an existing piece of music, which has two primary objectives: preserving the original musical content and ensuring alignment between the edited music and the desired target.

Existing music editing methods focus on training music editing models from scratch [Copet et al., 2023] or fine-tuning pretrained TTA models [Zhang et al., 2024b], both of which require additional datasets or computational costs. Inspired by recent advances in image editing [Brooks et al., 2023; Huberman-Spiegelglas et al., 2024; Hertz et al., 2022; Zhang et al., 2021], emerging music editing methods have instead pursued zero-shot techniques to reduce computational overhead. Existing zero-shot text-guided music editing pipelines [Zhang et al., 2024c; Manor and Michaeli, 2024a; Liu et al., 2024b] introduce noise into the source music during the forward diffusion process to suppress high-frequency components (e.g., timbral information) and subsequently perform editing during the denoising phase based on target guidance in the diffusion latent space. The process of retrieving a noisy latent representation from the data is commonly referred to as the inversion step. Due to the imperfect diffusion inversion process, the latent representation obtained may not fully preserve the original music content. This issue becomes more serious when the source prompt used as the condition cannot accurately capture the detailed characteristics of the music input [Kawar et al., 2023; Paissan et al., 2023]. We refer to the distortion in the inverted latent representation as an "inversion error". Notably, this distortion occurs even during the reconstruction of only a few DDIM inversion steps [Song et al., 2021a] , which alters the melodic information in the reconstructed results compared to the original music as shown in the CQT spectrogram [Brown, 1991] in Fig. 8.1.

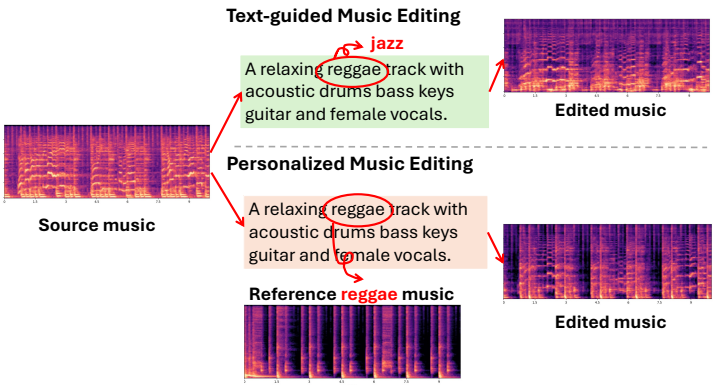


Figure 8.2: SteerMusic: Steering the music style with text-guided music editing or personalized music editing.

In music editing, such distortion can even be compounded, resulting in a failure to preserve instruction-irrelevant content in original music. Although methods such as textual inversion [Gal et al., 2022] have been proposed to mitigate this issue by tuning the embeddings for near-lossless audio reconstruction [Niu et al., 2024c], there is no way to manipulate the target text prompts within the tuned source textual embeddings in editing tasks. A more promising solution to avoid inversion error is the delta denoising score (DDS) [Hertz et al., 2023], a score distillation method that performs editing directly in the data space, which defines a differentiable function rendering the source input. DDS computes the difference in denoising scores between the source and target prompts through a single forward step. This approach eliminates the dependency of a full or partial forward diffusion process that could introduce inversion errors. By operating in the data space, DDS enables high-fidelity editing while preserving instruction-irrelevant content in the input.

Text-guided editing enables flexible and intuitive modifications, requiring users to provide only an arbitrary text instruction to perform the desired edit. However, one limitation of text-guided music editing is that it lacks fine-grained control over the direction and nuance of editing. For instance, editing a guitar performance into the one played by person A with a guitar brand B. Text-based editing alone struggles to specify the exact "guitar" required. Moreover, person A and guitar brand B might be unseen concepts for the models, rendering attempts to specify these words in the text prompt ineffective. To enhance user-personalized controllability in text-to-music generation, DreamSound [Plitsis et al., 2024] introduces a pioneering approach that adapts image personalization techniques [Gal et al., 2022; Ruiz et al., 2023] to the music domain, allowing the extraction of user-defined musical characteristics from reference audio. Besides, DreamSound also demonstrates the potential of leveraging the personalization techniques to perform personalized music editing by manipulating learned musical concepts on the noisy source latent through the denoising process of a personalized diffusion model. Despite their attempts, DreamSound still suffers from the inversion error and struggles to preserve music content while editing

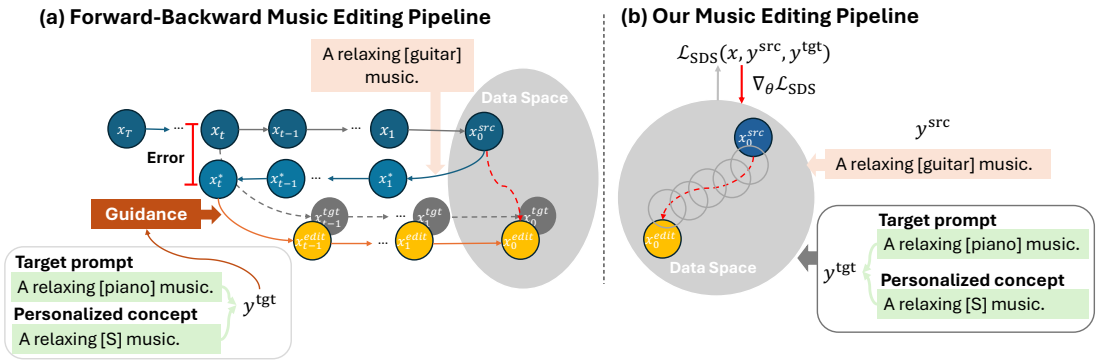


Figure 8.3: Overview of two music editing pipelines: (a) shows the conventional approach, which performs editing during denoising after an inversion process in the diffusion latent space; (b) shows our solution, which directly edits in data space by optimizing the differentiable function $x = g(\theta)$. The differentiable function is initialized with x^{src}. $[S]$ denotes a user-defined concept token, and gray circles represent the optimization trajectory from source to target.

the specific concept given in the text prompt.

In this chapter, we propose two music editing methods, *SteerMusic* and *SteerMusic+*, that can be easily adapted to existing text-to-music DPM based on the score distillation technique. We summarize our key contributions as follows.

1. We propose *SteerMusic*, a zero-shot text-guided music editing pipeline based on a DDS framework, which focuses on coarse-level editing, producing high-fidelity results while preserving source music contents.

2. We propose *SteerMusic+*, a personalized music editing method that leverages user-defined musical concepts to enable customized editing. SteerMusic+, an extension of SteerMusic, enables editing results that are not attainable through text prompts alone. For example, from reggae to the customized reggae given the reference as in Fig. 8.2.

3. We provide extensive experiments to demonstrate that the proposed methods produce superior editing results compared to the existing state-of-the-art methods in terms of musical consistency and edit fidelity.

8.2 Preliminary

Score distillation refines generated samples using the score (i.e., the gradient of the log-density) from a pretrained diffusion model ϵ_ϕ to enforce predefined constraints. It is typically implemented via probability density distillation, where gradients from the source diffusion model are used to recursively refine a differentiable function until the desired outcome is achieved. Score distillation sampling (SDS) [Poole et al.,

2022] pioneered score distillation by optimizing a differentiable function $x = g(\theta)$ to match a target prompt y^{tgt}, where the function $g(\theta)$ renders the source input x with parameters θ. It minimizes the loss $\mathcal{L}_{\text{Diff}} = \mathbb{E}_{t,\epsilon}[w(t)\|\epsilon_\phi(x_t, y^{\text{tgt}}, t) - \epsilon\|_2^2]$, where x_t is a noisy version of x at time t. By omitting the UNet Jacobian, the gradient is $\nabla_\theta \mathcal{L}_{\text{SDS}}(\phi, x = g(\theta), y^{\text{tgt}}) = \mathbb{E}_{t,\epsilon}[w(t)(\epsilon_\phi(x_t, y^{\text{tgt}}, t) - \epsilon)\frac{\partial x}{\partial \theta}]$, where y^{tgt} is the target prompt, x_t is a noisy latent of $x = g(\theta)$ at time step t, and $w(t)$ is a weighting function. However, SDS often produces blurry results in image editing [Wang et al., 2023d; Hertz et al., 2023]. The delta denoising score (DDS) [Hertz et al., 2023] addresses this issue by computing the delta score between the source prompt y^{src} and the target prompt y^{tgt}.

In image editing, DDS refines only regions relevant to the target prompt y^{tgt}, preserving the rest of the image. Given source input x^{src} with prompt y^{src} and target prompt y^{tgt}, the gradient over θ is

$$\nabla_\theta \mathcal{L}_{\text{DDS}}(\phi, x = g(\theta), y^{\text{tgt}}, x^{\text{src}}, y^{\text{src}}) = \mathbb{E}_{t,\epsilon}[w(t)(\epsilon_\phi(x_t, y^{\text{tgt}}, t) - \epsilon_\phi(x_t^{\text{src}}, y^{\text{src}}, t))\frac{\partial x}{\partial \theta}] \quad (8.1)$$

where x_t and x_t^{src} share the same sampled noise ϵ and the timestep t.

Although different variants of SDS and DDS have been proposed for the image domain [Yu et al., 2025; Hertz et al., 2023; Nam et al., 2024; Lin et al., 2025], the application of DDS in music editing remains underexplored. In the next section, we will explain how to incorporate DDS into the music editing tasks.

8.3 Method

In this section, we introduce two music editing methods: *SteerMusic* for zero-shot text-guided editing, and *SteerMusic+* for personalized editing using user-defined concepts from reference music. Unlike the forward-backward pipeline in Fig.8.3 (a), our methods edit directly in a data space (Fig.8.3 (b)), yielding better musical content consistency.

8.3.1 SteerMusic: Zero-Shot Text-Guided Music Editing

We introduce *SteerMusic*, a zero-shot text-guided music editing method that performs editing in the data space. In this setting, our goal is to edit source music signal x^{src} by modifying its corresponding text prompt y^{src}, which y^{src} is a brief description of the source music that includes the specific musical attribute intended for modification. The modified text prompt y^{tgt} acts as the target prompt to guide the editing. To obtain desirable results, the musical content shared by y^{tgt} and y^{src} should be preserved, changing only the content that is distinct in y^{tgt}. For example, if the only change in y^{tgt} compared to y^{src} is replacing only the word "piano" with "guitar"

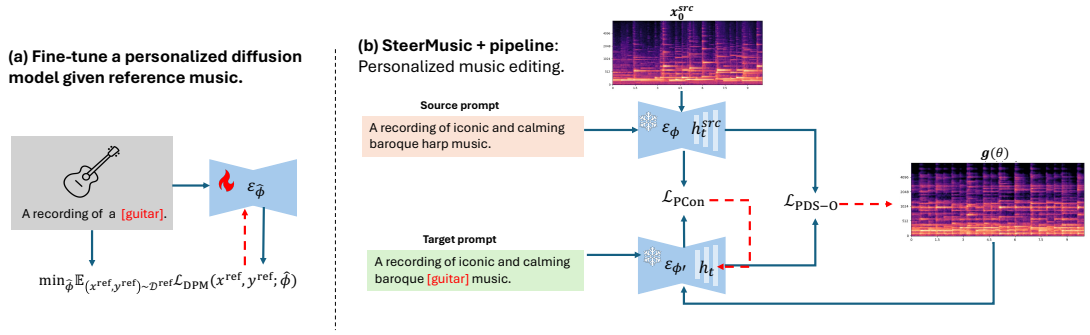


Figure 8.4: Overview of the SteerMusic+ pipeline: (a) Personalized diffusion model (PDM) fine-tuned using $\mathcal{D}^{\text{ref}}$ and a user-defined [guitar] concept token. (b) Personalized editing using the PDM $\epsilon_{\phi'}$ from (a). Red dashed lines indicate gradient flows.

while keeping the rest of the sentence unchanged, the edited music should preserve the melody and tempo, while modifying only the musical instrument.

To achieve this goal, we adopt DDS, which has been previously explored only in the image domain. To the best of our knowledge, we are the first to investigate the potential application in music editing. Following DDS, we define $x = g(\theta)$ rendering source music signal x^{src}. We picked $g(\theta) = \theta$ as the differentiable function in Eq. 8.1, where we initialize $\theta = x_0^{\text{src}}$. i.e. $x = x_0^{\text{src}}$. In SteerMusic, we set ϵ_ϕ as a pretrained TTA or text-to-music DPMs.

Similar to Hertz et al. [2023], the delta score in Eq. 8.1 steers the optimization process toward the target prompt while reducing the noisy editing direction commonly associated with vanilla SDS, leading to enhanced edit fidelity. the SteerMusic method enables flexible editing using text instructions alone; however, it is limited to coarse-grained editing, as text prompts often lack the precision information to capture fine-grained musical details.

8.3.2 SteerMusic+: Personalized Music Editing

As mentioned before, text-guided editing lacks customization for precise music editing, such as transferring music to a specific style. To enable fine-grained personalized editing, we propose *SteerMusic+*, an extension of SteerMusic. In this setting, we have a set of source music and prompt pair $\{x^{\text{src}}, y^{\text{src}}\}$. The target prompt y^{tgt} is constructed by modifying y^{src} to manipulate a user-defined concept token $[S]$, representing the desired customization (see example in Fig. 8.4 (b)). SteerMusic+ uses two pretrained diffusion models:

- A pretrained diffusion probabilistic model (DPM), denoted as ϵ_ϕ, which serves as a reference for maintaining consistency with the original music content; and
- A personalized diffusion model (PDM), denoted as $\epsilon_{\phi'}$, is fine-tuned on a small set of data containing reference music to capture the user-defined concept $[S]$

and guide the editing process toward the desired direction.

We define successful personalized editing by the criteria:

- The instruction-irrelevant part (i.e., $\{y^{\text{tgt}} \cap y^{\text{src}}\}$) in the source music should be maintained.
- The edited musical attributes should perceptually align with the intended personalized musical concept $[S]$.

We now present the SteerMusic+ method that enables personalized music editing with enhanced music consistency.

Personalized Diffusion Model (PDM) serves a foundational part in SteerMusic+. Since training a PDM has been extensively studied in Plitsis et al. [2024]; Ruiz et al. [2023]; Gal et al. [2022]; Kumari et al. [2023], we assume the availability of a pretrained text-to-music PDM in SteerMusic+, as SteerMusic+ is a plug-in pipeline compatible with existing PDMs. The text-to-music PDM $\epsilon_{\phi'}$ captures the user-defined musical concept S by fine-tuning a pretrained DPM ϵ_ϕ under a small set of reference music $\mathcal{D}^{\text{ref}} = \{(x^{\text{ref}}, y^{\text{ref}})_n\}_{n=1}^N$, which N can be as few as 1 [Plitsis et al., 2024]. The fine-tuning is achieved via optimizing the objective

$$\phi' \in \arg\min_{\hat{\phi}} \mathbb{E}_{(x^{\text{ref}}, y^{\text{ref}}) \sim \mathcal{D}^{\text{ref}}} \mathcal{L}_{\text{DPM}}(x^{\text{ref}}, y^{\text{ref}}; \hat{\phi}) \tag{8.2}$$

where $\hat{\phi}$ is initialized with a pretrained DPM weights ϕ. As illustrated in Fig. 8.4 (a), the prompt y^{ref} takes form of "a recording of a $[S]$", where the placeholder $[S]$ corresponds to a defined new concept word embedding. During inference, the PDM can generate music with the newly learned concept (e.g., "A disco song with a $[S]$"). The dataset $\mathcal{D}^{\text{ref}}$ consists of reference audio clips that encapsulate the concept users aim to extract. According to Plitsis et al. [2024], the concept represents a musical style, which can be either an instance of instrument sounds or a specific genre that cannot be yielded even with the most detailed textual description. For instance, if the objective is to capture the user's guitar playing style, the reference audio should feature performance on the user's guitar playing. Conversely, if the goal is to capture the concept of jazz, the reference audio should consist of recordings that exemplify the same jazz style.

While the PDM can generate new music based on concepts from $\mathcal{D}^{\text{ref}}$, it cannot directly edit existing music using y^{tgt} that contains the concepts. Next, we introduce key components of SteerMusic+ that enable personalized editing with PDMs.

Personalized Delta Score (PDS) is an extension of Eq. 8.1 to enable personalized music editing. We define the PDS loss as $\mathcal{L}_{\text{PDS}}(\phi', \phi, x = g(\theta), y^{\text{tgt}}, x^{\text{src}}, y^{\text{src}}) = \mathbb{E}_{t,\epsilon}[w(t)\|\epsilon_{\phi'}(x_t, y^{\text{tgt}}, t) - \epsilon_\phi(x_t^{\text{src}}, y^{\text{src}}, t)\|_2^2]$. By omitting the UNet Jacobian, the gradient over θ is given by $\nabla_\theta \mathcal{L}_{\text{PDS}} = \mathbb{E}_{t,\epsilon}[w(t)(\epsilon_{\phi'}(x_t, y^{\text{tgt}}, t) - \epsilon_\phi(x_t^{\text{src}}, y^{\text{src}}, t))\frac{\partial x}{\partial \theta}]$, where x_t and x_t^{src} share the same sampled noise ϵ at the time step t. This modified delta score can be decomposed into two components: the score of $\epsilon_{\phi'}$ provides the desired direction to guide the editing to match the target prompt with the concept $[S]$. The score of ϵ_ϕ reduces the noisy direction of unintended modification areas. The delta

score between the PDM $\epsilon_{\phi'}$ and the DPM ϵ_{ϕ} may not produce an effective direction toward y^{tgt}, as $\epsilon_{\phi'}$ shifted to the reference distribution $\mathcal{D}^{\text{ref}}$. We introduce an additional component to compensate for the distribution shift induced by the score of $\epsilon_{\phi'}$.

Distribution Shift Regularization. To bridge the distribution gap between $\epsilon_{\phi'}$ and ϵ_{ϕ}, we introduce a regularization term to regularize the edited score to the personalized diffusion model $\epsilon_{\phi'}$. We wish to minimize the distribution shift between two diffusion models by adding a constraint as

$$\begin{aligned} &\min_{\theta} \mathcal{L}_{\text{PDS}}(\phi', \phi, x = g(\theta), y^{\text{tgt}}, x^{\text{src}}, y^{\text{src}}), \\ &\text{subject to } \mathcal{L}_{\text{shift}}(\phi', \phi, x = g(\theta), y^{\text{tgt}}) - \zeta \leq 0. \end{aligned} \tag{8.3}$$

where ζ is a small constant; the regularization term is $\mathcal{L}_{\text{shift}}(\phi', \phi, x = g(\theta), y^{\text{tgt}}) = \mathbb{E}_{t,\epsilon}[w(t)\|\epsilon_{\phi'}(x_t, y^{\text{tgt}}, t) - \epsilon_{\phi}(x_t, y^{\text{tgt}}, t)\|_2^2]$. The gradient with respect to θ is given by $\nabla_\theta \mathcal{L}_{\text{shift}} = \mathbb{E}_{t,\epsilon}[w(t)(\epsilon_{\phi'}(x_t, y^{\text{tgt}}, t) - \epsilon_{\phi}(x_t, y^{\text{tgt}}, t))\frac{\partial x}{\partial \theta}]$. Therefore, the overall gradient through θ is

$$\begin{aligned} \nabla_\theta \mathcal{L}_{\text{PDS-O}}(\phi', \phi, x = g(\theta), y, x^{\text{src}}, y^{\text{src}}) \\ = \underbrace{\nabla_\theta \mathcal{L}_{\text{PDS}}(\phi', \phi, x = g(\theta), y^{\text{tgt}}, x^{\text{src}}, y^{\text{src}})}_{\text{'Delta score points to edit direction'}} \\ + \lambda \underbrace{\nabla_\theta \mathcal{L}_{\text{shift}}(\phi', \phi, x = g(\theta), y^{\text{tgt}})}_{\text{'Delta score regularizes distribution shift'}} \end{aligned} \tag{8.4}$$

where λ is a constant that adjusts regularization strength.

Personalized Contrastive (PCon) Loss. Eq. 8.4 formulates a regularized reference guided editing direction, where the regularized delta score encourages alignment with the target prompt while mitigating the distribution shift. However, it does not explicitly enforce the fidelity to the concept, which may lead to suboptimal editing quality. To further enhance the fidelity of the edit, we incorporate a PCon loss between temporal features, which is modified from a patch-wise contrastive loss [Nam et al., 2024]. PCon loss extracts intermediate features h_l^{src} and h_l that pass through the residual block and the self-attention block from ϵ_{ϕ} conditioned on y^{src} and $\epsilon_{\phi'}$ conditioned on y^{tgt} at the l-th self-attention layer, respectively. The features are then reshaped to size $\mathbb{R}^{T_l \times F_l \times C_l}$, where T_l, F_l, and C_l represent the size of the temporal, spatial, and channel dimensions in the l-th layer, respectively. The patch corresponding to the temporal location on the feature map h_l^{src} is designated as 'positive', and vice versa. The PCon loss is defined as

$$\mathcal{L}_{\text{PCon}}(x, x^{\text{src}}) = \mathbb{E}_h[\sum_l \sum_{t'} \ell(h_l^{t'}, h_l^{\text{src},t'}, h_l^{\text{src},T_l \setminus t'})] \tag{8.5}$$

$$\ell(h, h^+, h^-) = -\log(\frac{\exp(h \cdot h^+/\tau)}{\exp(h \cdot h^+/\tau) + \exp(h \cdot h^-/\tau)})$$

Table 8.1: Model comparison on zero-shot text-guided the music editing task using the ZoME-Bench dataset.

Method	$FAD_{CLAP}\downarrow$	$FAD_{Vggish}\downarrow$	CQT1-PCC↑	LPAPS↓	CLAP↑
DDIM	<u>0.477</u>	4.713	0.330	5.377	**0.264**
SDEdit	0.638	6.749	0.169	6.208	0.218
MusicMagus	0.593	7.631	<u>0.338</u>	<u>5.243</u>	0.238
ZETA	0.509	<u>3.380</u>	0.293	5.458	0.252
SteerMusic	**0.278**	**2.426**	**0.480**	**3.772**	<u>0.259</u>

where $t' \in \{1, ..., T_l\}$ represents the temporal location query patch, the positive patch as $h_l^{src,t'}$ and the other patches as $h_l^{src,T_l \setminus t'}$. $\exp(h \cdot h^+ / \tau)$ is a a positive sample with the same temporal location, $\exp(h \cdot h^- / \tau)$ is a negative sample with a mismatched temporal location in the self-attention features, τ is a temperature parameter as $\tau > 0$.

The gradient of $\mathcal{L}_{\text{PCon}}(x, x^{\text{src}})$ will propagate to the hidden state of self-attention layers h in personalized diffusion $\epsilon_{\phi'}$. Given that the personalized diffusion $\epsilon_{\phi'}$ has a distribution shift over the reference dataset $\mathcal{D}^{\text{ref}}$, the $\mathcal{L}_{\text{PCon}}$ explicitly encourages feature similarity in the frequency domain in self-attention, particularly for attributes that distinguish the target concept. This reinforcement leads the model to prioritize concept consistency over strict temporal alignment with the source music. Consequently, $\mathcal{L}_{\text{PCon}}$ amplifies the distinctive characteristics of the target concept of $\epsilon_{\phi'}$ in SteerMusic+, ensuring that the edited music maintains stronger fidelity to the desired style while allowing structural variations.

8.4 Experiments and Results

In this section, we present comprehensive experiments on SteerMusic and SteerMusic+. We first evaluate SteerMusic in a zero-shot text-guided music editing setting, followed by assessing SteerMusic+ for personalized music editing.

8.4.1 Evaluation Metrics

We evaluate music editing objectively based on two aspects: musical consistency (content preservation before and after editing) and editing fidelity. We follow Manor and Michaeli [2024a] and calculate the following objective metrics for measurement. To evaluate *musical consistency*, we use:

- **Fréchet Audio Distance (FAD)** [Kilgour et al., 2019] which measures the distributional difference between source and edited music (lower the better). We calculate FAD based on both VGGish [Hershey et al., 2017] and clap-laion-music [Wu et al., 2023b] embeddings, denoted as FAD_{Vggish} and FAD_{CLAP} respectively.
- **LPAPS** [Iashin and Rahtu, 2021], an audio version of LPIPS [Zhang et al., 2018b], which quantifies the consistency of the edited audio relative to the source audio (lower the better).

- **Top-1 Constant-Q Transform Pearson Correlation Coefficient (CQT1-PCC)**, which measures melody consistency between the source and edited music (higher the better). The CQT1-PCC [Brown, 1991] extracts the main melody of the music audio and has been shown to outperform traditional chroma-based features in representing melodic characteristics [Hou et al., 2025]. While existing metrics such as FAD and LPAPS provide insight into audio quality and perceptual similarity, they fall short in comprehensively capturing melodic structure. Moreover, existing transcription models [Cwitkowitz et al., 2024; Bittner et al., 2022; Gardner et al., 2022; Chang et al., 2024; Mancusi et al., 2025] effectively extract melodies from real music, they are unreliable for synthesized audio. To address these limitations, we introduce CQT1-PCC as a supplementary objective metric, specifically designed to quantify melodic consistency in generative music editing. This metric enables a more targeted evaluation of whether the core melodic structure of the source audio is retained in the edited result.

To evaluate *editing fidelity*, we use:

- **CLAP Score** [Wu et al., 2023b] which measures the alignment between edited music and the target prompt in text-guided music editing (higher is better).
- **CDPAM** [Manocha et al., 2021], a perceptual audio metric that leverages deep learning representations to measure perceptual distance between audios such as music and speech [Jacobellis et al., 2024; Hai et al., 2024; Gui et al., 2024]. We use CDPAM to evaluate audio perceptual similarity between reference music and the edited result in personalized music editing (lower the better).

8.4.2 Experimental Setup

To evaluate our methods, we use a pretrained AudioLDM2 [Liu et al., 2024c] as the foundational model. In the text-guided music editing experiment, we set the optimisation iteration as 400 with a 30 guidance scale [Ho and Salimans, 2021] in the diffusion model in SteerMusic. In personalized music editing experiment, we follow the setting of the official repository of DreamSound[2] and fine-tune the DreamSound model 100 steps with a $1e^{-5}$ learning rate. We set 400 optimisation steps with a 15 guidance scale and a 0.05 λ value in the PDS-O equation of SteerMusic+. All experiments were performed on a single NVIDIA H100 GPU.

8.4.3 Zero-Shot Text-Guided Music Editing

In this part, we evaluate our the SteerMusic method on the zero-shot text-guided the music editing task.

Dataset. We use the ZoME-Bench dataset [Liu et al., 2024b] which includes 1,000 10-second audio samples from MusicCaps [Agostinelli et al., 2023], each paired with

[2] https://github.com/zelaki/DreamSound

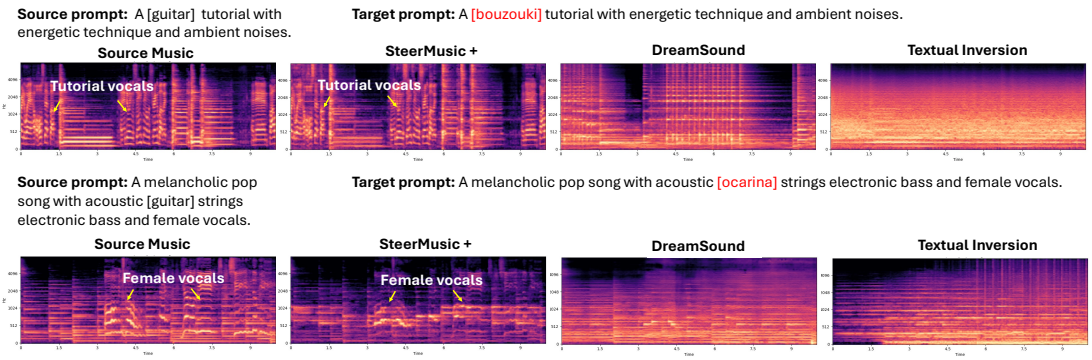


Figure 8.5: A visualization of edited results between SteerMusic+ and baselines in personalized music editing. SteerMusic+ preserves instruction-irrelevant musical content on the source music.

Table 8.2: Model comparison on personalized the music editing task using the ZoME-Bench dataset.

Method	$FAD_{CLAP}\downarrow$	$FAD_{Vggish}\downarrow$	CQT1-PCC↑	LPAPS↓	CDPAM↓
Textual Inv.	0.789	9.688	0.216	5.083	0.713
DreamSound	0.902	10.686	0.292	5.082	0.609
SteerMusic+	**0.362**	**4.434**	**0.399**	**4.125**	**0.593**

source and target text prompts. We evaluate our models on four well-defined editing tasks that require modifying a specific aspect of the audio while preserving the original melody: change instrument (131 clips), change genre (134), change mood (100), and change background (95). To assess long-form editing, we use the MusicDelta subset of MedleyDB [Bittner et al., 2016], with excerpts ranging from 20 seconds to 5 minutes, comprising 34 excerpts of varying styles and lengths, with prompts from Manor and Michaeli [2024a].

Baseline. We compare SteerMusic with zero-shot text-guided music editing methods that plug in the same pretrained AudioLDM2 [Liu et al., 2024c], including SDEdit [Meng et al., 2021], DDIM [Song et al., 2021a], ZETA [Manor and Michaeli, 2024a], and MusicMagus [Zhang et al., 2024c]. To ensure statistical reliability, experiments are conducted using multiple random seeds. We are unable to include MelodyFlow [Le Lan et al., 2024] and MEDIC [Liu et al., 2024b] due to the lack of source code. We exclude AudioEditor [Jia et al., 2025] and AudioMorphX [Liang et al., 2024], which are designed for general sound editing rather than music.

Experimental results. Tab.8.1 compares SteerMusic with zero-shot baselines across various style transfer tasks. SteerMusic achieves higher source consistency, shown by improved CQT1-PCC, lower LPAPS, and FAD. While DDIM attains a slightly higher CLAP score (+5e-3), its low CQT1-PCC and high LPAPS indicate poor preservation of source content due to lack of further source consistency constraints during denoising. In contrast, SteerMusic effectively balances source consistency and edit fidelity,

Table 8.3: Model comparison of SteerMusic and SteerMusic+ on the MusicDelta dataset.

Method	$FAD_{CLAP}\downarrow$	$FAD_{Vggish}\downarrow$	CQT1-PCC↑	LPAPS↓	CDPAM↓	CLAP↑
DDIM	0.646	3.336	0.245	4.972	-	0.316
ZETA	0.665	3.789	0.296	5.071	-	0.319
SDEdit	0.818	8.757	0.137	5.996	-	0.310
SteerMusic	**0.622**	**2.559**	**0.351**	**4.122**	-	**0.321**
DreamSound	0.847	8.972	0.220	5.318	0.583	-
SteerMusic+	**0.666**	**5.506**	**0.273**	**4.574**	**0.581**	-

fulfilling the core objective of music editing.

To assess real-world applicability, we further evaluate our method on the MusicDelta dataset in Tab. 8.3, which consists of varying lengths of music clips. MusicMagus fails in this experiment as it was designed for 5-second editing and doesn't support long-form music, and hence is excluded. SteerMusic consistently outperforms other baselines in terms of edit fidelity and source consistency, demonstrating its robustness and effectiveness in handling longer and more complex music editing.

8.4.4 Personalized Music Editing

In this part, we evaluate our SteerMusic+ method, which is designed for personalized the music editing task.

Dataset. To cover both common and exotic concepts, we selected eight representative musical concepts from the 32 defined in Plitsis et al. [2024]: four instruments style (Guitar, Bouzouki, Ocarina, Sitar) and four genres (Morricone, Reggae, Hiphop, Sarabande). Each concept includes a placeholder instruction and five 10-second reference clips from YouTube and FreeSound. We use the "change instrument" and "change genre" tasks from ZoME-Bench [Liu et al., 2024b], replacing the original target prompt with the selected concept token. Due to the lack of detailed instructions in MusicDelta, we use the standardized prompt as "A recording of a [style] song.", where [style] is either the source style or the target style concept [S].

Baseline. We set two existing personalized music editing methods proposed by Plitsis et al. [2024] as the baselines, Textual inversion and DreamSound. Textual inversion optimizes a concept embedding, whereas DreamSound fine-tunes an AudioLDM2 with rare-token identifiers [Ruiz et al., 2023]. Both methods perform personalized music editing by manipulating the concept token during the denoising process. We follow the official code provided by Plitsis et al. [2024] to obtain text-to-music PDM. We reproduce the personalized music editing methods by calculating noisy latent representations x_t from x_0^{src} of DPM conditioned on the source prompt with a predefined shallow time step t using DDIM inversion [Song et al., 2021a], where $t = 30$. We denoise x_t on the PDM[3] conditioned on the target prompt linked to the learned

[3] Textual inversion optimizes only the concept token embedding rather than fine-tuning a PDM, we denoise x_t using the DPM employed during DDIM inversion. Consequently, textual inversion is incompatible with SteerMusic+ pipeline, in which relies on a PDM.

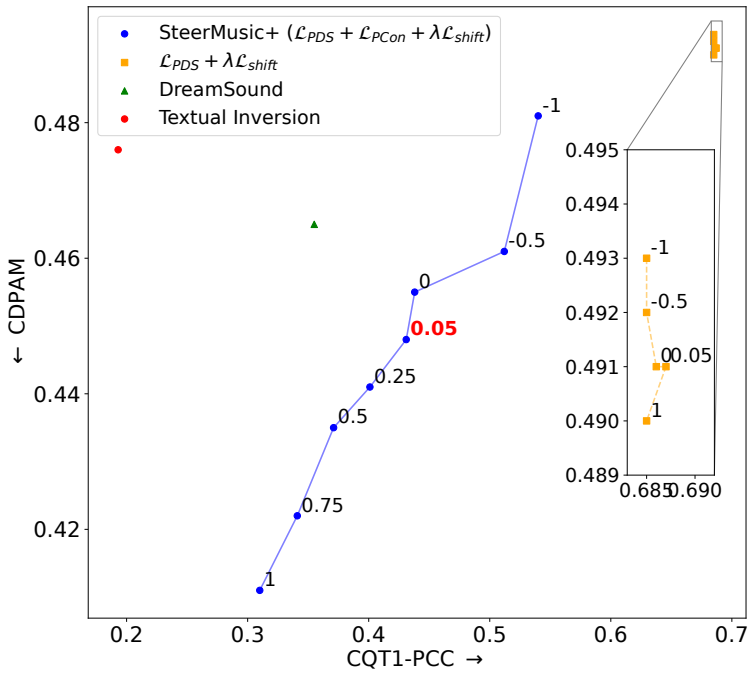


Figure 8.6: Ablation study on SteerMusic+ with adherence to music consistency vs. edit fidelity on the edited music with λ values in Eq. 8.4. The horizontal axis (CQT1-PCC) indicates source melody preservation; the vertical axis (CDPAM) indicates alignment with the target concept.

concept to obtain x_0^{tgt}. To ensure statistical reliability, experiments are conducted using multiple random seeds. We do not compare with Jen-1 DreamStyler [Chen et al., 2024a], which is a personalized music generation method.

Experimental results. We plug SteerMusic+ into the PDM used in DreamSound[3]. Tab. 8.2 shows that SteerMusic+ achieves superior musical consistency compared to the baselines. It indicates that the edited outputs of SteerMusic+ successfully preserve instruction-irrelevant music content in the source music. Furthermore, SteerMusic+ performs accurate edits that align well with the concepts captured from references, as indicated by the low CDPAM compared to the baseline methods.

In long-form music editing (Tab. 8.3), SteerMusic+ still outperforms the baseline. We exclude Textual Inv. from comparison as it fails to produce meaningful results on MusicDelta, likely due to the limitation of its base model with complex inputs. Fig. 8.5 shows a personalized instrument style transfer example, where SteerMusic+ effectively preserves instruction-irrelevant content from the source music.

Ablation study. We conduct an ablation study to understand the effects of different components in SteerMusic+. We used the concept [bouzouki] as it is an uncommon musical instrument that typically requires the use of personalized models. The regularization weight λ, which acts as the Lagrange multiplier for the constraint in Eq. 8.4, was tested within [-1,1] to balance the constraint enforcement and the optimization stability. Yellow dots indicate results obtained by $\mathcal{L}_{\text{PDS-O}}$ in Eq. 8.4 with varying λ

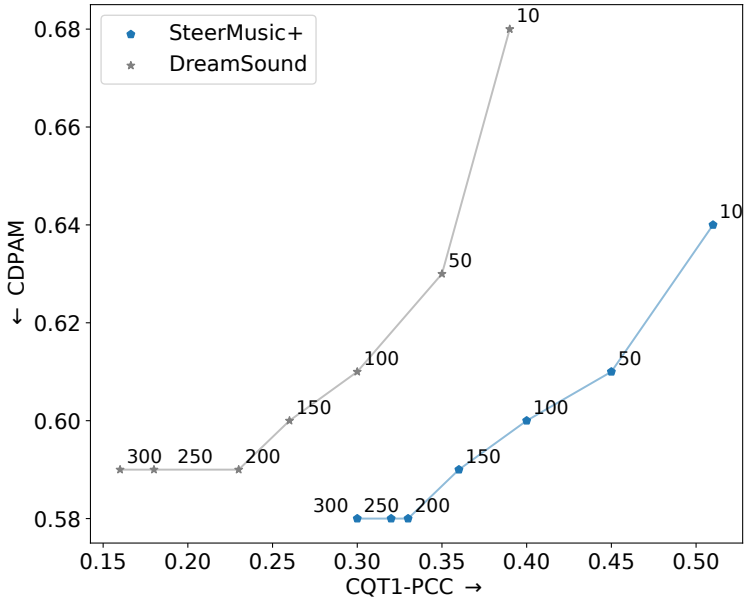


Figure 8.7: Adherence to music consistency vs. edit fidelity to edited results with different fine-tune steps in PDM. The horizontal axis (CQT1-PCC) indicates source melody preservation; the vertical axis (CDPAM) indicates alignment with the target concept.

values. From the zoomed-in view in Fig. 8.6, as the CDPAM value decreases with λ, the loss of $\mathcal{L}_{\text{shift}}$ within $\mathcal{L}_{\text{PDS-O}}$ helps steer the editing toward the concept. However, the effect of $\mathcal{L}_{\text{shift}}$ is minor, suggesting that the gradient of $\mathcal{L}_{\text{PDS}}$ alone does not sufficiently capture the target concept during editing. As a result, the edited music struggles to align with the reference musical characteristics. In contrast, SteerMusic+ (with $\mathcal{L}_{\text{PCon}}$ in Eq. 8.5) leads to a significant decrease in the CDPAM value, which indicates better alignment with the target concept. Since $\mathcal{L}_{\text{PCon}}$ was proposed to explicitly enhance the characteristics of the target concept on the editing, we observe that $\mathcal{L}_{\text{shift}}$ becomes more effective in this setting. Specifically, when λ is negative, the edit preserves the original content, while positive λ pushes the edit toward the target concept.

This highlights a fundamental trade-off between the source music consistency and the adherence to the target style, consistent with the findings of Manor and Michaeli [2024a]. In personalized music editing, where target characteristics are derived from reference music, shifting the output towards the concept often disrupts the original content. This differs from text-guided editing, which follows abstract textual cues rather than concrete musical references. The key challenge in personalized editing is balancing the retention of the original music content, while integrating the distinctive attributes of the reference music. For practical use, we recommend keeping a smaller λ value (e.g., $\lambda = 0.05$) to avoid over-editing.

Limitation. As shown in Fig. 8.7, the number of fine-tuning steps in the PDM signif-

icantly affects the performance of DreamSound and SteerMusic+. A small number of steps (e.g., 50) leads to poor concept learning and low edit fidelity, while a large number of steps (e.g., 200) causes overfitting and source music structural loss. These findings highlight the importance of balancing PDM fine-tuning steps to achieve both high edit fidelity and content preservation in personalized music editing settings. The editing fidelity of SteerMusic+ is fundamentally limited by the ability of the PDMs to capture the concept attributes. Consequently, failure cases can often be attributed to inherent shortcomings in the PDM's representational capacity. We refer readers to Plitsis et al. [2024], which investigated the capacity of PDMs to capture musical attributes and offer insights into improving this capacity.

8.5 Discussions

For coarse-grained text-guided editing (i.e., SteerMusic), textual instructions often provide only high-level semantic descriptions of the desired modification, while many musical attributes, such as timbre, articulation, and playing style, remain unspecified. Consequently, the editing result largely depends on the prior knowledge learned by the underlying foundation model during training. Although this enables flexible zero-shot editing, it also limits user controllability, as users cannot precisely specify which musical attributes should be modified beyond the information explicitly expressed in the text prompt.

In contrast, fine-grained editing using audio examples provides richer guidance but introduces a different challenge. A reference audio clip inherently contains multiple entangled musical attributes, including timbre, melody, rhythm, dynamics, and recording characteristics. SteerMusic+ relies on the underlying personalized diffusion model, DreamSound [Plitsis et al., 2024], to implicitly capture these attributes from reference audio without explicitly disentangling them. Consequently, the effectiveness and editability of SteerMusic+ are fundamentally bounded by the capability of the personalized diffusion model. As discussed in Section 8.4.4, the editing performance for individual musical concepts varies considerably, reflecting the underlying model's ability to learn and represent different concepts from the personalization data. Improving attribute-disentangled representations or developing more controllable personalization techniques would therefore be promising directions for future work.

8.6 Summary

In this chapter, we present two music editing methods, *SteerMusic* and *SteerMusic+*, from coarse-grained to fine-grained music editing. To the best of our knowledge, this is the first work to fully leverage DDS and score distillation in the music editing framework. Our methods address the limitations of prior approaches by enhancing musical consistency while producing high-fidelity edits aligned with target prompts. SteerMusic+ further introduces a personalized editing pipeline that extracts user-

defined style concepts from reference music for fine-grained control. We validate our methods through comprehensive experiments that show clear improvements over existing baselines in both consistency and fidelity. For practical application, the proposed methods could be extended by using a better-trained TTA diffusion backbone with a higher sampling rate to achieve high-fidelity editing results. SteerMusic+ could be combined with SoundMorpher, the method proposed in chapter 7, which enables music editing with newly explored instrument timbres. In future studies, personalized music editing methods could focus on improving user-controllable editing outcomes for more nuanced and expressive edits.

Statement of Authorship

This chapter is based on the following publication:

Niu, X., Cheuk, K. W., Zhang, J., Murata, N., Lai, C. H., Mancusi, M., ... & Mitsufuji, Y. (2026, March). Steermusic: Enhanced musical consistency for zero-shot text-guided and personalized music editing. In Proceedings of the AAAI Conference on Artificial Intelligence (Vol. 40, No. 3, pp. 2000-2010).

This work was done during the internship at SonyAI.

Contributions of Candidate

I conceived the research idea, designed the proposed SteerMusic and SteerMusic+ framework, implemented the model, conducted all experiments, analyzed the results, and prepared the manuscript.

Contributions of Co-authors

Dr. Kin Wai Cheuck, Mr. Naoki Murata, and Dr. Chieh-Hsin Lai provided technical guidance on development of the SteerMusic and SteerMusic+ framework, offered valuable feedback throughout the research, and contributed to revising the manuscript. Dr. Jing Zhang, Dr. Michele Mancusi, Dr. Woosung Choi, Dr. Giorgio Fabbro, Dr. Wei-Hsiang Liao, Dr. Charles Patrick Martin, and Dr. Yuki Mitsufuji supervised the research, provided technical discussions and feedback on the methodology, and contributed to the manuscript revision.

Chapter 9

Conclusion

Audio is a fundamental modality that conveys rich perceptual information such as the environment and physical events that are experienced through human hearing [Phan et al., 2019; Amiriparian et al., 2017; Niu and Martin, 2022]. With recent advances in generative AI, audio-based AIGC has significantly enhanced human creativity and users' audio experience by enabling the generation of diverse and high-quality audio content while substantially lowering the barrier for users without specialized expertise. Audio content can be broadly categorized into three main domains based on its semantic characteristics: sound, speech, and music. However, the boundaries between these domains are not strictly isolated; instead, they often overlap depending on user requirements and application scenarios. This thesis adopts a unified perspective on generative audio synthesis, treating sound, speech, and music as intersecting and complementary domains within a coherent generative audio synthesis framework. In this thesis, six methods are proposed across the domains of sound, speech, music and creative content to tackle the challenges of efficiency, flexibility, and controllability in generative audio synthesis tasks. In short, these contributions advance the capability of generative audio systems to better align with user intent, multi-modal conditions, and real-world application demands.

9.1 Summary of Chapters

In this thesis, we present a series of methods aimed at tackling the three primary challenges in this area: (1) efficiency, (2) flexibility, and (3) controllability.

In chapter 3, we begin with general sound synthesis and introduce SoundLoCD, which is a method with improved training cost and controllability for the task of text-to-sound generation. We demonstrate that adding a contrastive training strategy during discrete latent diffusion training explicitly enhances the connection between text condition and output audios, further enhancing the generated audio quality with complex text descriptions. In addition, to reduce the computational costs during training, we inject a small amount of LoRA parameters on a pretrained diffusion model. We further provide a comprehensive comparison with the baseline model and detailed analyses of the ablation study, to prove the effectiveness of SoundLoCD. The proposed approach enables small-scale training with improved generative quality,

making it particularly suitable for creative applications and domains with limited computational resources.

In chapter 4, we tackle the limitations of the text-to-sound generation pipeline, which fails to support fine-grained temporal control in sound events using text descriptions and cannot produce intelligible and context-aware human voice. To this end, we propose BVS, a video-to-audio generation model that generates synchronized audio from videos with intelligible and environmentally aware speech. To have a better multi-modal fusion, BVS is designed as a two-stage model, in which the first stage focus on multi-modal fusion and the second stage focus on generation. We show how BVS successfully achieves video-to-audio generation with intelligible speech and flexible downstream applications, including video-to-audio with context-aware speech synthesis and immersive audio background conversion. We additionally provide an ablation study to verify the effectiveness of each of the proposed components in BVS. Our proposed method allows for flexible application scenarios with enhanced controllability of speech, which potentially benefits applications such as film production and VR/AR to improve users' immersive experiences.

In chapter 5, we extend our research to speech synthesis and propose a novel method to capture discrete structured optimal paths at the same time allows for gradient optimization during training. Instead of traditional approaches, we convert the optimal path problem to a stochastic manner by a probabilistic softening solution. We further show the method can be applied to probabilistic generative frameworks, i.e., VAEs and propose BDP-VAE, which captures optimal paths in its latent space. We validate our methods on toy experiments and demonstrate our method on two real-world applications, text-to-speech and singing voice synthesis. The BDP-VAE framework successfully solves the limitation of existing methods that relies on external aligners to capture phone-duration information or soften this discrete relationship, and allows the model to be trained on an end-to-end pipeline. This solution improves the training efficiency of existing two-stage models into a one-stage end-to-end training pipeline. Our method can potentially be integrated into other probabilistic frameworks or planning in model-based reinforcement learning to obtain latent paths that leave room for future extensions.

In chapter 6, we present a novel method for the voice conversion task, another branch of speech synthesis, that achieves both training efficiency and application flexibility. The proposed method supports multi-modal conditioning by accepting both text and audio inputs, enabling versatile and precise voice conversion across diverse conditions. We conduct comprehensive experiments, including performance comparisons, parameter sensitivity analyses, and ablation studies, to thoroughly validate the effectiveness of the proposed framework. With the benefit of its flexibility and efficiency, the model can be readily extended to real-world applications, such as personalized voice generation for social media and interactive content.

In chapter 7, we further extend our research into the creative audio domain and propose a novel sound morphing framework, SoundMorpher. SoundMorpher addresses the limitations of existing sound morphing methods, which are often designed for specific tasks and rely on task-specific training datasets. Instead, our ap-

proach uses a pretrained audio generation model to perform general-purpose sound morphing. To further enhance the smoothness and perceptual consistency of morphing trajectories, we introduce a binary search–based refinement algorithm. We demonstrate that SoundMorpher can be effectively applied to a variety of scenarios, including timbre morphing, environmental sound morphing, and music morphing. Its flexibility makes it particularly valuable for creative arts applications, as it eliminates the need for retraining while enabling diverse and adaptive morphing tasks.

In chapter 8, we propose SteerMusic for the music editing task. SteerMusic is designed to enhance the consistency of source music during style editing by performing modifications directly in the data space, in contrast to conventional text-guided music editing approaches that often operate in the latent space. To further improve the customization and controllability of text-guided music editing, we extend this framework to SteerMusic+, which enables fine-grained editing by capturing style concepts from a set of reference music samples provided by users. Experimental results indicate both methods achieve superior performance in terms of source consistency and editing fidelity. In addition, we present detailed analyses and user studies to validate their effectiveness. The proposed frameworks support both coarse-grained and fine-grained editing, making them highly suitable for creative arts and personalized music production applications.

9.2 Summary of Research Contributions

As outlined in section 1.3, this thesis investigates methods in generative audio synthesis in sound, speech, and music domains, which tackle three central research questions:

- **RQ1 Efficiency**: Can generative audio models be designed to efficiently extend to new tasks or datasets while using limited computational resources, without sacrificing synthesis quality?
- **RQ2 Flexibility**: Can a model support flexible audio generation across diverse domains and user demands, enabling seamless manipulation or cross-domain fusion of speech, sounds, and music?
- **RQ3 Controllability**: Can generative models provide fine-grained control over semantic, temporal, and acoustic perception dimensions to accurately fulfill complex user requests?

chapter 3 through chapter 8 present six methods developed to address these three core research challenges. These methods are evaluated across six key tasks in sound, speech, and music synthesis, which are text-to-sound generation, video-to-audio generation, text-to-speech and singing voice synthesis, voice conversion, sound morphing, and text-guided and personalized music editing. An overview of the six methods is provided in table 1.0, with each method corresponding to a published paper and

addressing at least one of the research questions. The summarized contributions of this thesis are summarized as follows:

9.2.1 Summary of Contributions to RQ1: Efficiency

Efficiency is an important factor in generative audio synthesis tasks, typically evaluated through metrics such as inference time, computational resource requirements, and the overall complexity of the training pipeline [Song et al., 2021a; Kim et al., 2020; Li et al., 2025; Liu et al., 2024a; Jeong et al., 2021a]. In this thesis, we tackle the computational resource problem and training complexity problem for two generative audio synthesis tasks, respectively, which are text-to-sound generation and voice synthesis (i.e., including text-to-speech and singing voice synthesis).

In chapter 3, we observed early text-to-sound research introduced conditional diffusion pipelines combining spectrogram VQ-VAE, text encoders, and latent diffusion models. Prior work such as DiffSound [Yang et al., 2023b] and AudioGen [Kreuk et al., 2023] demonstrated the feasibility of conditional audio generation but either lacked explicit constraints for controllable text-conditioned synthesis or required substantial computational resources and large datasets for training. Motivated by these limitations, chapter 3 introduce a LoRA-based conditional discrete contrastive latent diffusion model, SoundLoCD, that injects a small number of trainable parameters into a frozen transformer and employs contrastive learning to pull matched text–audio pairs closer while separating mismatched ones. Experiments on AudioCaps [Kim et al., 2019a] and ESC-50 [Piczak, 2015] show that SoundLoCD achieves better fidelity, diversity, and text–sound correspondence than those generated by DiffSound baseline while using only 2.38M trainable parameters and significantly fewer computational resources. From the perspective of RQ1, SoundLoCD contributes by:

- Dramatically reducing trainable parameter count through LoRA-based adaptation.
- Enhancing audio generation quality despite reduced computational capacity.

In chapter 5, we introduce BDP-VAE, a framework designed to enable end-to-end training for generative tasks that depend on latent structural relationships. We observed that many generative applications, such as speech and music, require identifying structured dependencies between data and conditions, often relying on intermediate processes that complicate training. For example, in text-to-speech and singing voice synthesis tasks, phoneme-duration alignment between phoneme texts and synthesized output is critical, which determines the realism of synthesized voice. To improve the naturalness of results, existing methods usually involve a two-stage training pipeline, in which the model relies on an external aligner to capture the unobserved relationships [Ren et al., 2019, 2020; Liu et al., 2022; Wang et al., 2017b]. Instead, chapter 5 approaches models of stochastic optimal paths as latent variables within a VAE and demonstrates how these paths can be learned directly through the

computational graph of monotonic alignment, thereby forming an end-to-end framework. This design enables end-to-end training for generative tasks in which models rely on unobserved structural information, and is validated on real-world text-to-speech and singing voice synthesis tasks. In relation to RQ1, BDP-VAE advances efficiency by:

- Proposing a unified optimization framework to seek structured optimal paths under DAGs.
- Converting non-end-to-end text-to-speech pipelines into a unified end-to-end training framework without losing the sparsity of learned unobserved structure relationships.

9.2.2 Summary of Contributions to RQ2: Flexibility

Flexibility is another critical requirement in generative audio synthesis, referring to a model's ability to generalize across diverse domains and adapt to varying user demands while maintaining controllability over generated outputs. Unlike task-specific methods that are typically optimized for a specific application, flexible models aim to support seamless manipulation and cross-domain fusion of heterogeneous audio types or user's requirements within one framework. In this thesis, we tackle the flexibility problem on two generative audio synthesis tasks, which are voice conversion and sound morphing.

In chapter 6, we propose HybridVC, a flexible and efficient voice conversion framework, that supports both text and audio prompts to improve controllability and adaptability in speech generation. Flexibility in voice conversion refers to a model's ability to adapt to diverse speakers, styles, and conditioning signals while maintaining synthesis quality. Prior voice conversion systems often struggled with one guidance, such as text or audio [Yao et al., 2023; Li et al., 2023b,a; Kuan et al., 2023], and required substantial computational resources [Kim et al., 2021; Casanova et al., 2022], creating a trade-off between precision and flexibility. To address these limitations, chapter 6 introduce a latent conditional variational autoencoder (CVAE) enhanced with contrastive learning to better align textual descriptions with audio speaker style representations. This hybrid prompting method allows the model to leverage the flexibility of text and audio references, making the model produce either diverse voices from text or precise voices from reference audio. Experiments on VCTK [Yamagishi et al., 2019] and PromptSpeech [Guo et al., 2023] demonstrate competitive intelligibility, naturalness, and audio quality, while the model can be trained in approximately 15 hours on modest hardware, highlighting its efficiency and practical extensibility. From the perspective of RQ2, the work demonstrates clear progress by:

- Supporting multiple conditions: text and audio.
- Reducing dependence on parallel data and enabling adaptation to unseen speakers during training.

- Maintaining comparable performance while reducing the training time and computational resources.

In chapter 7, we propose an open-world sound morphing framework designed to generate perceptually uniform transitions between audio signals using a diffusion model. Traditional morphing techniques typically assume a linear relationship between the morphing factor and perceived sound, which oversimplifies auditory perception and often limits morphing quality [Williams et al., 2014; Brookes and Williams, 2010; Caetano and Rodet, 2010, 2011; Roma et al., 2020; Caetano, 2019; Gupta et al., 2023; Kamath et al., 2025]. Motivated by these limitations, chapter 7 build upon a pretrained text-to-audio diffusion model as a plug-in framework and explicitly models the relationship between morph factors and perceptual stimuli through log mel-spectrogram features. Thereby, SoundMorpher supports diverse morphing tasks without the requirement of retraining the model on a specific dataset and enabling a constant perceptual difference across transitions to obtain smoother intermediate sounds. Extensive experiments demonstrate the effectiveness and versatility of SoundMorpher in real-world scenarios, with potential applications in creative music composition, film post-production, and interactive audio technologies. In relation to RQ2, the chapter advances flexibility by:

- Supporting perceptually consistent transitions across different audio content, for instance, music, timbre, and environmental sounds.
- Expanding sound morphing capabilities beyond task-specific sound morphing without the need to retrain the model on a specific dataset.

9.2.3 Summary of Contributions to RQ3: Controllability

Controllability is a fundamental requirement in generative audio synthesis, focusing on a model's ability to provide precise, fine-grained control over generated audios while preserving intelligibility and semantic alignment. As generative systems become increasingly interactive, users expect the capability to manipulate attributes such as style, prosody, timbre, and content without degrading audio quality or requiring extensive retraining. In this thesis, we explore the controllability problem on two generative audio synthesis tasks, which are video-to-audio generation and text-guided music editing.

In chapter 4, we introduce a controllable framework that extends video-to-audio generation beyond sound effects to include intelligible and context-aware speech. Existing video-to-audio methods largely focus on Foley sounds and struggle to produce intelligible speech [Su et al., 2024; Chen et al., 2024c; Cheng et al., 2025; Luo et al., 2023; Liu et al., 2024d], while environmental speech systems are typically text-driven and fail to align temporally with dynamic video content [Lee et al., 2024b; Jung et al., 2025]. Consequently, controllability in multimodal audio generation has remained constrained, particularly when fine-grained alignment between visual semantics and spoken content is required. Motivated by this gap, chapter 4 propose Beyond Video-to-SFX (BVS), a two-stage architecture consisting of a video-guided audio semantic

model that predicts unified semantic tokens conditioned on phonetic cues, followed by a video-conditioned semantic-to-acoustic model that refines them into detailed audio signals. This design enables synchronized audio generation with environmentally aware speech, supporting scenarios such as context-aware speech synthesis and immersive background conversion, with experiments validating its effectiveness. From the perspective of RQ3, this work demonstrates measurable progress toward fine-grained controllability by:

- Introducing multimodal conditioning as a mechanism for user-steerable video-to-audio synthesis.
- Improving temporal and semantic alignment between generated audio content and condition.
- Expanding video-to-audio generative scope from pure sound effects to ambient sounds with context-aware speech.

In chapter 8, we propose a controllable framework for text-guided music editing that enhances consistency between source music and edited music while enabling both coarse-grained and fine-grained user control. Existing zero-shot text-guided music editing approaches often struggle to preserve the structure of the original piece during modification; meanwhile, text instruction usually provides limited information about the desired music style [Zhang et al., 2024c; Manor and Michaeli, 2024b; Liu et al., 2024b; Plitsis et al., 2024]. chapter 8 addresses this by leveraging score distillation to introduce two complementary methods: SteerMusic, a coarse-grained zero-shot editing technique based on delta denoising score, and SteerMusic+, which enables fine-grained personalized editing by manipulating a concept token representing a user-defined musical style. Together, these methods allow users to steer music editing outputs toward desired stylistic attributes while maintaining musical coherence. Overall, the key contribution to controllability is the introduction of structured editing controls that support personalized, fine-grained manipulation of generated music without requiring task-specific retraining. In relation to RQ3, this chapter contributes by:

- Enabling controllable music editing across two granularity levels.
- Preserving source music structural coherence during music editing.
- Supporting personalized music editing through user-defined musical concepts.
- Eliminating the need for retraining when introducing new stylistic constraints.

9.2.4 Reflection on Current Limitations

This thesis has made significant progress toward improving the efficiency, flexibility, and controllability of generative audio synthesis across speech, music, and environmental sounds. Nevertheless, several important challenges remain:

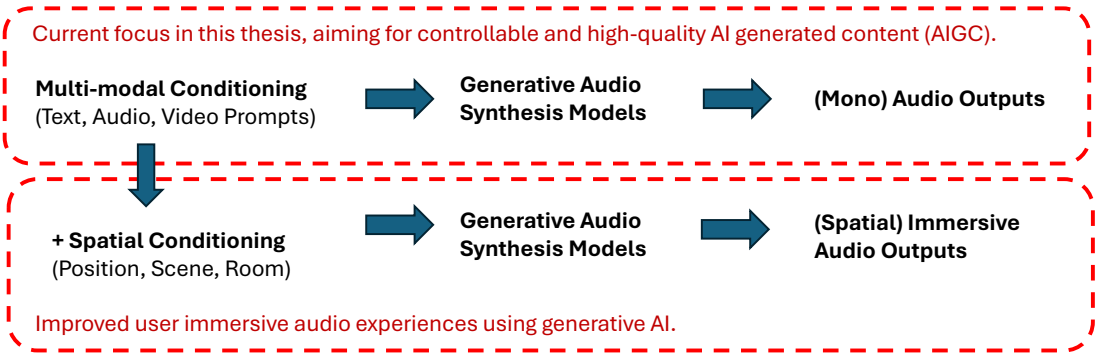


Figure 9.1: Future research: from multi-modal generative audio synthesis to immersive spatial generative audio synthesis.

Efficiency. Although parameter-efficient adaptation strategy substantially reduces training costs in chapter 3, the proposed method, SoundLoCD, still relies on diffusion-based generation. Consequently, inference remains computationally expensive due to iterative denoising, limiting its applicability in real-time or resource-constrained scenarios. Although the proposed BDP-VAE in chapter 5 simplifies the training pipeline and ensures consistency between training and inference, it produces stochastic optimization paths rather than an absolute optimal path, which may lead to performance variability and limit the model's ability to achieve globally optimal solutions in some application cases that the model is sensitive to variability and strongly relies on the global optimal solutions.

Flexibility. While this thesis investigates a broad range of audio generation tasks, including sound synthesis, voice generation, sound morphing, and music editing, each method is designed for a specific application and typically relies on task-specific architectures or pretrained foundation models. Developing a unified audio generation framework capable of supporting diverse generation and editing tasks remains an open challenge.

Controllability. This thesis introduces various control mechanisms, including text prompts, audio clips, and video samples. However, precise attribute-wise user control remains challenging. Text-guided editing often relies on the prior knowledge of foundation models to infer unspecified musical attributes, while audio-guided editing inherits the ambiguity of reference audio, where multiple musical attributes are entangled and cannot be explicitly disentangled by the current framework. Furthermore, the effectiveness of several proposed methods is fundamentally bounded by the representational capability of the underlying pretrained foundation models.

9.3 Future Research

The limitations discussed above also suggest several promising directions for future research. As illustrated in Figure 9.1, this thesis primarily focuses on developing methods for controllable and high-quality AI-generated audio. Building upon these

contributions, future work aims to extend generative audio synthesis toward immersive audio generation, enabling models to produce spatially aware and perceptually rich soundscapes that enhance users' immersive listening experiences.

Spatially-Conditioned Generative Models for Sound Synthesis. Our environmental sound synthesis models (e.g., SoundLoCD in chapter 3 and BVS in chapter 4) generates semantically and temporally aligned but spatially uniform audio. However, real-world auditory perception depends not only on environmental sound content but also on spatial context—direction, distance, and reverberation. Future research could integrate spatial conditioning into the generative process by introducing spatial tokens or positional embeddings that represent sound source location, listener perspective, or room acoustics [Sun et al., 2025; Heydari et al., 2025; Li et al., 2024c]. Such models would enable text-to-spatial-sound generation, a critical step toward immersive AR/VR audio synthesis and intelligent sound design. Furthermore, practical AR/VR applications require low-latency, streaming audio generation rather than offline synthesis. Although the methods proposed in this thesis improve training efficiency, extending them to low-latency or streaming audio generation introduces several additional challenges. First, diffusion-based generation relies on iterative denoising, making it computationally expensive for real-time inference. Secondly, interactive applications such as AR/VR require the generated audio to respond continuously to dynamic user interactions and environmental changes while maintaining temporal consistency and high perceptual quality. Addressing these challenges remains an important direction for future research.

Spatially-aware Voice Generation and Scene-Consistent Speech Rendering. Voice generation and conversion methods discussed in chapter 5 and chapter 4 are conditioned on text and timbral cues, but they assume a single-channel (non-spatial) signal. In spatial or multi-speaker environments, the same voice can appear from different positions or acoustic conditions. Future work could explore spatial voice embeddings, such as latent representations that encode both speaker identity and spatial cues (i.e., azimuth, elevation, and distance). This would allow generative models to produce consistent voices anchored in a virtual space [He and Liu, 2025; Zhang et al., 2025]. This line of work bridges speech synthesis with immersive communication and XR narration, enabling personalized voices rendered naturally in 3D environments.

Spatially Controllable Music Generation and Editing. While this thesis proposes methods in chapter 7 and chapter 8 that allow for flexible manipulation of musical style and content, the spatial dimension (e.g., instruments or sound sources are positioned in a mix) is not supported. A promising research direction is spatially controllable music generation, where a generative model could edit or synthesize both musical attributes and spatial placement. This would enable new forms of interactive, AI-assisted mixing and immersive music composition, aligning closely with future creative workflows in 3D audio production.

Towards Unified Audio Foundation Models. Although this thesis primarily develops generative models, such as diffusion models, for audio synthesis, recent advances in multimodal audio LLMs have demonstrated strong capabilities in unified audio

understanding and reasoning following across speech, sound, and music. Rather than viewing diffusion models and audio LLMs as competing paradigms, we believe they will play complementary roles in future generative audio systems. Diffusion models remain advantageous for synthesizing high-fidelity audio and enhanced controllability, while audio LLMs are used for semantic understanding, cross-modal reasoning, long-context modeling, and instruction following. A promising future direction is therefore to combine these two paradigms within a unified framework, where audio LLMs act as high-level planners that interpret user intentions and generate semantic representations, while diffusion models serve as neural decoders that synthesize high-quality audio conditioned on these representations. Such a hierarchical architecture would naturally extend the unified perspective advocated throughout this thesis. Instead of developing separate systems for speech, environmental sounds, and music, future research could investigate a shared multimodal audio foundation model with task-specific lightweight adaptation modules, combining the strong semantic reasoning capability of audio LLMs with the superior generation fidelity and controllability of diffusion-based models.

Bibliography

Agostinelli, A.; Denk, T. I.; Borsos, Z.; Engel, J.; Verzetti, M.; Caillon, A.; Huang, Q.; Jansen, A.; Roberts, A.; Tagliasacchi, M.; et al., 2023. Musiclm: Generating music from text. *arXiv preprint arXiv:2301.11325*, (2023). (cited on pages 2, 3, 11, 16, and 122)

Amiriparian, S.; Gerczuk, M.; Ottl, S.; Cummins, N.; Freitag, M.; Pugachevskiy, S.; Baird, A.; and Schuller, B., 2017. Snore sound classification using image-based deep spectrum features. In *Interspeech 2017, 20-24 August 2017, Stockholm, Sweden*, 3512–3516. ISCA. (cited on page 129)

Amos, B. and Kolter, J. Z., 2017. Optnet: Differentiable optimization as a layer in neural networks. In *Proceedings of the 34th International Conference on Machine Learning, The International Conference on Machine Learning (ICML) 2017, Sydney, NSW, Australia, 6-11 August 2017*, vol. 70 of *Proceedings of Machine Learning Research*, 136–145. PMLR. http://proceedings.mlr.press/v70/amos17a.html. (cited on page 21)

Baevski, A.; Zhou, Y.; Mohamed, A.; and Auli, M., 2020. wav2vec 2.0: A framework for self-supervised learning of speech representations. *Advances in neural information processing systems*, 33 (2020), 12449–12460. (cited on page 28)

BakIr, G.; Hofmann, T.; Smola, A. J.; Schölkopf, B.; and Taskar, B., 2007. *Predicting structured data*. MIT press. (cited on page 21)

Batlle-Roca, R.; Ibáñez-Martínez, L.; Serra, X.; Gómez, E.; and Rocamora, M., 2025. Musgo: a community-driven framework for assessing openness in music-generative ai. *arXiv preprint arXiv:2507.03599*, (2025). (cited on page 20)

Bińkowski, M.; Sutherland, D. J.; Arbel, M.; and Gretton, A., 2018. Demystifying MMD GANs. In *International Conference on Learning Representations*. (cited on page 30)

Bittner, R.; Wilkins, J.; Yip, H.; and Bello, J. P., 2016. Medleydb 2.0 audio. doi: 10.5281/zenodo.1715175. https://doi.org/10.5281/zenodo.1715175. (cited on page 123)

Bittner, R. M.; Bosch, J. J.; Rubinstein, D.; Meseguer-Brocal, G.; and Ewert, S., 2022. A lightweight instrument-agnostic model for polyphonic note transcription and multipitch estimation. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 781–785. IEEE. (cited on page 122)

Borsos, Z.; Marinier, R.; Vincent, D.; Kharitonov, E.; Pietquin, O.; Sharifi, M.; Roblek, D.; Teboul, O.; Grangier, D.; Tagliasacchi, M.; et al., 2023a. Audiolm: a language modeling approach to audio generation. *IEEE/ACM transactions on audio, speech, and language processing*, 31 (2023), 2523–2533. (cited on pages 3 and 16)

Borsos, Z.; Sharifi, M.; and Tagliasacchi, M., 2022. Speechpainter: Text-conditioned speech inpainting. In *Proc. Interspeech 2022*, 431–435. (cited on page 22)

Borsos, Z.; Sharifi, M.; Vincent, D.; Kharitonov, E.; Zeghidour, N.; and Tagliasacchi, M., 2023b. Soundstorm: Efficient parallel audio generation. *arXiv preprint arXiv:2305.09636*, (2023). (cited on page 16)

Bosman, I. d. V.; Buruk, O. O.; Jørgensen, K.; and Hamari, J., 2024. The effect of audio on the experience in virtual reality: a scoping review. *Behaviour & Information Technology*, 43, 1 (2024), 165–199. (cited on page 35)

Brookes, T. and Williams, D., 2010. Perceptually-motivated audio morphing: Warmth. In *Audio Engineering Society Convention 128*. Audio Engineering Society. (cited on pages 24, 96, and 134)

Brooks, T.; Holynski, A.; and Efros, A. A., 2023. Instructpix2pix: Learning to follow image editing instructions. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 18392–18402. (cited on page 114)

Brown, J. C., 1991. Calculation of a constant q spectral transform. *The Journal of the Acoustical Society of America*, 89, 1 (1991), 425–434. (cited on pages 27, 114, and 122)

Burnard, P., 2012. *Musical creativities in practice*. Oxford University Press. (cited on page 95)

Byun, J. and Loh, C. S., 2015. Audial engagement: Effects of game sound on learner engagement in digital game-based learning environments. *Computers in Human Behavior*, 46 (2015), 129–138. (cited on page 35)

Caetano, M., 2011. *Morphing isolated quasi-harmonic acoustic musical instrument sounds guided by perceptually motivated features*. Ph.D. thesis, Paris 6. (cited on page 24)

Caetano, M., 2019. Morphing musical instrument sounds with the sinusoidal model in the sound morphing toolbox. In *International Symposium on Computer Music Multidisciplinary Research*. Springer. (cited on pages 96, 104, and 134)

Caetano, M. and Osaka, N., 2012. A formal evaluation framework for sound morphing. In *ICMC*. (cited on pages 96, 97, 99, 101, 105, and 111)

Caetano, M. and Rodet, X., 2011. Sound morphing by feature interpolation. In *International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 161–164. (cited on pages 24, 96, 97, and 134)

Caetano, M. and Rodet, X., 2013. Musical instrument sound morphing guided by perceptually motivated features. *IEEE Transactions on Audio, Speech, and Language Processing*, 21, 8 (2013), 1666–1675. (cited on pages 25 and 95)

Caetano, M. F. and Rodet, X., 2010. Automatic timbral morphing of musical instrument sounds by high-level descriptors. In *International Computer Music Conference*, 11–21. (cited on pages 24, 96, 97, and 134)

Cai, X.; Xu, T.; Yi, J.; Huang, J.; and Rajasekaran, S., 2019. Dtwnet: a dynamic time warping network. *Advances in neural information processing systems*, 32 (2019). (cited on page 60)

Casanova, E.; Weber, J.; Shulby, C. D.; Junior, A. C.; Gölge, E.; and Ponti, M. A., 2022. Yourtts: Towards zero-shot multi-speaker tts and zero-shot voice conversion for everyone. In *International conference on machine learning*, 2709–2720. PMLR. (cited on pages 20, 84, 90, and 133)

Casey, M. A.; Veltkamp, R.; Goto, M.; Leman, M.; Rhodes, C.; and Slaney, M., 2008. Content-based music information retrieval: Current directions and future challenges. *Proceedings of the IEEE*, (2008). (cited on page 108)

Cervera, M.; Paissan, F.; Ravanelli, M.; and Subakan, C., 2025. Virtual consistency for audio editing. *arXiv preprint arXiv:2509.17219*, (2025). (cited on page 113)

Chang, H.; Zhang, H.; Jiang, L.; Liu, C.; and Freeman, W. T., 2022. MaskGIT: Masked generative image transformer. In *The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. (cited on page 52)

Chang, H.-J.; Bhati, S.; Glass, J.; and Liu, A. H., 2025. USAD: Universal speech and audio representation via distillation. *ASRU*, (2025). (cited on page 52)

Chang, S.; Benetos, E.; Kirchhoff, H.; and Dixon, S., 2024. Yourmt3+: Multi-instrument music transcription with enhanced transformer architectures and cross-dataset stem augmentation. In *2024 IEEE 34th International Workshop on Machine Learning for Signal Processing (MLSP)*, 1–6. IEEE. (cited on page 122)

Chen, B.; Li, P.; Yao, Y.; and Wang, A., 2024a. Jen-1 dreamstyler: Customized musical concept learning via pivotal parameters tuning. *arXiv preprint arXiv:2406.12292*, (2024). (cited on pages 20 and 125)

Chen, H.; Xie, W.; Vedaldi, A.; and Zisserman, A., 2020a. VggSound: A large-scale audio-visual dataset. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 721–725. IEEE. (cited on pages 29, 40, 90, 103, and 104)

Chen, J.; Xue, W.; Tan, X.; Ye, Z.; Liu, Q.; and Guo, Y., 2024b. Fastsag: towards fast non-autoregressive singing accompaniment generation. *arXiv preprint arXiv:2405.07682*, (2024). (cited on pages 18 and 48)

Chen, K.-Y.; Chen, K.-L.; Yu, Y.-C.; and Ding, J.-J., 2025. Guitar tone morphing by diffusion-based model. In *2025 Asia Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, 329–333. IEEE. (cited on page 95)

Chen, P.; Zhang, Y.; Tan, M.; Xiao, H.; Huang, D.; and Gan, C., 2020b. Generating visually aligned sound from videos. *IEEE Transactions on Image Processing*, 29 (2020), 8292–8302. (cited on page 2)

Chen, Q.; Tan, M.; Qi, Y.; Zhou, J.; Li, Y.; and Wu, Q., 2022a. V2C: visual voice cloning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) 2022, New Orleans, LA, USA, June 18-24, 2022*, 21210–21219. IEEE. doi:10.1109/CVPR52688.2022.02056. https://doi.org/10.1109/CVPR52688.2022.02056. (cited on page 76)

Chen, S.; Wang, C.; Chen, Z.; Wu, Y.; Liu, S.; Chen, Z.; Li, J.; Kanda, N.; Yoshioka, T.; Xiao, X.; et al., 2022b. Wavlm: Large-scale self-supervised pre-training for full stack speech processing. *IEEE Journal of Selected Topics in Signal Processing*, (2022). (cited on pages 28 and 86)

Chen, Z.; Seetharaman, P.; Russell, B.; Nieto, O.; Bourgin, D.; Owens, A.; and Salamon, J., 2024c. Video-guided foley sound generation with multimodal controls. *arXiv preprint arXiv:2411.17698*, (2024). (cited on page 134)

Cheng, H. K.; Ishii, M.; Hayakawa, A.; Shibuya, T.; Schwing, A.; and Mitsufuji, Y., 2025. Mmaudio: Taming multimodal joint training for high-quality video-to-audio synthesis. In *The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. (cited on pages 18, 48, 54, and 134)

Chiu, C. and Raffel, C., 2018. Monotonic chunkwise attention. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net. https://openreview.net/forum?id=Hko85plCW. (cited on page 22)

Choi, J.; Kim, J.-H.; Li, J.; Chung, J. S.; and Liu, S., 2025. V2sflow: Video-to-speech generation with speech decomposition and rectified flow. In *International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*. IEEE. (cited on page 48)

Chowdhury, S.; Nag, S.; Joseph, K.; Srinivasan, B. V.; and Manocha, D., 2024. Melfusion: Synthesizing music from image and language cues using diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 26826–26835. (cited on page 20)

Chung, H. W.; Hou, L.; Longpre, S.; Zoph, B.; Tay, Y.; Fedus, W.; Li, Y.; Wang, X.; Dehghani, M.; Brahma, S.; et al., 2024. Scaling instruction-finetuned language models. *JMLR*, (2024). (cited on page 98)

Chung, Y.-A.; Zhang, Y.; Han, W.; Chiu, C.-C.; Qin, J.; Pang, R.; and Wu, Y., 2021. W2v-bert: Combining contrastive learning and masked language modeling for self-supervised speech pre-training. In *IEEE Automatic Speech Recognition and Understanding Workshop*. IEEE. (cited on pages 51 and 54)

Comanducci, L.; Antonacci, F.; Sarti, A.; et al., 2023. Timbre transfer using image-to-image denoising diffusion implicit models. In *Proceedings of the 24th International Society for Music Information Retrieval Conference, Milan, Italy, November 5-9*. (cited on page 24)

Copet, J.; Kreuk, F.; Gat, I.; Remez, T.; Kant, D.; Synnaeve, G.; Adi, Y.; and Défossez, A., 2023. Simple and controllable music generation. *Advances in Neural Information Processing Systems*, 36 (2023), 47704–47720. (cited on pages 2, 23, and 114)

Copet, J.; Kreuk, F.; Gat, I.; Remez, T.; Kant, D.; Synnaeve, G.; Adi, Y.; and Défossez, A., 2024. Simple and controllable music generation. *Advances in Neural Information Processing Systems*, 36 (2024). (cited on page 20)

Cwitkowitz, F.; Cheuk, K. W.; Choi, W.; Martínez-Ramírez, M. A.; Toyama, K.; Liao, W.-H.; and Mitsufuji, Y., 2024. Timbre-trap: A low-resource framework for instrument-agnostic music transcription. In *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 1291–1295. IEEE. (cited on page 122)

Davis, S. and Mermelstein, P., 1980. Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. *IEEE transactions on acoustics, speech, and signal processing*, 28, 4 (1980), 357–366. (cited on page 103)

Défossez, A.; Copet, J.; Synnaeve, G.; and Adi, Y., 2022. High fidelity neural audio compression. *arXiv preprint arXiv:2210.13438*, (2022). (cited on page 54)

Deng, Y.; Kim, Y.; Chiu, J. T.; Guo, D.; and Rush, A. M., 2018. Latent alignment and variational attention. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, 9735–9747. https://proceedings.neurips.cc/paper/2018/hash/b691334ccf10d4ab144d672f7783c8a3-Abstract.html. (cited on page 21)

Desai, S.; Raghavendra, E. V.; Yegnanarayana, B.; Black, A. W.; and Prahallad, K., 2009. Voice conversion using artificial neural networks. In *International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 3893–3896. IEEE. (cited on pages 23 and 84)

Dhariwal, P.; Jun, H.; Payne, C.; Kim, J. W.; Radford, A.; and Sutskever, I., 2020. Jukebox: A generative model for music. *arXiv preprint arXiv:2005.00341*, (2020). (cited on pages 2, 3, 16, and 20)

Dhariwal, P. and Nichol, A., 2021. Diffusion models beat gans on image synthesis. *The Conference on Neural Information Processing Systems 2021*, 34 (2021), 8780–8794. (cited on page 36)

Djolonga, J. and Krause, A., 2017. Differentiable learning of submodular functions. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, 1013–1023. https://proceedings.neurips.cc/paper/2017/hash/192fc044e74dffea144f9ac5dc9f3395-Abstract.html. (cited on page 21)

Ebner, P. P. and Eltelt, A., 2020. Audio inpainting with generative adversarial network. *arXiv preprint arXiv:2003.07704*, (2020). (cited on page 22)

Eiter, T. and Mannila, H., 1994. *Computing discrete Fréchet distance*. Technical Report, Christian Doppler Laboratory for Expert. (cited on page 103)

Engel, J.; Hantrakul, L.; Gu, C.; and Roberts, A., 2020. Ddsp: Differentiable digital signal processing. *arXiv preprint arXiv:2001.04643*, (2020). (cited on pages 3, 18, 20, and 25)

Engel, J.; Resnick, C.; Roberts, A.; Dieleman, S.; Norouzi, M.; Eck, D.; and Simonyan, K., 2017. Neural audio synthesis of musical notes with wavenet autoencoders. In *The International Conference on Machine Learning (ICML)*. PMLR. (cited on page 25)

Evans, Z.; Parker, J. D.; Rice, M.; Carr, C.; Zukowski, Z.; Taylor, J.; and Pons, J., 2026. Stable audio 3. *arXiv preprint arXiv:2605.17991*, (2026). (cited on page 17)

Gal, R.; Alaluf, Y.; Atzmon, Y.; Patashnik, O.; Bermano, A. H.; Chechik, G.; and Cohen-Or, D., 2022. An image is worth one word: Personalizing text-to-image generation using textual inversion. *arXiv preprint arXiv:2208.01618*, (2022). (cited on pages 20, 115, and 119)

Gardner, J. P.; Simon, I.; Manilow, E.; Hawthorne, C.; and Engel, J., 2022. MT3: Multi-task multitrack music transcription. In *International Conference on Learning Representations*. https://openreview.net/forum?id=iMSjopcOn0p. (cited on page 122)

Garofolo, J.; Lamel, L.; Fisher, W.; Fiscus, J.; Pallett, D.; Dahlgren, N.; and Zue, V., 1992. Timit acoustic-phonetic continuous speech corpus. *Linguistic Data Consortium*, (11 1992). (cited on page 79)

Gemmeke, J. F.; Ellis, D. P. W.; Freedman, D.; Jansen, A.; Lawrence, W.; Moore, R. C.; Plakal, M.; and Ritter, M., 2017. Audio set: An ontology and human-labeled dataset for audio events. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing*. (cited on page 53)

Ghosal, D.; Majumder, N.; Mehrish, A.; and Poria, S., 2023. Text-to-audio generation using instruction-tuned llm and latent diffusion model. *arXiv preprint arXiv:2304.13731*, (2023). (cited on pages 17, 18, 36, and 42)

Glazer, N.; Navon, A.; Segal, Y.; Shamsian, A.; Segev, H.; Buchnick, A.; Pirchi, M.; Hetz, G.; and Keshet, J., 2025. Umbratts: Adapting text-to-speech to environmental contexts with flow matching. *arXiv preprint arXiv:2506.09874*, (2025). (cited on pages 20 and 49)

Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; and Bengio, Y., 2020. Generative adversarial networks. *Communications of the ACM*, 63, 11 (2020), 139–144. (cited on page 12)

Graves, A.; Fernández, S.; Gomez, F.; and Schmidhuber, J., 2006. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In *Proceedings of the 23rd international conference on Machine learning*, 369–376. (cited on page 19)

Griffin, D. and Lim, J., 1984. Signal estimation from modified short-time fourier transform. *IEEE Transactions on acoustics, speech, and signal processing*, 32, 2 (1984), 236–243. (cited on page 28)

Gu, S.; Chen, D.; Bao, J.; Wen, F.; Zhang, B.; Chen, D.; Yuan, L.; and Guo, B., 2022a. Vector quantized diffusion model for text-to-image synthesis. In *The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. (cited on pages 18, 36, and 39)

Gu, S.; Chen, D.; Bao, J.; Wen, F.; Zhang, B.; Chen, D.; Yuan, L.; and Guo, B., 2022b. Vector quantized diffusion model for text-to-image synthesis. In *The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 10696–10706. IEEE/CVF. (cited on page 84)

Gui, A.; Gamper, H.; Braun, S.; and Emmanouilidou, D., 2024. Adapting frechet audio distance for generative music evaluation. In *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 1331–1335. IEEE. (cited on page 122)

Guo, Z.; Leng, Y.; Wu, Y.; Zhao, S.; and Tan, X., 2023. Prompttts: Controllable text-to-speech with text descriptions. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 1–5. IEEE. (cited on pages 20, 87, 89, and 133)

Gupta, C.; Kamath, P.; Wei, Y.; Li, Z.; Nanayakkara, S.; and Wyse, L., 2023. Towards controllable audio texture morphing. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 1–5. IEEE. (cited on pages 25, 96, 105, and 134)

Gutiérrez, E.; Font, F.; Serra, X.; and Wyse, L., 2025. A statistics-driven differentiable approach for sound texture synthesis and analysis. *arXiv preprint arXiv:2506.04073*, (2025). (cited on page 16)

Gölge, E. and Davis, K., 2022. https://archive.is/MotSP#selection-187.0-187.26. (cited on pages 23 and 84)

Hai, J.; Wang, H.; Yang, D.; Thakkar, K.; Dehak, N.; and Elhilali, M., 2024. DPM-TSE: A diffusion probabilistic model for target sound extraction. In *International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*. (cited on pages 104 and 122)

Halperin, T.; Ephrat, A.; and Peleg, S., 2019. Dynamic temporal alignment of speech to lips. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 3980–3984. IEEE. (cited on page 60)

Han, B.; Dai, J.; Hao, W.; He, X.; Guo, D.; Chen, J.; Wang, Y.; Qian, Y.; and Song, X., 2023. Instructme: An instruction guided music edit and remix framework with latent diffusion models. *arXiv preprint arXiv:2308.14360*, (2023). (cited on page 23)

Hasegawa-Johnson, M.; Cole, J.; Hirschberg, J.; Jilka, M.; and Tannenbaum, R., 2005. Penn phonetics toolkit (p2tk): A software suite for sound analysis as a function of time. Technical report, Department of Linguistics, University of Pennsylvania. (cited on page 60)

He, S. and Liu, R., 2025. Multi-source spatial knowledge understanding for immersive visual text-to-speech. In *International Conference on Acoustics, Speech and Signal*. IEEE. (cited on pages 20, 49, and 137)

Heckerman, D., 1998. A tutorial on learning with bayesian networks. In *Learning in Graphical Models* (Ed. M. I. Jordan), vol. 89 of *NATO ASI Series*, 301–354. Springer Netherlands. doi:10.1007/978-94-011-5014-9_11. https://doi.org/10.1007/978-94-011-5014-9_11. (cited on page 21)

Hegde, S. B.; Prajwal, K.; Mukhopadhyay, R.; Namboodiri, V. P.; and Jawahar, C., 2022. Lip-to-speech synthesis for arbitrary speakers in the wild. In *Proceedings of the 30th ACM International Conference on Multimedia*, 6250–6258. (cited on page 2)

Hershey, S.; Chaudhuri, S.; Ellis, D. P.; Gemmeke, J. F.; Jansen, A.; Moore, R. C.; Plakal, M.; Platt, D.; Saurous, R. A.; Seybold, B.; et al., 2017. Cnn architectures for large-scale audio classification. In *2017 ieee international conference on acoustics, speech and signal processing (icassp)*, 131–135. IEEE. (cited on page 121)

Hertz, A.; Aberman, K.; and Cohen-Or, D., 2023. Delta denoising score. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2328–2337. (cited on pages 115, 117, and 118)

Hertz, A.; Mokady, R.; Tenenbaum, J.; Aberman, K.; Pritch, Y.; and Cohen-Or, D., 2022. Prompt-to-prompt image editing with cross attention control. *arXiv preprint arXiv:2208.01626*, (2022). (cited on page 114)

Heydari, M.; Souden, M.; Conejo, B.; and Atkins, J., 2025. Immersediffusion: A generative spatial audio latent diffusion model. In *ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 1–5. IEEE. (cited on page 137)

Ho, J.; Jain, A.; and Abbeel, P., 2020a. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33 (2020), 6840–6851. (cited on pages 3, 13, 14, and 114)

Ho, J.; Jain, A.; and Abbeel, P., 2020b. Denoising diffusion probabilistic models. In *NeurIPS*, 6840–6851. (cited on pages 18 and 36)

Ho, J. and Salimans, T., 2021. Classifier-free diffusion guidance. In *The Conference on Neural Information Processing Systems 2021 Workshop on Deep Generative Models and Downstream Applications*. (cited on pages 15 and 122)

Hou, S.; Liu, S.; Yuan, R.; Xue, W.; Shan, Y.; Zhao, M.; and Zhang, C., 2025. Editing music with melody and text: Using controlnet for diffusion transformer. In *ICASSP 2025 - 2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 1–5. doi:10.1109/ICASSP49660.2025.10890309. (cited on pages 20, 26, and 122)

Hsu, W.-N.; Bolte, B.; Tsai, Y.-H. H.; Lakhotia, K.; Salakhutdinov, R.; and Mohamed, A., 2021. Hubert: Self-supervised speech representation learning by masked prediction of hidden units. *IEEE/ACM transactions on audio, speech, and language processing*, 29 (2021), 3451–3460. (cited on page 28)

Hu, E. J.; Shen, Y.; Wallis, P.; Allen-Zhu, Z.; Li, Y.; Wang, S.; Wang, L.; and Chen, W., 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, (2021). (cited on page 100)

Hu, E. J.; Shen, Y.; Wallis, P.; Allen-Zhu, Z.; Li, Y.; Wang, S.; Wang, L.; and Chen, W., 2022. LoRA: Low-rank adaptation of large language models. *The Tenth International Conference on Learning Representations, ICLR*, (2022). (cited on pages 36, 38, and 43)

Hu, H.; Zhu, X.; He, T.; Guo, D.; Zhang, B.; Wang, X.; Guo, Z.; Jiang, Z.; Hao, H.; Guo, Z.; et al., 2026. Qwen3-tts technical report. *arXiv preprint arXiv:2601.15621*, (2026). (cited on pages 3 and 17)

Huang, J.; Ren, Y.; Huang, R.; Yang, D.; Ye, Z.; Zhang, C.; Liu, J.; Yin, X.; Ma, Z.; and Zhao, Z., 2023a. Make-an-audio 2: Temporal-enhanced text-to-audio generation. *arXiv preprint arXiv:2305.18474*, (2023). (cited on pages 3, 17, 36, and 42)

Huang, J.; Ren, Y.; Huang, R.; Yang, D.; Ye, Z.; Zhang, C.; Liu, J.; Yin, X.; Ma, Z.; and Zhao, Z., 2023b. Make-an-audio 2: Temporal-enhanced text-to-audio generation. *arXiv preprint arXiv:2305.18474*, (2023). (cited on page 17)

Huang, P.-Y.; Xu, H.; Li, J.; Baevski, A.; Auli, M.; Galuba, W.; Metze, F.; and Feichtenhofer, C., 2022. Masked autoencoders that listen. *The Conference on Neural Information Processing Systems*, (2022). (cited on page 98)

Huang, Q.; Park, D. S.; Wang, T.; Denk, T. I.; Ly, A.; Chen, N.; Zhang, Z.; Zhang, Z.; Yu, J.; Frank, C.; et al., 2023c. Noise2music: Text-conditioned music generation with diffusion models. *arXiv preprint arXiv:2302.03917*, (2023). (cited on page 20)

Huang, R.; Huang, J.; Yang, D.; Ren, Y.; Liu, L.; Li, M.; Ye, Z.; Liu, J.; Yin, X.; and Zhao, Z., 2023d. Make-an-audio: Text-to-audio generation with prompt-enhanced diffusion models. In *International Conference on Machine Learning*, 13916–13932. PMLR. (cited on pages 17, 20, and 84)

Huang, R.; Huang, J.; Yang, D.; Ren, Y.; Liu, L.; Li, M.; Ye, Z.; Liu, J.; Yin, X.; and Zhao, Z., 2023e. Make-an-audio: Text-to-audio generation with prompt-enhanced diffusion models. *International Conference on Machine Learning*, (2023). (cited on pages 36 and 42)

Huang, Y.-S. and Yang, Y.-H., 2020. Pop music transformer: Beat-based modeling and generation of expressive pop piano compositions. In *Proceedings of the 28th ACM international conference on multimedia*, 1180–1188. (cited on page 2)

Huberman-Spiegelglas, I.; Kulikov, V.; and Michaeli, T., 2024. An edit friendly ddpm noise space: Inversion and manipulations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 12469–12478. (cited on page 114)

Hussain, S.; Neekhara, P.; Huang, J.; Li, J.; and Ginsburg, B., 2023. Ace-vc: Adaptive and controllable voice conversion using explicitly disentangled self-supervised speech representations. In *International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 1–5. IEEE. (cited on pages 23 and 84)

Hyrkas, J., 2021. Network modulation synthesis: New algorithms for generating musical audio using autoencoder networks. *arXiv preprint arXiv:2109.01948*, (2021). (cited on page 96)

Iashin, V. and Rahtu, E., 2021. Taming visually guided sound generation. In *British Machine Vision Conference*. BMVA Press. (cited on page 121)

Iashin, V.; Xie, W.; Rahtu, E.; and Zisserman, A., 2024. Synchformer: Efficient synchronization from sparse cues. In *IEEE International Conference on Acoustics, Speech and Signal Processing*. (cited on pages 33 and 54)

Jacobellis, D.; Cummings, D.; and Yadwadkar, N. J., 2024. Machine perceptual quality: Evaluating the impact of severe lossy compression on audio and image models. *arXiv preprint arXiv:2401.07957*, (2024). (cited on pages 104 and 122)

Jain, D. K.; Kumar, A.; Cai, L.; Singhal, S.; and Kumar, V., 2020. Att: Attention-based timbre transfer. In *2020 IJCNN*, 1–6. IEEE. (cited on page 24)

Jeong, M.; Kim, H.; Cheon, S. J.; Choi, B. J.; and Kim, N. S., 2021a. Diff-tts: A denoising diffusion model for text-to-speech. In *Proc. Interspeech 2021*, 3605–3609. (cited on pages 19 and 132)

Jeong, M.; Kim, H.; Cheon, S. J.; Choi, B. J.; and Kim, N. S., 2021b. Diff-tts: A denoising diffusion model for text-to-speech. In *Interspeech 2021, 22nd Annual Conference of the International Speech Communication Association, Brno, Czechia, 30 August - 3 September 2021*, 3605–3609. ISCA. doi:10.21437/Interspeech.2021-469. https://doi.org/10.21437/Interspeech.2021-469. (cited on page 60)

Jia, Y.; Chen, Y.; Zhao, J.; Zhao, S.; Zeng, W.; Chen, Y.; and Qin, Y., 2025. AudioEditor: A training-free diffusion-based audio editing framework. In *ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 1–5. IEEE. (cited on page 123)

Jiang, D.-N.; Lu, L.; Zhang, H.-J.; Tao, J.-H.; and Cai, L.-H., 2002. Music type classification by spectral contrast feature. In *ICME*. IEEE. (cited on page 108)

Jung, J.; Ahn, J.; Jung, C.; Nguyen, T. D.; Jang, Y.; and Chung, J. S., 2025. Voicedit: Dual-condition diffusion transformer for environment-aware speech synthesis. In *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), 2025*. (cited on pages 20, 49, 51, and 134)

Kamath, P.; Gupta, C.; and Nanayakkara, S., 2025. Morphfader: Enabling fine-grained controllable morphing with text-to-audio models. *IEEE ICASSP*, (2025). (cited on pages 25, 96, 106, and 134)

Kamath, P.; Morreale, F.; Bagaskara, P. L.; Wei, Y.; and Nanayakkara, S., 2024. Sound designer-generative ai interactions: Towards designing creative support tools for professional sound designers. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems*, 1–17. (cited on page 95)

Kawar, B.; Zada, S.; Lang, O.; Tov, O.; Chang, H.; Dekel, T.; Mosseri, I.; and Irani, M., 2023. Imagic: Text-based real image editing with diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 6007–6017. (cited on pages 85 and 114)

Kazazis, S.; Depalle, P.; and McAdams, S., 2016. Sound morphing by audio descriptors and parameter interpolation. In *Proceedings of the 19th International Conference on Digital Audio Effects (DAFx-16). Brno, Czech Republic*. (cited on pages 25, 95, and 97)

Kemppinen P., H. E., 2020. Timbrer: Learning musical timbre transfer in the frequency domain. https://github.com/harskish/Timbrer. (cited on page 104)

Kern, A. C. and Ellermeier, W., 2020. Audio in vr: Effects of a soundscape and movement-triggered step sounds on presence. *Frontiers in Robotics and AI*, 7 (2020), 20. (cited on page 35)

Kilgour, K.; Zuluaga, M.; Roblek, D.; and Sharifi, M., 2019. Fréchet audio distance: A reference-free metric for evaluating music enhancement algorithms. In *Proc. Interspeech 2019*, 2350–2354. (cited on pages 90, 103, and 121)

Kim, C. D.; Kim, B.; Lee, H.; and Kim, G., 2019a. Audiocaps: Generating captions for audios in the wild. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics*. (cited on pages 18, 40, 53, and 132)

Kim, J.; Kim, S.; Kong, J.; and Yoon, S., 2020. Glow-tts: A generative flow for text-to-speech via monotonic alignment search. *Advances in Neural Information Processing Systems*, 33 (2020), 8067–8077. (cited on pages 13, 19, 21, 60, 74, 77, and 132)

Kim, J.; Kong, J.; and Son, J., 2021. Conditional variational autoencoder with adversarial learning for end-to-end text-to-speech. In *International Conference on Machine Learning*, 5530–5540. PMLR. (cited on pages 23, 84, 85, 88, and 133)

Kim, J. W.; Bittner, R.; Kumar, A.; and Bello, J. P., 2019b. Neural music synthesis for flexible timbre control. In *International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*. (cited on pages 25 and 96)

Kingma, D.; Salimans, T.; Poole, B.; and Ho, J., 2021. Variational diffusion models. *Advances in neural information processing systems*, 34 (2021), 21696–21707. (cited on page 14)

Kingma, D. P. and Dhariwal, P., 2018. Glow: Generative flow with invertible 1x1 convolutions. *Advances in neural information processing systems*, 31 (2018). (cited on page 77)

Kingma, D. P. and Welling, M., 2013. Auto-encoding variational bayes. In *Int. Conf. on Learning Representations*. Banff, Canada. (cited on pages 13 and 98)

Kingma, D. P. and Welling, M., 2014. Auto-encoding variational bayes. In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*. http://arxiv.org/abs/1312.6114. (cited on page 21)

Kong, J.; Kim, J.; and Bae, J., 2020a. Hifi-gan: Generative adversarial networks for efficient and high fidelity speech synthesis. *Advances in neural information processing systems*, 33 (2020), 17022–17033. (cited on pages 19, 28, 89, and 98)

Kong, Q.; Cao, Y.; Iqbal, T.; Wang, Y.; Wang, W.; and Plumbley, M. D., 2020b. Panns: Large-scale pretrained audio neural networks for audio pattern recognition. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 28 (2020), 2880–2894. (cited on pages 29, 30, 40, 90, 103, and 104)

Kong, Z.; Goel, A.; Badlani, R.; Ping, W.; Valle, R.; and Catanzaro, B., 2024. Audio flamingo: A novel audio language model with few-shot learning and dialogue abilities. *arXiv preprint arXiv:2402.01831*, (2024). (cited on page 17)

Kreuk, F.; Synnaeve, G.; Polyak, A.; Singer, U.; Défossez, A.; Copet, J.; Parikh, D.; Taigman, Y.; and Adi, Y., 2022. AudioGen: Textually guided audio generation. *arXiv preprint arXiv:2209.15352*, (2022). (cited on pages 17 and 20)

Kreuk, F.; Synnaeve, G.; Polyak, A.; Singer, U.; Défossez, A.; Copet, J.; Parikh, D.; Taigman, Y.; and Adi, Y., 2023. Audiogen: Textually guided audio generation. In *International Conference on Learning Representations (ICLR)*. (cited on pages 2, 4, 11, 18, 36, and 132)

Krishnamurthy, B. and Rattani, A., 2026. Voxmorph: Scalable zero-shot voice identity morphing via disentangled embeddings. *arXiv preprint arXiv:2601.20883*, (2026). (cited on page 95)

Kuan, C.-Y.; Li, C. A.; Hsu, T.-Y.; Lin, T.-Y.; Chung, H.-L.; Chang, K.-W.; Chang, S.-y.; and Lee, H.-y., 2023. Towards general-purpose text-instruction-guided voice conversion. *arXiv preprint arXiv:2309.14324*, (2023). (cited on pages 23, 84, 89, 90, and 133)

Kubichek, R., 1993. Mel-cepstral distance measure for objective speech quality assessment. In *Proceedings of IEEE pacific rim conference on communications computers and signal processing*, vol. 1, 125–128. IEEE. (cited on pages 27 and 76)

Kumar, K.; Kumar, R.; De Boissiere, T.; Gestin, L.; Teoh, W. Z.; Sotelo, J.; De Brebisson, A.; Bengio, Y.; and Courville, A. C., 2019. MelGan: Generative adversarial networks for conditional waveform synthesis. In *The Conference on Neural Information Processing Systems*. (cited on pages 28 and 40)

Kumari, N.; Zhang, B.; Zhang, R.; Shechtman, E.; and Zhu, J.-Y., 2023. Multi-concept customization of text-to-image diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 1931–1941. (cited on pages 20 and 119)

Kundu, S.; Singh, S.; and Iwahori, Y., 2024. Emotion-guided image to music generation. *arXiv preprint arXiv:2410.22299*, (2024). (cited on page 20)

Lai, C.-H.; Song, Y.; Kim, D.; Mitsufuji, Y.; and Ermon, S., 2025. The principles of diffusion models. *arXiv preprint arXiv:2510.21890*, (2025). (cited on pages 13 and 114)

Lan, G. L.; Shi, B.; Ni, Z.; Srinivasan, S.; Kumar, A.; Ellis, B.; Kant, D.; Nagaraja, V.; Chang, E.; Hsu, W.-N.; et al., 2024. High fidelity text-guided music generation and editing via single-stage flow matching. *arXiv preprint arXiv:2407.03648*, (2024). (cited on page 20)

Latif, S.; Shoukat, M.; Shamshad, F.; Usama, M.; Ren, Y.; Cuayáhuitl, H.; Wang, W.; Zhang, X.; Togneri, R.; Cambria, E.; et al., 2023. Sparks of large audio models: A survey and outlook. *arXiv preprint arXiv:2308.12792*, (2023). (cited on page 16)

Le, M.; Vyas, A.; Shi, B.; Karrer, B.; Sari, L.; Moritz, R.; Williamson, M.; Manohar, V.; Adi, Y.; Mahadeokar, J.; et al., 2023. Voicebox: Text-guided multilingual universal speech generation at scale. *Advances in neural information processing systems*, 36 (2023), 14005–14034. (cited on page 20)

Le Lan, G.; Shi, B.; Ni, Z.; Srinivasan, S.; Kumar, A.; Ellis, B.; Kant, D.; Nagaraja, V. K.; Chang, E.; Hsu, W.-N.; et al., 2024. High fidelity text-guided music editing via single-stage flow matching. In *Audio Imagination: The Conference on Neural Information Processing Systems 2024 Workshop AI-Driven Speech, Music, and Sound Generation*. (cited on pages 20 and 123)

Le Vaillant, G. and Dutoit, T., 2023a. Interpolation of synthesizer presets using timbre-regularized auto-encoders. *Authorea Preprints*, (2023). (cited on page 25)

Le Vaillant, G. and Dutoit, T., 2023b. Synthesizer preset interpolation using transformer auto-encoders. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 1–5. IEEE. (cited on page 25)

Le Vaillant, G. and Dutoit, T., 2024. Latent space interpolation of synthesizer parameters using timbre-regularized auto-encoders. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, (2024). (cited on page 25)

Lee, J.; Im, J.; Kim, D.; and Nam, J., 2024a. Video-foley: Two-stage video-to-sound generation via temporal event condition for foley sound. *arXiv preprint arXiv:2408.11915*, (2024). (cited on pages 4, 18, and 48)

Lee, Y.; Yeon, I.; Nam, J.; and Chung, J. S., 2024b. Voiceldm: Text-to-speech with environmental context. In *IEEE International Conference on Acoustics, Speech and Signal Processing*. (cited on pages 4, 20, 49, 51, 53, 55, 56, and 134)

Li, B.; Zhao, Y.; Zhelun, S.; and Sheng, L., 2022. Danceformer: Music conditioned 3d dance generation with parametric motion transformer. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 1272–1279. (cited on page 60)

Li, J.; Tu, W.; and Xiao, L., 2023a. Freevc: Towards high-quality text-free one-shot voice conversion. In *International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 1–5. IEEE. (cited on pages 3, 23, 84, 86, 89, 90, 92, and 133)

Li, N.; Liu, S.; Liu, Y.; Zhao, S.; Liu, M.; and Zhou, M., 2018. Close to human quality TTS with transformer. *CoRR*, abs/1809.08895 (2018). http://arxiv.org/abs/1809.08895. (cited on page 60)

Li, P. P.; Chen, B.; Yao, Y.; Wang, Y.; Wang, A.; and Wang, A., 2024a. Jen-1: Text-guided universal music generation with omnidirectional diffusion models. In *2024 IEEE Conference on Artificial Intelligence (CAI)*, 762–769. IEEE. (cited on page 20)

Li, S.; Zhang, Y.; Tang, F.; Ma, C.; Dong, W.; and Xu, C., 2024b. Music style transfer with time-varying inversion of diffusion models. In *Proceedings of the AAAI*, vol. 38, 547–555. (cited on page 24)

Li, W. and Wei, T.-J., 2022. Asgan-vc: One-shot voice conversion with additional style embedding and generative adversarial networks. In *APSIPA ASC*. IEEE. (cited on pages 23 and 84)

Li, X.; Liu, J.; Liang, Y.; Niu, Z.; Chen, W.; and Chen, X., 2025. Meanaudio: Fast and faithful text-to-audio generation with mean flows. *arXiv preprint arXiv:2508.06098*, (2025). (cited on page 132)

Li, Y.; Tagliasacchi, M.; Rybakov, O.; Ungureanu, V.; and Roblek, D., 2021. Real-time speech frequency bandwidth extension. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 691–695. IEEE. (cited on pages 18 and 36)

Li, Y. A.; Han, C.; and Mesgarani, N., 2023b. Styletts-vc: One-shot voice conversion by knowledge transfer from style-based tts models. In *IEEE SLT*, 920–927. IEEE. (cited on pages 3, 23, 84, and 133)

Li, Z.; Zhao, B.; and Yuan, Y., 2024c. Tas: Personalized text-guided audio spatialization. In *Proceedings of the 32nd ACM International Conference on Multimedia*, 9029–9037. (cited on page 137)

Liang, J.; Yuan, Y.; Jia, D.; Zhuang, X.; Liu, Z.; Chen, Y.; Chen, Z.; Wang, Y.; and Wang, Y., 2024. AudioMorphix: Training-free audio editing with diffusion probabilistic models. https://openreview.net/forum?id=a8dQutiF9E. (cited on page 123)

Liang, Y.; Yang, K.; Liu, F.; Li, A.; Li, X.; and Zheng, C., 2025. Lightl2s: Ultra-low complexity lip-to-speech synthesis for multi-speaker scenarios. In *Proc. Interspeech 2025*, 3783–3787. (cited on page 48)

Lin, Y.-B.; Lin, K.; Yang, Z.; Li, L.; Wang, J.; Lin, C.-C.; Wang, X.; Bertasius, G.; and Wang, L., 2025. Zero-shot audio-visual editing via cross-modal delta denoising. *arXiv preprint arXiv:2503.20782*, (2025). (cited on page 117)

Liu, H.; Chen, Z.; Yuan, Y.; Mei, X.; Liu, X.; Mandic, D.; Wang, W.; and Plumbley, M. D., 2023a. AudioLDM: text-to-audio generation with latent diffusion models. In *Proceedings of the 40th International Conference on Machine Learning*, 21450–21474. (cited on pages 17, 18, 25, 36, 42, 84, and 90)

Liu, H.; Huang, R.; Liu, Y.; Cao, H.; Wang, J.; Cheng, X.; Zheng, S.; and Zhao, Z., 2024a. AudioLCM: Efficient and high-quality text-to-audio generation with minimal inference steps. In *Proceedings of the 32nd ACM International Conference on Multimedia*, 7008–7017. (cited on page 132)

Liu, H.; Wang, J.; Li, X.; Huang, R.; Liu, Y.; Xu, J.; and Zhao, Z., 2024b. MEDIC: Zero-shot music editing with disentangled inversion control. *arXiv preprint arXiv:2407.13220*, (2024). (cited on pages 23, 114, 122, 123, 124, and 135)

Liu, H.; Yuan, Y.; Liu, X.; Mei, X.; Kong, Q.; Tian, Q.; Wang, Y.; Wang, W.; Wang, Y.; and Plumbley, M. D., 2024c. AudioLDM2: Learning holistic audio generation with self-supervised pretraining. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, (2024). (cited on pages 3, 17, 98, 107, 122, and 123)

Liu, J.; Li, C.; Ren, Y.; Chen, F.; and Zhao, Z., 2022. Diffsinger: Singing voice synthesis via shallow diffusion mechanism. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 11020–11028. (cited on pages 21, 60, 77, and 132)

Liu, S.; Cao, Y.; Wang, D.; Wu, X.; Liu, X.; and Meng, H., 2021. Any-to-many voice conversion with location-relative sequence-to-sequence modeling. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 29 (2021), 1717–1728. (cited on page 86)

Liu, S.; Hussain, A. S.; Sun, C.; and Shan, Y., 2023b. m^2 ugen: Multi-modal music understanding and generation with the power of large language models. *arXiv preprint arXiv:2311.11255*, (2023). (cited on page 23)

Liu, X.; Su, K.; and Shlizerman, E., 2024d. Tell what you hear from what you see-video to audio generation through text. *Advances in Neural Information Processing Systems*, (2024). (cited on pages 18, 48, 52, 54, 55, and 134)

Liu, Y. and Jin, C., 2024a. ICGAN: An implicit conditioning method for interpretable feature control of neural audio synthesis. *arXiv preprint arXiv:2406.07131*, (2024). (cited on page 17)

Liu, Y. and Jin, C., 2024b. Simi-SFX: A similarity-based conditioning method for controllable sound effect synthesis. *arXiv preprint arXiv:2412.18710*, (2024). (cited on page 17)

Liu, Y.; Ott, M.; Goyal, N.; Du, J.; Joshi, M.; Chen, D.; Levy, O.; Lewis, M.; Zettlemoyer, L.; and Stoyanov, V., 2020. Roberta: A robustly optimized bert pretraining approach. *International Conference on Learning Representations (ICLR)*, (2020). (cited on page 87)

Logan, B. et al., 2000. Mel frequency cepstral coefficients for music modeling. In *Ismir*. (cited on pages 103 and 108)

Loshchilov, I. and Hutter, F., 2019. Decoupled weight decay regularization. *International Conference on Learning Representations*, (2019). (cited on page 54)

Lu, H.; Wu, Z.; Wu, X.; Li, X.; Kang, S.; Liu, X.; and Meng, H., 2021. Vaenar-tts: Variational auto-encoder based non-autoregressive text-to-speech synthesis. *arXiv preprint arXiv:2107.03298*, (2021). (cited on pages 19 and 77)

Lu, J.; Clark, C.; Lee, S.; Zhang, Z.; Khosla, S.; Marten, R.; Hoiem, D.; and Kembhavi, A., 2024. Unified-io 2: Scaling autoregressive multimodal models with vision language audio and action. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 26439–26455. (cited on page 17)

Lüddecke, T. and Ecker, A., 2022. Image segmentation using text and image prompts. In *The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 7086–7096. IEEE/CVF. (cited on page 87)

Luo, S.; Yan, C.; Hu, C.; and Zhao, H., 2023. Diff-foley: Synchronized video-to-audio synthesis with latent diffusion models. *The Conference on Neural Information Processing Systems*, (2023). (cited on pages 4, 18, 48, and 134)

Luo, Y.-J.; Agres, K.; and Herremans, D., 2019. Learning disentangled representations of timbre and pitch for musical instrument sounds using gaussian mixture variational autoencoders. *arXiv preprint arXiv:1906.08152*, (2019). (cited on page 25)

Ma, X.; Zhou, C.; Li, X.; Neubig, G.; and Hovy, E., 2019. FlowSeq: Non-autoregressive conditional sequence generation with generative flow. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 4282–4292. Association for Computational Linguistics, Hong Kong, China. doi: 10.18653/v1/D19-1437. https://aclanthology.org/D19-1437. (cited on page 70)

Majumder, N.; Hung, C.-Y.; Ghosal, D.; Hsu, W.-N.; Mihalcea, R.; and Poria, S., 2024. Tango 2: Aligning diffusion-based text-to-audio generations through direct preference optimization. In *Proceedings of the 32nd ACM International Conference on Multimedia*, 564–572. (cited on page 17)

Mancusi, M.; Halychanskyi, Y.; Cheuk, K. W.; Moliner, E.; Lai, C.-H.; Uhlich, S.; Koo, J.; Martínez-Ramírez, M. A.; Liao, W.-H.; Fabbro, G.; et al., 2025. Latent diffusion bridges for unsupervised musical audio timbre transfer. In *ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 1–5. IEEE. (cited on page 122)

Manocha, P.; Jin, Z.; Zhang, R.; and Finkelstein, A., 2021. Cdpam: Contrastive learning for perceptual audio similarity. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 196–200. IEEE. (cited on pages 104 and 122)

Manor, H. and Michaeli, T., 2024a. Zero-shot unsupervised and text-based audio editing using ddpm inversion. In *International Conference on Machine Learning*, 34603–34629. PMLR. (cited on pages 23, 54, 114, 121, 123, and 126)

Manor, H. and Michaeli, T., 2024b. Zero-shot unsupervised and text-based audio editing using ddpm inversion. In *Forty-first International Conference on Machine Learning*. (cited on page 135)

Mariani, G.; Tallini, I.; Postolache, E.; Mancusi, M.; Cosmo, L.; and Rodolà, E., 2023. Multi-source diffusion models for simultaneous music generation and separation. *arXiv preprint arXiv:2302.02257*, (2023). (cited on pages 23 and 114)

McAdams, S., 2013. Musical timbre perception. *The psychology of music*, 3 (2013). (cited on pages 95, 104, and 105)

McAdams, S., 2019. The perceptual representation of timbre. In *Timbre: Acoustics, perception, and cognition*, 23–57. Springer. (cited on page 95)

McAdams, S. and Goodchild, M., 2017. Musical structure: Sound and timbre. In *The Routledge companion to music cognition*. (cited on page 104)

McAdams, S.; Winsberg, S.; Donnadieu, S.; De Soete, G.; and Krimphoff, J., 1995. Perceptual scaling of synthesized musical timbres: Common dimensions, specificities, and latent subject classes. *Psychological research*, (1995). (cited on pages 104 and 105)

McAuliffe, M.; Socolof, M.; Mihuc, S.; Wagner, M.; and Sonderegger, M., 2017a. Montreal forced aligner: Trainable text-speech alignment using kaldi. In *Interspeech*, vol. 2017, 498–502. (cited on page 19)

McAuliffe, M.; Socolof, M.; Mihuc, S.; Wagner, M.; and Sonderegger, M., 2017b. the montreal forced aligner: Trainable text-speech alignment using kaldi. In *Interspeech 2017, 18th Annual Conference of the International Speech Communication Association, Stockholm, Sweden, August 20-24, 2017*, 498–502. ISCA. http://www.isca-speech.org/archive/Interspeech_2017/abstracts/1386.html. (cited on pages 21 and 60)

Mehrish, A.; Majumder, N.; Bharadwaj, R.; Mihalcea, R.; and Poria, S., 2023. A review of deep learning techniques for speech processing. *Information Fusion*, (2023), 101869. (cited on page 76)

Meng, C.; Song, Y.; Song, J.; Wu, J.; Zhu, J.-Y.; and Ermon, S., 2021. Sdedit: Image synthesis and editing with stochastic differential equations. *arXiv preprint arXiv:2108.01073*, (2021). (cited on page 123)

Mensch, A. and Blondel, M., 2018. Differentiable dynamic programming for structured prediction and attention. In *International Conference on Machine Learning*, 3462–3471. PMLR. (cited on pages 21, 74, and 79)

Mittag, G.; Naderi, B.; Chehadi, A.; and Möller, S., 2021. Nisqa: A deep cnn-self-attention model for multidimensional speech quality prediction with crowdsourced datasets. *InterSpeech*, (2021). (cited on pages 29 and 90)

Mohamed, S.; Rosca, M.; Figurnov, M.; and Mnih, A., 2020. Monte carlo gradient estimation in machine learning. *J. Mach. Learn. Res.*, 21, 132 (2020), 1–62. (cited on page 71)

Mohammadi, S. H. and Kain, A., 2017. An overview of voice conversion systems. *Speech Communication*, 88 (2017), 65–82. (cited on page 84)

Mokrỳ, O. and Rajmic, P., 2020. Audio inpainting: Revisited and reweighted. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 28 (2020), 2906–2918. (cited on page 22)

Nam, H.; Kwon, G.; Park, G. Y.; and Ye, J. C., 2024. Contrastive denoising score for text-guided latent diffusion image editing. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 9192–9201. (cited on pages 117 and 120)

Niu, X.; Cheuk, K. W.; Zhang, J.; Murata, N.; Lai, C.-H.; Mancusi, M.; Choi, W.; Fabbro, G.; Liao, W.-H.; Martin, C. P.; et al., 2026a. Steermusic: Enhanced musical consistency for zero-shot text-guided and personalized music editing. In *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 40, 2000–2010. (cited on page 9)

Niu, X.; Ma, J.; Harper-Harris, D.; Zhang, X.; Martin, C. P.; and Zhang, J., 2026b. Beyond video-to-sfx: Video to audio synthesis with environmentally aware speech. In *ICASSP 2026-2026 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 22697–22701. IEEE. (cited on page 9)

Niu, X. and Martin, C. P., 2022. Spatial-temporal-class attention network for acoustic scene classification. In *2022 IEEE International Conference on Multimedia and Expo (ICME)*, 1–6. IEEE. (cited on page 129)

Niu, X.; Walder, C.; Zhang, J.; and Martin, C. P., 2024a. Latent optimal paths by gumbel propagation for variational bayesian dynamic programming. In *Proceedings of the 41st International Conference on Machine Learning*, 38316–38343. (cited on page 9)

Niu, X.; Zhang, J.; and Martin, C. P., 2024b. HybridVC: Efficient voice style conversion with text and audio prompts. In *Proc. Interspeech 2024*, 4368–4372. (cited on page 9)

Niu, X.; Zhang, J.; and Martin, C. P., 2024c. SoundMorpher: Perceptually-uniform sound morphing with diffusion model. *arXiv preprint arXiv:2410.02144*, (2024). (cited on pages 9 and 115)

Niu, X.; Zhang, J.; Walder, C.; and Martin, C. P., 2024d. SoundLOCD: An efficient conditional discrete contrastive latent diffusion model for text-to-sound generation. In *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 261–265. IEEE. (cited on pages 9, 84, and 114)

Novack, Z.; McAuley, J.; Berg-Kirkpatrick, T.; and Bryan, N., 2024a. Ditto-2: Distilled diffusion inference-time t-optimization for music generation. *arXiv preprint arXiv:2405.20289*, (2024). (cited on page 20)

Novack, Z.; McAuley, J.; Berg-Kirkpatrick, T.; and Bryan, N. J., 2024b. Ditto: Diffusion inference-time t-optimization for music generation. *arXiv preprint arXiv:2401.12179*, (2024). (cited on page 20)

Osaka, N., 1995. Timbre interpolation of sounds using a sinusoidal model. *Proc. of ICMC, 1995*, (1995). (cited on pages 24 and 96)

Oxenham, A. J., 2018. How we hear: The perception and neural coding of sound. *Annual review of psychology*, 69, 1 (2018), 27–50. (cited on page 1)

O'Callaghan, C., 2009. Auditory perception. *Neuroscience*, (2009). (cited on page 1)

Paissan, F.; Della Libera, L.; Wang, Z.; Ravanelli, M.; Smaragdis, P.; and Subakan, C., 2023. Audio editing with non-rigid text prompts. *arXiv preprint arXiv:2310.12858*, (2023). (cited on page 114)

Park, H. J.; Yang, S. W.; Kim, J. S.; Shin, W.; and Han, S. W., 2023. Triaan-vc: Triple adaptive attention normalization for any-to-any voice conversion. In *International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 1–5. IEEE. (cited on pages 23 and 84)

Peng, K.; Ping, W.; Song, Z.; and Zhao, K., 2020. Non-autoregressive neural text-to-speech. In *International conference on machine learning*, 7586–7598. PMLR. (cited on page 60)

Petrov, S. and Klein, D., 2007. Discriminative log-linear grammars with latent variables. In *Advances in Neural Information Processing Systems 20, Proceedings of the Twenty-First Annual Conference on Neural Information Processing Systems, Vancouver, British Columbia, Canada, December 3-6, 2007*, 1153–1160. Curran Associates, Inc. (cited on page 21)

Phan, H.; Chén, O. Y.; Pham, L.; Koch, P.; De Vos, M.; McLoughlin, I.; and Mertins, A., 2019. Spatio-temporal attention pooling for audio scene classification. *arXiv preprint arXiv:1904.03543*, (2019). (cited on page 129)

Piczak, K. J., 2015. ESC: Dataset for Environmental Sound Classification. In *Proceedings of the 23rd Annual ACM Conference on Multimedia*. (cited on pages 40, 106, and 132)

Platt, J., 2000. Probabilistic outputs for support vector machines and comparison to regularized likelihood methods. In *Advances in Large Margin Classifiers*. (cited on page 79)

Plitsis, M.; Kouzelis, T.; Paraskevopoulos, G.; Katsouros, V.; and Panagakis, Y., 2024. Investigating personalization methods in text to music generation. In *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 1081–1085. IEEE. (cited on pages 3, 20, 24, 115, 119, 124, 127, and 135)

Poole, B.; Jain, A.; Barron, J. T.; and Mildenhall, B., 2022. Dreamfusion: Text-to-3d using 2d diffusion. *arXiv preprint arXiv:2209.14988*, (2022). (cited on page 116)

Popov, V.; Vovk, I.; Gogoryan, V.; Sadekova, T.; and Kudinov, M., 2021. Grad-tts: A diffusion probabilistic model for text-to-speech. In *International Conference on Machine Learning*, 8599–8608. PMLR. (cited on pages 19 and 60)

Prenger, R.; Valle, R.; and Catanzaro, B., 2019. Waveglow: A flow-based generative network for speech synthesis. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 3617–3621. IEEE. (cited on pages 13 and 19)

Primavera, A.; Piazza, F.; and Reiss, J. D., 2012. Audio morphing for percussive hybrid sound generation. In *45th International Conference Applications of Time-Frequency Processing in Audio*. (cited on page 24)

Qamar, I. P.; Stawarz, K.; Robinson, S.; Goguey, A.; Coutrix, C.; and Roudaut, A., 2020. Morphino: a nature-inspired tool for the design of shape-changing interfaces. In *Proceedings of the 2020 ACM Designing Interactive Systems Conference*, 1943–1958. (cited on page 96)

Qian, K.; Zhang, Y.; Chang, S.; Yang, X.; and Hasegawa-Johnson, M., 2019. Autovc: Zero-shot voice style transfer with only autoencoder loss. In *The International Conference on Machine Learning (ICML)*, 5210–5219. PMLR. (cited on pages 23 and 84)

Rabiner, L. R., 1989. A tutorial on hidden markov models and selected applications in speech recognition. *Proc. IEEE*, 77, 2 (1989), 257–286. doi:10.1109/5.18626. https://doi.org/10.1109/5.18626. (cited on page 21)

Radford, A.; Kim, J. W.; Hallacy, C.; Ramesh, A.; Goh, G.; Agarwal, S.; Sastry, G.; Askell, A.; Mishkin, P.; Clark, J.; et al., 2021. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, 8748–8763. PmLR. (cited on pages 36 and 43)

Radford, A.; Kim, J. W.; Xu, T.; Brockman, G.; McLeavey, C.; and Sutskever, I., 2023. Robust speech recognition via large-scale weak supervision. In *The International Conference on Machine Learning (ICML)*. (cited on page 56)

Radford, A.; Wu, J.; Child, R.; Luan, D.; Amodei, D.; Sutskever, I.; et al., 2019. Language models are unsupervised multitask learners. *OpenAI blog*, (2019). (cited on page 98)

Raffel, C.; Shazeer, N.; Roberts, A.; Lee, K.; Narang, S.; Matena, M.; Zhou, Y.; Li, W.; and Liu, P. J., 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *JMLR*, (2020). (cited on pages 18, 36, and 43)

Rekimoto, J., 2023. Wesper: Zero-shot and realtime whisper to normal voice conversion for whisper-based speech interactions. In *CHI*, 1–12. (cited on pages 20, 23, and 84)

Ren, Y.; Hu, C.; Tan, X.; Qin, T.; Zhao, S.; Zhao, Z.; and Liu, T.-Y., 2020. Fastspeech 2: Fast and high-quality end-to-end text to speech. *arXiv preprint arXiv:2006.04558*, (2020). (cited on pages 19, 21, 60, 76, and 132)

Ren, Y.; Ruan, Y.; Tan, X.; Qin, T.; Zhao, S.; Zhao, Z.; and Liu, T.-Y., 2019. Fastspeech: Fast, robust and controllable text to speech. *Advances in Neural Information Processing Systems*, 32 (2019). (cited on pages 2, 19, 26, 60, and 132)

Roma, G.; Green, O.; and Tremblay, P. A., 2020. Audio morphing using matrix decomposition and optimal transport. In *Proceedings of DAFx*. (cited on pages 24, 96, and 134)

Rombach, R.; Blattmann, A.; Lorenz, D.; Esser, P.; and Ommer, B., 2022a. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 10684–10695. (cited on pages 23, 84, and 86)

Rombach, R.; Blattmann, A.; Lorenz, D.; Esser, P.; and Ommer, B., 2022b. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 10684–10695. (cited on page 98)

Rouard, S.; San Roman, R.; Adi, Y.; and Roebel, A., 2025. 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)*, 1–5. IEEE. (cited on pages 11 and 23)

Ruiz, N.; Li, Y.; Jampani, V.; Pritch, Y.; Rubinstein, M.; and Aberman, K., 2023. Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 22500–22510. (cited on pages 20, 115, 119, and 124)

Saito, K.; Kim, D.; Shibuya, T.; Lai, C.-H.; Zhong, Z.; Takida, Y.; and Mitsufuji, Y., 2025. Soundctm: Unifying score-based and consistency models for full-band text-to-sound generation. In *The Thirteenth International Conference on Learning Representations*. (cited on page 20)

Sakoe, H. and Chiba, S., 1978. Dynamic programming algorithm optimization for spoken word recognition. *IEEE transactions on acoustics, speech, and signal processing*, 26, 1 (1978), 43–49. (cited on page 74)

Schafer, R. M., 1993. *The soundscape: Our sonic environment and the tuning of the world*. Simon and Schuster. (cited on page 95)

Sethares, W. A. and Bucklew, J. A., 2015. Kernel techniques for generalized audio crossfades. *Cogent Mathematics*, (2015), 1102116. (cited on page 97)

Shen, J.; Pang, R.; Weiss, R. J.; Schuster, M.; Jaitly, N.; Yang, Z.; Chen, Z.; Zhang, Y.; Wang, Y.; Ryan, R.; Saurous, R. A.; Agiomyrgiannakis, Y.; and Wu, Y., 2018a. Natural TTS synthesis by conditioning wavenet on MEL spectrogram predictions. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2018, Calgary, AB, Canada, April 15-20, 2018*, 4779–4783. IEEE. doi:10.1109/ICASSP.2018.8461368. https://doi.org/10.1109/ICASSP.2018.8461368. (cited on page 77)

Shen, J.; Pang, R.; Weiss, R. J.; Schuster, M.; Jaitly, N.; Yang, Z.; Chen, Z.; Zhang, Y.; Wang, Y.; Skerrv-Ryan, R.; et al., 2018b. Natural tts synthesis by conditioning wavenet on mel spectrogram predictions. In *2018 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, 4779–4783. IEEE. (cited on page 19)

Sheng, Z.; Ai, Y.; Liu, L.-J.; Pan, J.; and Ling, Z.-H., 2024. Voice attribute editing with text prompt. *arXiv preprint arXiv:2404.08857*, (2024). (cited on page 25)

Siddiq, S., 2015. Real-time morphing of impact sounds. In *Audio Engineering Society Convention 139*. ASE. (cited on page 96)

Slaney, M.; Covell, M.; and Lassiter, B., 1996. Automatic audio morphing. In *International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, vol. 2, 1001–1004. (cited on page 97)

Song, J.; Meng, C.; and Ermon, S., 2021a. Denoising diffusion implicit models. In *International Conference on Learning Representations*. (cited on pages 14, 38, 100, 114, 123, 124, and 132)

Song, Y.; Sohl-Dickstein, J.; Kingma, D. P.; Kumar, A.; Ermon, S.; and Poole, B., 2021b. Score-based generative modeling through stochastic differential equations. *9th International Conference on Learning Representations, ICLR*, (2021). (cited on pages 18 and 36)

Stevens, S. S.; Volkmann, J.; and Newman, E. B., 1937. A scale for the measurement of the psychological magnitude pitch. *The journal of the acoustical society of america*, (1937). (cited on page 108)

Story, B. H., 2019. History of speech synthesis. In *The Routledge handbook of phonetics*, 9–33. Routledge. (cited on page 11)

Su, K.; Liu, X.; and Shlizerman, E., 2024. From vision to audio and beyond: a unified model for audio-visual representation and generation. In *Proceedings of the 41st International Conference on Machine Learning*, 46804–46822. (cited on pages 52 and 134)

Sun, L.; Kang, S.; Li, K.; and Meng, H., 2015. Voice conversion using deep bidirectional long short-time memory based recurrent neural networks. In *International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 4869–4873. IEEE. (cited on pages 23 and 84)

Sun, P.; Cheng, S.; Li, X.; Ye, Z.; Liu, H.; Zhang, H.; Xue, W.; and Guo, Y., 2025. Both ears wide open: Towards language-driven spatial audio generation. In *The Thirteenth International Conference on Learning Representations*. (cited on page 137)

Sun, Q.; Fang, Y.; Wu, L.; Wang, X.; and Cao, Y., 2023. Eva-clip: Improved training techniques for clip at scale. *arXiv preprint arXiv:2303.15389*, (2023). (cited on page 54)

Tan, H. H.; Luo, Y.-J.; and Herremans, D., 2020. Generative modelling for controllable audio synthesis of expressive piano performance. *arXiv preprint arXiv:2006.09833*, (2020). (cited on page 25)

Tan, X.; Qin, T.; Bian, J.; Liu, T.-Y.; and Bengio, Y., 2024. Regeneration learning: A learning paradigm for data generation. In *Proceedings of the AAAI*, vol. 38, 22614–22622. (cited on page 98)

Tellman, E.; Haken, L.; and Holloway, B., 1995. Timbre morphing of sounds with unequal numbers of features. *Journal of the Audio Engineering Society*, 43, 9 (1995), 678–689. (cited on pages 24 and 96)

Tian, W.; Zhu, X.; Liu, H.; Zhao, Z.; Chen, Z.; Ding, C.; Di, X.; Zheng, J.; and Xie, L., 2025a. Dualdub: Video-to-soundtrack generation via joint speech and background audio synthesis. *arXiv preprint arXiv:2507.10109*, (2025). (cited on page 49)

Tian, Z.; Jin, Y.; Liu, Z.; Yuan, R.; Tan, X.; Chen, Q.; Xue, W.; and Guo, Y., 2025b. Audiox: Diffusion transformer for anything-to-audio generation. *arXiv preprint arXiv:2503.10522*, (2025). (cited on page 17)

Toda, T.; Black, A. W.; and Tokuda, K., 2007. Voice conversion based on maximum-likelihood estimation of spectral parameter trajectory. *IEEE Transactions on Audio, Speech, and Language Processing*, 15, 8 (2007), 2222–2235. (cited on pages 23 and 84)

Tsai, F.-D.; Wu, S.-L.; Kim, H.; Chen, B.-Y.; Cheng, H.-C.; and Yang, Y.-H., 2024. Audio prompt adapter: Unleashing music editing abilities for text-to-music with lightweight finetuning. *ISMIR*, (2024). (cited on page 23)

Valle, R.; Badlani, R.; Kong, Z.; Lee, S.-g.; Goel, A.; Kim, S.; Santos, J. F.; Dai, S.; Gururani, S.; Aljafari, A.; et al., 2025. Fugatto 1: Foundational generative audio transformer opus 1. In *The Thirteenth International Conference on Learning Representations*. (cited on pages 3 and 17)

van den Oord, A.; Dieleman, S.; Zen, H.; Simonyan, K.; Vinyals, O.; Graves, A.; Kalchbrenner, N.; Senior, A.; Kavukcuoglu, K.; et al., 2016. Wavenet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499*, 12 (2016), 1. (cited on pages 2, 3, 11, 12, 16, 19, and 26)

van den Oord, A.; Vinyals, O.; et al., 2017a. Neural discrete representation learning. In *The Conference on Neural Information Processing Systems*. (cited on pages 13, 16, 28, and 36)

van den Oord, A.; Vinyals, O.; et al., 2017b. Neural discrete representation learning. *Advances in neural information processing systems*, 30 (2017). (cited on pages 28 and 79)

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I., 2017. Attention is all you need. *Advances in neural information processing systems*, 30 (2017). (cited on pages 21 and 77)

Verdu, S. and Poor, H. V., 1987. Abstract dynamic programming models under commutativity conditions. *SIAM Journal on Control and Optimization*, 25, 4 (1987), 990–1006. (cited on page 60)

Viertola, I.; Iashin, V.; and Rahtu, E., 2025. Temporally aligned audio for video with autoregression. In *2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE. (cited on pages 18 and 48)

Wang, C.; Chen, S.; Wu, Y.; Zhang, Z.; Zhou, L.; Liu, S.; Chen, Z.; Liu, Y.; Wang, H.; Li, J.; et al., 2023a. Neural codec language models are zero-shot text to speech synthesizers. *URL: https://arxiv. org/abs/2301.02111. doi: doi*, 10 (2023). (cited on pages 2, 4, and 20)

Wang, C.; Chen, S.; Wu, Y.; Zhang, Z.; Zhou, L.; Liu, S.; Chen, Z.; Liu, Y.; Wang, H.; Li, J.; et al., 2023b. Neural codec language models are zero-shot text to speech synthesizers. *arXiv preprint arXiv:2301.02111*, (2023). (cited on page 84)

Wang, D.; Deng, L.; Yeung, Y. T.; Chen, X.; Liu, X.; and Meng, H., 2021. Vqmivc: Vector quantization and mutual information-based unsupervised speech representation disentanglement for one-shot voice conversion. *arXiv preprint arXiv:2106.10132*, (2021). (cited on pages 23 and 84)

Wang, H.; Ma, J.; Pascual, S.; Cartwright, R.; and Cai, W., 2024a. V2a-mapper: A lightweight solution for vision-to-audio generation by connecting foundation models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, 15492–15501. (cited on page 18)

Wang, L.; Wang, J.; Qiang, C.; Deng, F.; Zhang, C.; Zhang, D.; and Gai, K., 2025a. Audiogen-omni: A unified multimodal diffusion transformer for video-synchronized audio, speech, and song generation. *arXiv preprint arXiv:2508.00733*, (2025). (cited on page 17)

Wang, Y.; Guo, W.; Huang, R.; Huang, J.; Wang, Z.; You, F.; Li, R.; and Zhao, Z., 2024b. Frieren: Efficient video-to-audio generation network with rectified flow matching. *The Conference on Neural Information Processing Systems*, (2024). (cited on pages 4, 18, and 48)

Wang, Y.; Ju, Z.; Tan, X.; He, L.; Wu, Z.; Bian, J.; et al., 2023c. Audit: Audio editing by following instructions with latent diffusion models. *Advances in Neural Information Processing Systems*, 36 (2023), 71340–71357. (cited on pages 22 and 23)

Wang, Y.; Skerry-Ryan, R.; Stanton, D.; Wu, Y.; Weiss, R. J.; Jaitly, N.; Yang, Z.; Xiao, Y.; Chen, Z.; Bengio, S.; et al., 2017a. Tacotron: Towards end-to-end speech synthesis. In *Proc. Interspeech 2017*, 4006–4010. (cited on page 19)

Wang, Y.; Skerry-Ryan, R. J.; Stanton, D.; Wu, Y.; Weiss, R. J.; Jaitly, N.; Yang, Z.; Xiao, Y.; Chen, Z.; Bengio, S.; Le, Q. V.; Agiomyrgiannakis, Y.; Clark, R. A. J.; and Saurous, R. A., 2017b. Tacotron: Towards end-to-end speech synthesis. In *Interspeech*. (cited on pages 11 and 132)

Wang, Y.; Stanton, D.; Zhang, Y.; Ryan, R.-S.; Battenberg, E.; Shor, J.; Xiao, Y.; Jia, Y.; Ren, F.; and Saurous, R. A., 2018. Style tokens: Unsupervised style modeling, control and transfer in end-to-end speech synthesis. In *The International Conference on Machine Learning (ICML)*, 5180–5189. PMLR. (cited on page 84)

Wang, Y.; Zhan, H.; Liu, L.; Zeng, R.; Guo, H.; Zheng, J.; Zhang, Q.; Zhang, X.; Zhang, S.; and Wu, Z., 2025b. Maskgct: Zero-shot text-to-speech with masked generative codec transformer. *International Conference on Learning Representations (ICLR)*, (2025). (cited on pages 20, 50, 51, 53, 54, 55, and 56)

Wang, Y.; Zheng, J.; Zhang, J.; Zhang, X.; Liao, H.; and Wu, Z., 2025c. Metis: A foundation speech generation model with masked generative pre-training. *arXiv preprint arXiv:2502.03128*, (2025). (cited on pages 20, 50, 51, 53, 54, and 56)

Wang, Z.; Duan, Q.; Tai, Y.-W.; and Tang, C.-K., 2024c. C3llm: Conditional multimodal content generation using large language models. *arXiv preprint arXiv:2405.16136*, (2024). (cited on page 17)

Wang, Z.; Lu, C.; Wang, Y.; Bao, F.; Li, C.; Su, H.; and Zhu, J., 2023d. Prolificdreamer: High-fidelity and diverse text-to-3d generation with variational score distillation. *Advances in Neural Information Processing Systems*, 36 (2023), 8406–8441. (cited on page 117)

Williams, D.; Randall-Page, P.; and Miranda, E. R., 2014. Timbre morphing: near real-time hybrid synthesis in a musical installation. In *NIME*. (cited on pages 24, 96, and 134)

Wu, D.-Y.; Chen, Y.-H.; and Lee, H.-Y., 2020. Vqvc+: One-shot voice conversion by vector quantization and u-net architecture. *arXiv preprint arXiv:2006.04154*, (2020). (cited on pages 23 and 84)

Wu, D.-Y. and Lee, H.-y., 2020. One-shot voice conversion by vector quantization. In *International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 7734–7738. IEEE. (cited on pages 23 and 84)

Wu, Y.; Chen, K.; Zhang, T.; Hui, Y.; Berg-Kirkpatrick, T.; and Dubnov, S., 2023a. Large-scale contrastive language-audio pretraining with feature fusion and keyword-to-caption augmentation. In *International Conference on Acoustics, Speech, and Signal Processing (ICASSP) 2023*, 1–5. IEEE. (cited on pages 29, 33, 43, 55, 87, and 92)

Wu, Y.; Chen, K.; Zhang, T.; Hui, Y.; Berg-Kirkpatrick, T.; and Dubnov, S., 2023b. 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)*, 1–5. IEEE. (cited on pages 121 and 122)

Xinyuan, H. L.; Cai, Z.; Garg, A.; Duh, K.; García-Perera, L. P.; Khudanpur, S.; Andrews, N.; and Wiesner, M., 2025. Scalable controllable accented tts. *arXiv preprint arXiv:2508.07426*, (2025). (cited on page 20)

Xu, J.; Guo, Z.; Hu, H.; Chu, Y.; Wang, X.; He, J.; Wang, Y.; Shi, X.; He, T.; Zhu, X.; et al., 2025. Qwen3-omni technical report. *arXiv preprint arXiv:2509.17765*, (2025). (cited on page 17)

Xue, J.; Deng, Y.; Gao, Y.; and Li, Y., 2024. Auffusion: Leveraging the power of diffusion and large language models for text-to-audio generation. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, (2024). (cited on pages 18 and 48)

Yamagishi, J.; Veaux, C.; and MacDonald, K., 2019. Cstr vctk corpus: English multi-speaker corpus for cstr voice cloning toolkit (version 0.92). *The Rainbow Passage which the speakers read out can be found in the International Dialects of English Archive:(http://web. ku. edu/˜ idea/readings/rainbow. htm).*, (2019). (cited on pages 89 and 133)

Yang, D.; Liu, S.; Huang, R.; Lei, G.; Weng, C.; Meng, H.; and Yu, D., 2023a. Instructtts: Modelling expressive tts in discrete latent space with natural language style prompt. *arXiv preprint:2301.13662*, (2023). (cited on page 84)

Yang, D.; Tian, J.; Tan, X.; Huang, R.; Liu, S.; Guo, H.; Chang, X.; Shi, J.; Bian, J.; Zhao, Z.; et al., 2024a. Uniaudio: Towards universal audio generation with large language models. In *Forty-first International Conference on Machine Learning*. (cited on page 3)

Yang, D.; Yu, J.; Wang, H.; Wang, W.; Weng, C.; Zou, Y.; and Yu, D., 2023b. Diffsound: Discrete diffusion model for text-to-sound generation. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, (2023). (cited on pages 2, 3, 4, 16, 18, 35, 36, 37, 40, and 132)

Yang, D.; Yu, J.; Wang, H.; Wang, W.; Weng, C.; Zou, Y.; and Yu, D., 2023c. Diffsound: Discrete diffusion model for text-to-sound generation. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 31 (2023), 1720–1733. (cited on pages 26 and 84)

Yang, L.-C.; Chou, S.-Y.; and Yang, Y.-H., 2017. Midinet: A convolutional generative adversarial network for symbolic-domain music generation. *arXiv preprint arXiv:1703.10847*, (2017). (cited on page 20)

Yang, Y.; Li, H.; Li, T.; Cao, B.; Zhang, X.; Chen, L.; and Liu, Q., 2025. Melodia: Training-free music editing guided by attention probing in diffusion models. *arXiv preprint arXiv:2511.08252*, (2025). (cited on page 113)

Yang, Z.; Yu, Z.; Xu, Z.; Singh, J.; Zhang, J.; Campbell, D.; Tu, P.; and Hartley, R., 2024b. Impus: Image morphing with perceptually-uniform sampling using diffusion models. *International Conference on Learning Representations (ICLR)*, (2024). (cited on pages 99 and 100)

Yao, J.; Yang, Y.; Lei, Y.; Ning, Z.; Hu, Y.; Pan, Y.; Yin, J.; Zhou, H.; Lu, H.; and Xie, L., 2023. Promptvc: Flexible stylistic voice conversion in latent space driven by natural language prompts. *arXiv preprint arXiv:2309.09262*, (2023). (cited on pages 23, 84, 85, 89, and 133)

Yasuda, Y.; Wang, X.; and Yamagishd, J., 2021. end-to-end text-to-speech using latent duration based on vq-vae. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 5694–5698. IEEE. (cited on page 19)

Yu, L.; Blunsom, P.; Dyer, C.; Grefenstette, E.; and Kocisky, T., 2016a. The neural noisy channel. *arXiv preprint arXiv:1611.02554*, (2016). (cited on page 22)

Yu, L.; Buys, J.; and Blunsom, P., 2016b. Online segment to segment neural transduction. *arXiv preprint arXiv:1609.08194*, (2016). (cited on page 22)

Yu, Y.; Srivastava, A.; and Canales, S., 2021. Conditional lstm-gan for melody generation from lyrics. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, 17, 1 (2021), 1–20. (cited on page 20)

Yu, Z.; Yang, Z.; and Zhang, J., 2025. Dreamsteerer: Enhancing source image conditioned editability using personalized diffusion models. *Advances in Neural Information Processing Systems*, 37 (2025), 120699–120734. (cited on page 117)

Yuan, Y.; Liu, H.; Liu, X.; Kang, X.; Wu, P.; Plumbley, M. D.; and Wang, W., 2023. Text-driven foley sound generation with latent diffusion model. *arXiv preprint*, (2023). doi:10.48550/arXiv.2306.10359. (cited on pages 18, 36, and 42)

Zandie, R.; Mahoor, M. H.; Madsen, J.; and Emamian, E. S., 2021. Ryanspeech: A corpus for conversational text-to-speech synthesis. *ISCA*, (2021). doi:10.21437/Interspeech.2021-341. https://doi.org/10.21437/Interspeech.2021-341. (cited on page 75)

Zeghidour, N.; Luebs, A.; Omran, A.; Skoglund, J.; and Tagliasacchi, M., 2021. Soundstream: An end-to-end neural audio codec. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 30 (2021), 495–507. (cited on pages 18, 28, and 36)

Zen, H.; Dang, V.; Clark, R.; Zhang, Y.; Weiss, R. J.; Jia, Y.; Chen, Z.; and Wu, Y., 2019. Libritts: A corpus derived from librispeech for text-to-speech. *arXiv preprint arXiv:1904.02882*, (2019). (cited on pages 87 and 89)

Zen, H.; Tokuda, K.; and Black, A. W., 2009. Statistical parametric speech synthesis. *speech communication*, 51, 11 (2009), 1039–1064. (cited on page 19)

Zhang, J.-X.; Ling, Z.-H.; Liu, L.-J.; Jiang, Y.; and Dai, L.-R., 2019. Sequence-to-sequence acoustic modeling for voice conversion. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 27, 3 (2019), 631–644. (cited on pages 23 and 84)

Zhang, K.; Li, Y.; Zuo, W.; Zhang, L.; Van Gool, L.; and Timofte, R., 2021. Plug-and-play image restoration with deep denoiser prior. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44, 10 (2021), 6360–6376. (cited on page 114)

Zhang, R.; Isola, P.; Efros, A. A.; Shechtman, E.; and Wang, O., 2018a. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 586–595. (cited on page 33)

Zhang, R.; Isola, P.; Efros, A. A.; Shechtman, E.; and Wang, O., 2018b. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 586–595. (cited on page 121)

Zhang, X.; Gu, Y.; Chen, H.; Fang, Z.; Zou, L.; Xue, L.; and Wu, Z., 2023. Leveraging content-based features from multiple acoustic models for singing voice conversion. *arXiv preprint arXiv:2310.11160*, (2023). (cited on pages 23 and 84)

Zhang, X.; Liu, D.; Liu, H.; Zhang, Q.; Meng, H.; Perera, L. P. G.; Chng, E.; and Yao, L., 2024a. Speaking in wavelet domain: A simple and efficient approach to speed up speech diffusion model. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 159–171. (cited on pages 20 and 114)

Zhang, X.; Southwell, B. J.; Pan, S.; Niu, X.; Ahmed, B.; and Epps, J., 2026. Why your tokenizer fails in information fusion: A timing-aware pre-quantization fusion for video-enhanced audio tokenization. *arXiv preprint arXiv:2604.12145*, (2026). (cited on page 17)

Zhang, Y.; Guo, W.; Pan, C.; Zhu, Z.; Jin, T.; and Zhao, Z., 2025. Isdrama: Immersive spatial drama generation through multimodal prompting. In *Proceedings of the 33rd ACM International Conference on Multimedia*, 9618–9627. (cited on page 137)

Zhang, Y.; Ikemiya, Y.; Choi, W.; Murata, N.; Martínez-Ramírez, M. A.; Lin, L.; Xia, G.; Liao, W.-H.; Mitsufuji, Y.; and Dixon, S., 2024b. Instruct-musicgen: Unlocking text-to-music editing for music language models via instruction tuning. *arXiv preprint arXiv:2405.18386*, (2024). (cited on pages 22, 23, 113, and 114)

Zhang, Y.; Ikemiya, Y.; Xia, G.; Murata, N.; Martínez-Ramírez, M. A.; Liao, W.-H.; Mitsufuji, Y.; and Dixon, S., 2024c. Musicmagus: zero-shot text-to-music editing via diffusion models. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence*, 7805–7813. (cited on pages 23, 104, 107, 113, 114, 123, and 135)

Zhou, K.; Sisman, B.; Liu, R.; and Li, H., 2021. Seen and unseen emotional style transfer for voice conversion with a new emotional speech dataset. In *International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 920–924. IEEE. (cited on page 84)

Zhu, Y.; Wu, Y.; Olszewski, K.; Ren, J.; Tulyakov, S.; and Yan, Y., 2022. Discrete contrastive diffusion for cross-modal music and image generation. In *International Conference on Learning Representations (ICLR)*. (cited on pages 36, 37, 38, 39, and 87)

Zou, Y.; Liu, J.; and Jiang, W., 2021. Non-parallel and many-to-one musical timbre morphing using ddsp-autoencoder and spectral feature interpolation. In *International Conference on Culture-oriented Science & Technology*. (cited on pages 25 and 96)