DIFFMUSIC: A ZERO-SHOT DIFFUSION-BASED FRAMEWORK FOR SOLVING MUSIC INVERSE PROBLEMS

Jia-Wei Liao
d11922016@ntu.edu.tw

Pin-Chi Pan
r12942103@ntu.edu.tw

Sheng-Ping Yang
r12922163@ntu.edu.tw

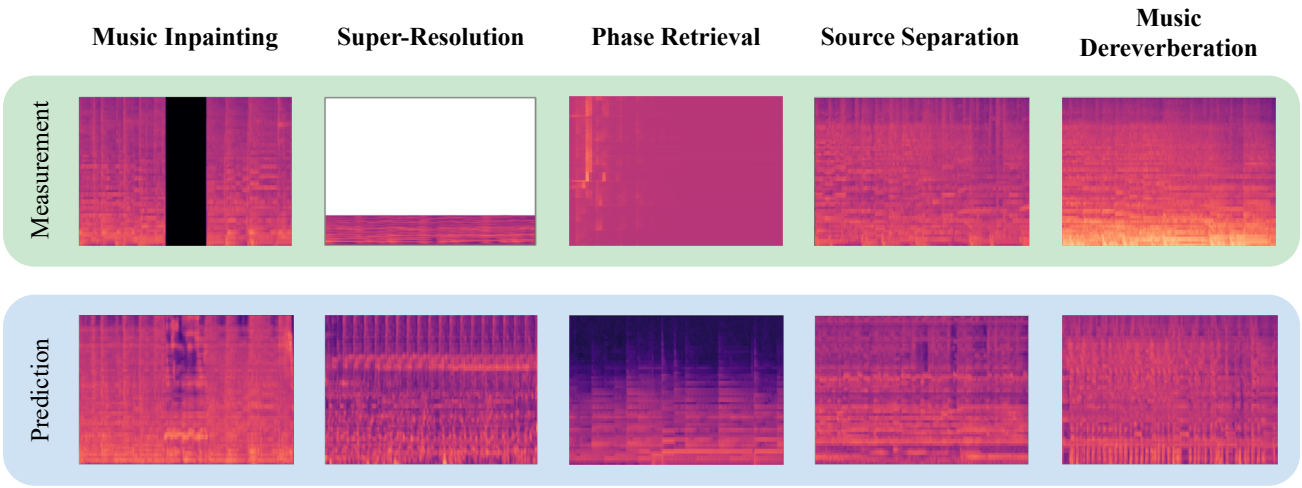

Figure 1: DiffMusic is a zero-shot framework designed to address a variety of music inverse problems, including music inpainting, super-resolution, phase retrieval, source separation, and music dereverberation.

ABSTRACT

The growing demand for high-quality music processing and diverse applications has revealed the limitations of traditional methods, which are often constrained by rigid architectures and require extensive task-specific training. To address these challenges, we introduce DiffMusic, a zero-shot **Diff**usion-based framework for solving **Music** inverse problems. Leveraging the generative power of pretrained Diffusion Models (DM), DiffMusic supports plug-and-play adaptability across various applications without additional training or fine-tuning. The framework incorporates Iterative Refinement Sampling (IRS) and a Vocoder Mel-spectrogram Constraint (VMC) to ensure high-quality music reconstruction. Through comprehensive experiments, DiffMusic demonstrates state-of-the-art performance, surpassing several task-specific benchmarks. Our code is available at https://github.com/jwliao1209/DiffMusic.

1 Introduction

The domain of music processing has experienced remarkable progress, propelled by the increasing demand for superior audio restoration and a wide array of applications. Traditional approaches to music inverse problems such as inpainting, super-resolution, and dereverberation have typically depended on specialized architectures and the extensive training of large datasets. Although these methods have achieved notable success, they often suffer from rigid structures and necessitate significant computational resources for fine-tuning and execution, thereby limiting their flexibility and applicability in practical settings.

The advent of diffusion models has revolutionized generative tasks, especially in image synthesis and restoration. More recently, these models have been adapted for music applications, demonstrating their potential in producing high-fidelity sound and facilitating complex editing tasks. Diffusion-based methodologies provide a probabilistic framework adept at modeling intricate distributions, making them particularly suitable for music processing tasks that demand precision and high quality.

Music inverse problems can be mathematically framed as optimization challenges, aiming to reconstruct the desired music signal from observed or incomplete data. This is typically expressed as:

$$\min_{\boldsymbol{x} \in \mathcal{M}} \|\mathcal{T}(\boldsymbol{x}) - \boldsymbol{y}\|_F^2 \tag{1}$$

In this equation, $\boldsymbol{x}$ denotes the target music signal confined within a feasible manifold $\mathcal{M}$, $\boldsymbol{y}$ represents the observed measurements, and $\mathcal{T}$ is the forward operator that maps the target signal to its measurements. This formulation is fundamental to various applications, including music inpainting, super-resolution, and phase retrieval. However, solving such optimization problems often involves computationally intensive algorithms, which are not ideal for real-time or resource-constrained environments.

In this study, we propose **DiffMusic**, a zero-shot **Diff**usion-based framework designed to tackle a broad

spectrum of **Music** inverse problems. Unlike traditional methods that require task-specific training or fine-tuning, DiffMusic leverages pretrained diffusion models—such as AudioLDM2 and MusicLDM—facilitating plug-and-play adaptability across various applications. Our framework incorporates an innovative Vocoder Mel-spectrogram Constraint (VMC) to maintain the integrity of the generated music and employs iterative refinement through sampling, ensuring high-quality outputs even with limited computational resources.

As depicted in Fig. 1, DiffMusic effectively addresses several critical challenges in music processing. It restores missing or corrupted segments of musical pieces, preserving their original style and continuity. Additionally, it enhances low-resolution music to achieve high-resolution quality, which is essential for professional music production. The framework also isolates or extracts specific music components, such as vocals or instruments, from composite signals. Furthermore, it reconstructs complete music signals by estimating phase information from spectral amplitude data. Lastly, DiffMusic eliminates reverberation effects to recover clean music signals, particularly in acoustically challenging environments. Our contributions are summarized as follows:

1. We propose DiffMusic, a novel zero-shot framework incorporating specifically designed operators to provide efficient and flexible solutions to music inverse problems, eliminating the need for additional training or fine-tuning.

2. We integrate advanced generative techniques with Iterative Refinement Sampling (IRS) and Vocoder Mel-spectrogram Constraint (VMC) to enhance the quality of generated music.

3. DiffMusic demonstrate the versatility of DiffMusic through comprehensive experiments across multiple tasks, achieving performance that not only rivals, but also surpasses several task-specific benchmarks, outperforming existing methods.

2 Related Work

2.1 Diffusion Models

Diffusion models [1, 2] have recently gained attention as powerful generative frameworks, outperforming GAN-based models [3, 4] in unconditional image generation tasks. Despite their success, these models are computationally intensive, particularly when generating high-resolution images. To mitigate this limitation, Rombach et al. [5] introduced the Latent Diffusion Model (LDM), which utilizes a trained Variational Autoencoder (VAE) [6] to encode high-resolution images into a compact latent space. This compression significantly improves computational efficiency during the diffusion process while maintaining high visual fidelity. Furthermore, to enable more flexible and fine-grained control in downstream applications, methods such as those proposed by Zhang et al. [7], Qin et al. [8], and Zavadski et al. [9] focus on efficient adaptation. These approaches allow users to fine-tune only specific output layers, reducing the computational cost of customization.

2.2 Music Inverse Problems

Diffusion models [10–12] have been increasingly applied to solve various inverse problems [13–17] in audio and music domains, demonstrating their versatility and effectiveness. For music inpainting, these models are used to reconstruct missing or corrupted audio segments, leveraging their generative capabilities to produce coherent and contextually accurate completions [18, 19]. In audio super-resolution, diffusion models enhance low-resolution audio signals by generating high-quality, high-resolution representations, achieving superior results compared to traditional interpolation-based methods [20]. Phase retrieval, a critical challenge in audio reconstruction, has also benefited from diffusion models, which estimate phase components more effectively by incorporating prior knowledge from the generative process [21]. Furthermore, for source separation, diffusion models have been utilized to disentangle and isolate individual sound sources from complex mixtures, offering improvements in separation quality [22, 23]. Lastly, in music dereverberation, diffusion models mitigate the effects of reverberation by generating dry, reverberation-free audio signals, thereby restoring clarity and fidelity. Collectively, these applications highlight the promise of diffusion models in addressing a wide range of inverse problems in music and audio processing, bridging the gap between theory and practical implementation.

3 Background of Diffusion Models

Diffusion models facilitate the generation of samples by progressively refining simple Gaussian noise into complex target distributions. This process begins with Gaussian noise and iteratively applies the score function $\nabla_{\boldsymbol{x}} \log p(\boldsymbol{x})$, associated with the target distribution $p(\boldsymbol{x})$. While directly calculating the true score function is computationally intractable due to its dependence on the derivative of $p(\boldsymbol{x})$, it can be approximated without explicitly modeling $p(\boldsymbol{x})$.

Song et al. [24] developed score-based models that utilize score matching as an effective strategy for learning the score function. Ho et al. [2] introduced the Denoising Diffusion Probabilistic Model (DDPM), which adopts a noise-perturbation approach for learning the score function. This method improves estimation in low-density regions and enhances mode coverage, making it suitable for generating highly complex data such as natural images.

The training of DDPM involves gradually corrupting clean data by adding Gaussian noise through a schedule of controlled variance over discrete timesteps, a process referred to as forward diffusion. A parameterized model, $\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t)$, is trained to predict the noise added at each step based on the noisy input $\boldsymbol{x}_t$ and the corresponding noise

level t. The sampling of trained DDPM starts with random noise and iteratively applies $\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t)$ to reconstruct the data through reverse diffusion, ultimately producing samples from the learned distribution.

In forward diffusion, the data is perturbed by combining the original signal and Gaussian noise as follows: $\boldsymbol{x}_t = \sqrt{\bar{\alpha}_t}\boldsymbol{x} + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}_t$, where $\bar{\alpha}_t$ defines the noise strength, $t \in [0, T]$, and $\boldsymbol{\epsilon}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. As t approaches T, the perturbed data $\boldsymbol{x}_t$ converges to pure Gaussian noise. The model $\boldsymbol{\epsilon}_\theta$ is optimized by minimizing the regression loss between the true noise and the predicted noise: $\mathbb{E}_{t,\boldsymbol{x},\boldsymbol{\epsilon}_t}[\|\boldsymbol{\epsilon}_\theta(\boldsymbol{x}_t, t) - \boldsymbol{\epsilon}_t\|_F^2]$.

For sampling, Song et al. [25] extended DDPM with the Denoising Diffusion Implicit Model (DDIM), which generalizes the sampling process as:

$$\boldsymbol{x}_{t-1} = \sqrt{\bar{\alpha}_{t-1}}\left(\frac{\boldsymbol{x}_t - \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}_\theta(\boldsymbol{x}_t, t)}{\sqrt{\bar{\alpha}_t}}\right) + \sqrt{1-\bar{\alpha}_{t-1}-\sigma_t^2}\boldsymbol{\epsilon}_\theta(\boldsymbol{x}_t, t) + \sigma_t\boldsymbol{\epsilon}_t. \tag{2}$$

The first term in this equation represents the estimation of the clean sample $\hat{\boldsymbol{x}}_0$ from the noisy input $\boldsymbol{x}_t$, derived using Tweedie's Formula [26]. This relationship establishes a connection between noise prediction and the score function, demonstrating that the denoising objective is mathematically equivalent to the score-matching objective.

4 Method

DiffMusic (Fig. 2) is designed to address a wide range of music inverse problems by integrating advanced generative modeling techniques with domain-specific constraints. We introduce the **Iterative Refinement Sampling (IRS)** method combined with the **Vocoder Mel-spectrogram Constraint (VMC)** to reconstruct high quality music, leveraging pretrained diffusion models. Detailed descriptions of IRS and VMC are provided in Sec. 4.1.

Furthermore, Sec.4.2, Sec.4.3, Sec.4.4, Sec.4.5, and Sec. 4.6 outline the application of our framework to various music inverse problems, showcasing the effectiveness of our tailored operators for each task.

4.1 Iterative Refinement Sampling

DiffMusic utilizes a pretrained diffusion model to iteratively refine an initial latent representation, aligning it with the desired measurement constraints. This process, depicted in Fig. 3, begins with the initialization of the latent space by sampling a latent variable, denoted from a standard normal distribution $\boldsymbol{z}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$.

Subsequently, for each timestep t from T to 1, the framework engages in an iterative refinement process. Initially, it predicts a clean latent estimate $\hat{\boldsymbol{z}}_{0|t}$, using the pretrained diffusion model $\boldsymbol{\epsilon}_\theta$ by Tweedie's Formula [26]:

$$\hat{\boldsymbol{z}}_{0|t} := \mathbb{E}[\boldsymbol{z}_0|\boldsymbol{z}_t] = \frac{1}{\sqrt{\bar{\alpha}_t}}\left(\boldsymbol{z}_t - \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}_\theta(\boldsymbol{z}_t, t)\right). \tag{3}$$

This estimate is then decoded into an mel-spectrogram representation through a sequence of transformations involving the VAE decoder $\mathcal{D}$, vocoder $\mathcal{V}$, and mel-spectrogram transformation ψ: $\hat{\boldsymbol{x}}_{0|t} = (\psi \circ \mathcal{V} \circ \mathcal{D})(\hat{\boldsymbol{z}}_{0|t})$.

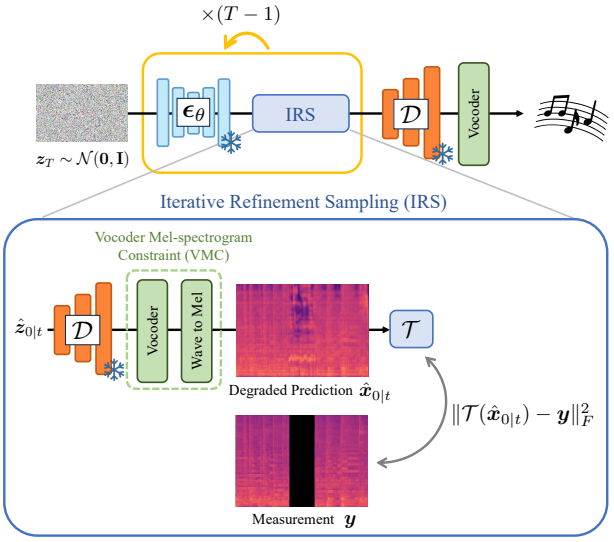

Figure 2: **DiffMusic Architecture.** It showcases the integration of pretrained diffusion models, the Vocoder Mel-spectrogram Constraint (VMC), and the iterative refinement process employed to reconstruct high-fidelity music from incomplete or degraded inputs. This visual representation underscores the framework's versatility in handling tasks such as music inpainting, super-resolution, and dereverberation.

To ensure that the reconstructed music adheres to the observed measurements $\mathbf{y}$, the latent estimate is adjusted by minimizing the discrepancy, measured by the Frobenius norm, with a hyperparameters γ:

$$\hat{\boldsymbol{z}}_{0|t}^* = \hat{\boldsymbol{z}}_{0|t} - \gamma\nabla_{\hat{\boldsymbol{z}}_{0|t}}\|\mathcal{T}(\hat{\boldsymbol{x}}_{0|t}) - \boldsymbol{y}\|_F^2. \tag{4}$$

Building on the insights from MPGD [15], the refined latent estimate $\hat{\boldsymbol{z}}_{0|t}^*$ resides on the tangent space of the music manifold $\mathcal{M}$, which closely approximates $\mathcal{M}$.

Following this adjustment, the noise component $\hat{\boldsymbol{\epsilon}}_t$ is recalculated to refine the latent variable for the subsequent iteration:

$$\hat{\boldsymbol{\epsilon}}_t = \frac{\boldsymbol{z}_t - \sqrt{\bar{\alpha}_t}\hat{\boldsymbol{z}}_{0|t}^*}{\sqrt{1-\bar{\alpha}_t}}. \tag{5}$$

The latent variable is then updated accordingly by DDIM sampling [25]:

$$\boldsymbol{z}_{t-1} = \sqrt{\bar{\alpha}_{t-1}}\hat{\boldsymbol{z}}_{0|t}^* + \sqrt{1-\bar{\alpha}_{t-1}}\hat{\boldsymbol{\epsilon}}_t. \tag{6}$$

Upon completing these iterations, the final latent variable, $\boldsymbol{z}_0$, is decoded to produce the music output $(\mathcal{V} \circ \mathcal{D})(\boldsymbol{z}_0)$.

This comprehensive approach enables DiffMusic to effectively address various music inverse problems by iteratively refining latent representations to meet specific measurement constraints. Detailed processing is provided in Algo. 1.

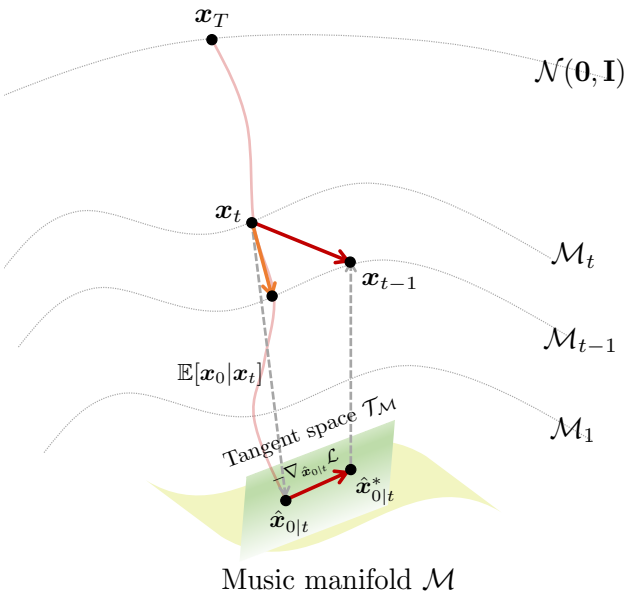

Figure 3: An illustration of Iterative Refinement Sampling.

Algorithm 1 Iterative Refinement Sampling of DiffMusic

Input: Measurement $\boldsymbol{y}$, inverse problem operator $\mathcal{T}$, UNet $\boldsymbol{\epsilon}_\theta(\cdot)$, VAE decoder $\mathcal{D}(\cdot)$, wave to mel-spectrogram transformation $\psi(\cdot)$, vocoder $\mathcal{V}(\cdot)$, sequence of noise schedule $\{\bar{\alpha}_t\}_{t=1}^T$, learning rate $\gamma > 0$.
$\boldsymbol{z}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$.
for $t = T$ to 1 **do**
  $\hat{\boldsymbol{z}}_{0|t} \leftarrow \frac{1}{\sqrt{\bar{\alpha}_t}} \left(\boldsymbol{z}_t - \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}_\theta(\boldsymbol{z}_t, t)\right)$.
  $\hat{\boldsymbol{x}}_{0|t} \leftarrow (\psi \circ \mathcal{V} \circ \mathcal{D})(\hat{\boldsymbol{z}}_{0|t})$.
  $\hat{\boldsymbol{z}}^*_{0|t} \leftarrow \hat{\boldsymbol{z}}_{0|t} - \gamma \nabla_{\hat{\boldsymbol{z}}_{0|t}} \|\mathcal{T}(\hat{\boldsymbol{x}}_{0|t}) - \boldsymbol{y}\|_F^2$.
  $\hat{\boldsymbol{\epsilon}}_t \leftarrow \frac{\boldsymbol{z}_t - \sqrt{\bar{\alpha}_t}\hat{\boldsymbol{z}}^*_{0|t}}{\sqrt{1-\bar{\alpha}_t}}$.
  $\boldsymbol{z}_{t-1} \leftarrow \sqrt{\bar{\alpha}_{t-1}}\hat{\boldsymbol{z}}^*_{0|t} + \sqrt{1-\bar{\alpha}_{t-1}}\hat{\boldsymbol{\epsilon}}_t$.
end for
return $(\mathcal{V} \circ \mathcal{D})(\boldsymbol{z}_0)$.

Applications in Music Inverse Problems. The architecture of our DiffMusic is designed to be versatile, enabling it to address a range of music inverse problems without the need for task-specific training. By iteratively refining the latent representation and enforcing measurement consistency through VMC, the framework can adapt to various challenges such as music inpainting, super-resolution, source separation, phase retrieval, and dereverberation. This adaptability is achieved through the integration of pretrained models and the application of domain-specific constraints, allowing DiffMusic to perform effectively across different tasks within the music processing domain. In summary, DiffMusic offers a unified and efficient approach to solving music inverse problems by combining pretrained diffusion models with innovative constraints like VMC. Its iterative refinement process, guided by measurement consistency and latent space adjustments, enables high-quality music reconstruction across a variety of applications.

4.2 Music Inpainting

In the field of music inpainting, the primary goal is to restore missing segments within an music signal, ensuring that the reconstructed portions are perceptually seamless and coherent with the existing content. This process is particularly relevant in scenarios where music recordings suffer from dropouts, deletions, or other forms of degradation.

Consider an original music signal represented as a time-frequency matrix $\boldsymbol{x}^* \in \mathbb{C}^{T\times F}$, where T denotes the time frames and F the frequency bins. A binary mask $\boldsymbol{M} \in \{0,1\}^{T\times F}$ is defined to identify the missing regions within the signal:

$$\boldsymbol{M}_{f,t} = \begin{cases} 0, & \text{if } t \in [t_{\text{start}}, t_{\text{end}}], \forall f, \\ 1, & \text{otherwise.} \end{cases} \tag{7}$$

Here, $[t_{\text{start}}, t_{\text{end}}]$ specifies the time interval of the missing segment. The degraded music signal $\boldsymbol{y}$ is then expressed as the element-wise product of the original signal and the mask: $\boldsymbol{y} = \boldsymbol{M} \odot \boldsymbol{x}^*$, where $\odot$ denotes the Hadamard (element-wise) product. To reconstruct the missing segments, an optimization problem is formulated to minimize the discrepancy between the observed (non-missing) parts of the original signal and the corresponding parts of the reconstructed signal. This discrepancy is often measured using the Frobenius norm, leading to the following loss function:

$$\mathcal{L}_{\text{inpaint}}(\boldsymbol{x}, \boldsymbol{y}) = \|\boldsymbol{M} \odot \boldsymbol{x} - \boldsymbol{y}\|_F^2, \tag{8}$$

In this context, $\hat{\boldsymbol{x}} = \text{argmin}_{\boldsymbol{x}\in\mathcal{M}} \mathcal{L}_{\text{inpaint}}(\boldsymbol{x}, \boldsymbol{y})$ represents the reconstructed signal, and $\|\cdot\|_F$ denotes the Frobenius norm, which computes the square root of the sum of the absolute squares of the matrix elements. The Hadamard product ensures that the loss is computed only over the observed (non-missing) entries, aligning the reconstruction with the available data.

By minimizing this loss function, the inpainting process iteratively adjusts the reconstructed signal $\hat{\boldsymbol{x}}$ to closely match the original signal in the observed regions, thereby promoting a seamless and coherent restoration of the missing music segments.

4.3 Super-Resolution

In the domain of music processing, super-resolution refers to the enhancement of an music signal's temporal or spectral resolution, aiming to reconstruct high-frequency components that are absent or diminished in the original recording. This technique is particularly valuable in applications such as restoring historical recordings, improving the quality of compressed music, and enhancing telecommunication signals.

Consider an original high-resolution music signal represented as a time-frequency matrix $\boldsymbol{x}^* \in \mathbb{C}^{T\times F}$, where T denotes the number of time frames and F the number of frequency bins. To simulate a low-resolution version of this signal, a downsampling operation $\boldsymbol{R}$ is applied, resulting in a degraded signal $\boldsymbol{y}$. This process can be mathematically expressed as: $\boldsymbol{y} = \boldsymbol{R}(\boldsymbol{x}^*)$. Here, $\boldsymbol{R}$ represents the

downsampling operator, which reduces the sampling rate of the original signal by a factor of r, known as the scaling factor. The downsampling can be further detailed by selecting every r-th sample from the original signal, effectively capturing only a subset of the time frames:

$$\boldsymbol{y}_t = \boldsymbol{x}_{rt}, \quad t = 0, 1, 2, \ldots, \left\lfloor \frac{T}{r} \right\rfloor, \tag{9}$$

In this equation, $\boldsymbol{y}_t$ denotes the t-th sample of the downsampled signal, and $\boldsymbol{x}_t$ represents the t-th sample of the original high-resolution signal. The operation effectively retains every r-th sample from the original signal, discarding the intermediate samples and thereby reducing the temporal resolution.

The primary objective of super-resolution is to reconstruct an estimate $\hat{\boldsymbol{x}}$ of the original high-resolution signal $\boldsymbol{x}$ from the downsampled signal $\boldsymbol{y}$. This involves predicting the missing intermediate samples and restoring the high-frequency content that was lost during the downsampling process. The reconstruction process can be approached by formulating an optimization problem that minimizes the discrepancy between the downsampled version of the reconstructed signal and the observed low-resolution signal. This discrepancy is often quantified using the Frobenius norm, leading to the following loss function:

$$\mathcal{L}_{\text{SR}}(\boldsymbol{x}, \boldsymbol{y}) = \|\boldsymbol{R}(\boldsymbol{x}) - \boldsymbol{y}\|_F^2. \tag{10}$$

By minimizing this loss function, the super-resolution algorithm iteratively adjusts the reconstructed signal $\hat{\boldsymbol{x}} = \operatorname{argmin}_{\boldsymbol{x} \in \mathcal{M}} \mathcal{L}_{\text{SR}}(\boldsymbol{x}, \boldsymbol{y})$ to ensure that its downsampled version closely matches the observed low-resolution signal $\boldsymbol{y}$, thus promoting an accurate restoration of the original high-resolution music content.

4.4 Phase Retrieval

In audio signal processing, phase retrieval is the task of reconstructing a time-domain signal from its magnitude spectrum, particularly when phase information is missing or unreliable. This challenge frequently arises in applications such as speech synthesis, audio restoration, and signal enhancement, where only the magnitude of the Short-Time Fourier Transform (STFT) is available.

Consider an original music signal represented as a time-domain vector $\boldsymbol{x}^*$. Applying the Short-Time Fourier Transform (STFT) to $\boldsymbol{x}^*$ yields a complex-valued matrix $\mathcal{F}(\boldsymbol{x})$, where $\mathcal{F}$ denotes the STFT operator. The magnitude spectrum $\boldsymbol{Y}$ is obtained by taking the absolute value of each element in $\mathcal{F}(\boldsymbol{x})$: $\boldsymbol{Y} = |\mathcal{F}(\boldsymbol{x})|$. Here, $\boldsymbol{Y}$ retains the amplitude information across time and frequency but discards the phase component. The objective of phase retrieval is to reconstruct an estimate $\hat{\boldsymbol{x}}$ of the original signal $\boldsymbol{x}^*$ using only the magnitude information $\boldsymbol{Y}$.

Reconstructing a signal solely from its magnitude spectrum is inherently ill-posed, as multiple signals can share the same magnitude spectrum but differ in phase, leading to different time-domain signals. This ambiguity necessitates the use of additional constraints or prior knowledge to achieve a meaningful reconstruction. To address this challenge, one common approach is to formulate an optimization problem that seeks to minimize the discrepancy between the magnitude of the Fourier transform of the estimated signal and the observed magnitude spectrum. This can be expressed through the following loss function:

$$\mathcal{L}_{\text{PR}}(\boldsymbol{x}, \boldsymbol{Y}) = \| \, |\mathcal{F}(\hat{\boldsymbol{x}})| - \boldsymbol{Y}\|_F^2, \tag{11}$$

This optimization process often involves iterative algorithms that alternate between the time and frequency domains, updating the estimate $\hat{\boldsymbol{x}}$ to reduce the loss function progressively. Such methods aim to recover a signal whose Fourier magnitude matches the observed data, while the phase is inferred to ensure a coherent time-domain reconstruction.

4.5 Source Separation

In audio signal processing, source separation involves decomposing a complex audio mixture into its constituent components, such as isolating individual instruments or voices from a combined recording. This technique is essential in applications like music remixing, speech enhancement, and audio analysis.

Consider a scenario where multiple music signals, denoted as $\boldsymbol{x}_i^*(t)$ for $i = 1, 2, \ldots, N$, are combined to form a mixed signal $\boldsymbol{y}(t)$. This mixing process can be mathematically represented as:

$$\boldsymbol{y}_t = \sum_{i=1}^{N} w_i \boldsymbol{x}_{i,t}^*, \quad t = 0, 1, \ldots, L-1, \tag{12}$$

where w_i represents the mixing weight for the i-th input signal, N is the total number of input signals, and L is the length of the signals. The weights w_i are typically non-negative and sum to one, ensuring a convex combination of the sources:

$$\sum_{i=1}^{N} w_i = 1 \quad \text{and} \quad w_i \geq 0. \tag{13}$$

The objective of source separation is to recover the original source signals $\boldsymbol{x}_{i,t}^*$ from the observed mixed signal $\boldsymbol{y}_t$. This problem is inherently ill-posed, as multiple sets of source signals can produce the same mixed output, especially when the number of sources exceeds the number of observations. To address this challenge, optimization techniques are employed to estimate the set of source signals $\hat{\boldsymbol{x}}_{i,t}$ that, when combined, best approximate the observed mixture. This estimation process can be formulated as minimizing the discrepancy between the observed mixed signal and the sum of the estimated source signals, often quantified using the Frobenius norm:

$$\mathcal{L}_{\text{seperate}}(\boldsymbol{x}_1, ..., \boldsymbol{x}_N, \boldsymbol{y}) = \left\| \sum_{i=1}^{N} w_i \boldsymbol{x}_i - \boldsymbol{y} \right\|_F^2. \tag{14}$$

By minimizing this loss function, the source separation algorithm iteratively adjusts the estimates $\hat{\boldsymbol{x}}_{i,t}$ to closely match the original source signals, thereby achieving an accurate decomposition of the mixed music signal.

Inverse Problem	Methods	AudioLDM2		MusicLDM	
		LSD ↓	FAD ↓	LSD ↓	FAD ↓
Music Inpainting	DPS [14]	**0.6207**	6.4334	**0.6318**	5.1730
	DSG [16]	0.7699	13.8478	0.7735	14.0801
	DiffMusic (Our)	0.6341	**4.9896**	0.6367	**4.3202**
Super-Resolution	DPS [14]	0.9815	9.2984	**0.9351**	7.9806
	DSG [16]	1.3427	15.0559	1.2783	17.1117
	DiffMusic (Our)	**0.9678**	**8.9296**	0.9778	**5.8756**
Phase Retrieval	DPS [14]	**0.8180**	7.7626	**0.7653**	6.7907
	DSG [16]	0.8258	14.6598	0.8873	16.4876
	DiffMusic (Our)	0.8323	**6.2551**	0.8939	**4.6492**
Source Separation	DPS [14]	0.9350	7.9542	0.9603	5.8537
	DSG [16]	0.8241	14.0942	0.9120	16.9260
	DiffMusic (Our)	**0.8293**	**6.2551**	**0.9334**	**4.9374**
Music Dereverberation	DPS [14]	0.6837	7.8759	0.7536	5.7646
	DSG [16]	0.7560	13.8926	0.8308	16.7926
	DiffMusic (Our)	**0.6604**	**7.1838**	**0.6788**	**4.8319**

Table 1: Quantitative results in music inverse problems. The best is marked in **bold**.

4.6 Music Dereverberation

In audio signal processing, music dereverberation aims to remove or reduce the reverberation effects from audio recordings, enhancing clarity and intelligibility. Reverberation occurs when sound reflects off surfaces in an environment, causing multiple delayed copies of the original signal to overlap. This can obscure details and reduce the quality of the recording.

A common model for reverberation involves convolving the original dry audio signal, denoted as $\boldsymbol{x}_t^*$, with a room impulse response (RIR) $\boldsymbol{h}_t$, resulting in the observed reverberant signal $\boldsymbol{y}_t$:

$$\boldsymbol{y}_t = \sum_{k=0}^{L-1} \boldsymbol{h}_{t-k} \boldsymbol{x}_k^*, \tag{15}$$

where L is the length of the impulse response. The RIR $\boldsymbol{h}_t$ characterizes how an environment modifies the original sound, typically including a direct path, early reflections, and late reverberations. To model the RIR, one approach is to use a recursive formulation incorporating a decay factor α and white Gaussian noise $\boldsymbol{\eta}_t$:

$$\boldsymbol{h}_t = \alpha \cdot \boldsymbol{h}_{t-1} + \boldsymbol{\eta}_t, \quad t = 0, 1, \ldots, L-1, \tag{16}$$

where $0 < \alpha < 1$ controls the decay rate of the reverberation. The goal of dereverberation is to estimate the original signal $\hat{\boldsymbol{x}}_t$ from the reverberant observation $\boldsymbol{y}_t$. This can be approached by formulating an optimization problem that minimizes the difference between the observed reverberant signal and the convolution of the estimated dry signal with the estimated impulse response. The loss function for this optimization is often expressed using the Frobenius norm:

$$\mathcal{L}_{\text{derev}}(\boldsymbol{x}, \boldsymbol{y}) = \|\boldsymbol{h} * \boldsymbol{x} - \boldsymbol{y}\|_F^2, \tag{17}$$

where $*$ denotes convolution operation. By minimizing this loss function, dereverberation algorithms iteratively adjust the estimate $\hat{\boldsymbol{x}}_t = \text{argmin}_{\boldsymbol{x} \in \mathcal{M}} \mathcal{L}_{\text{derev}}(\boldsymbol{x}, \boldsymbol{y})$ to closely approximate the original dry signal, effectively reducing the reverberation effects and enhancing the music quality.

5 Experiments

5.1 Experiment Settings

To evaluate the effectiveness of the proposed DiffMusic framework, we conducted experiments on the widely used Musdb18 [27] dataset, consisting of 100 songs. This dataset is comprehensive and serves as a benchmark for various music processing tasks. We tested DiffMusic across five distinct music inverse problems: music inpainting, super-resolution, phase retrieval, source separation, and music dereverberation.

For each task, we randomly clipped 5-second segments from the original audio sources. This ensured consistency across evaluations. The preprocessing involved converting audio to the appropriate format for Mel-spectrogram representation, compatible with the pretrained diffusion models used.

5.2 Implementation Details

We utilized AudioLDM2 [11] and MusicLDM [12] as the pretrained diffusion models within our framework which is built on the Diffusers library [28]. These models were chosen for their robust generalization in generative audio tasks. The evaluation metrics included Log Spectral Distance (LSD) [29] and Fréchet Audio Distance (FAD) [30], which quantify the perceptual and statistical similarities between reconstructed and ground-truth audio.

We applied consistent experimental configurations to ensure comparability across tasks and methods, including Diffusion Posterior Sampling (DPS) [14] and Diffusion Spherical Gaussian Constraint (DSG) [16] as baseline methods. We provide the detailed algorithms in Appendix A.1 and A.2. All experiments were conducted on hardware with computational resources sufficient to

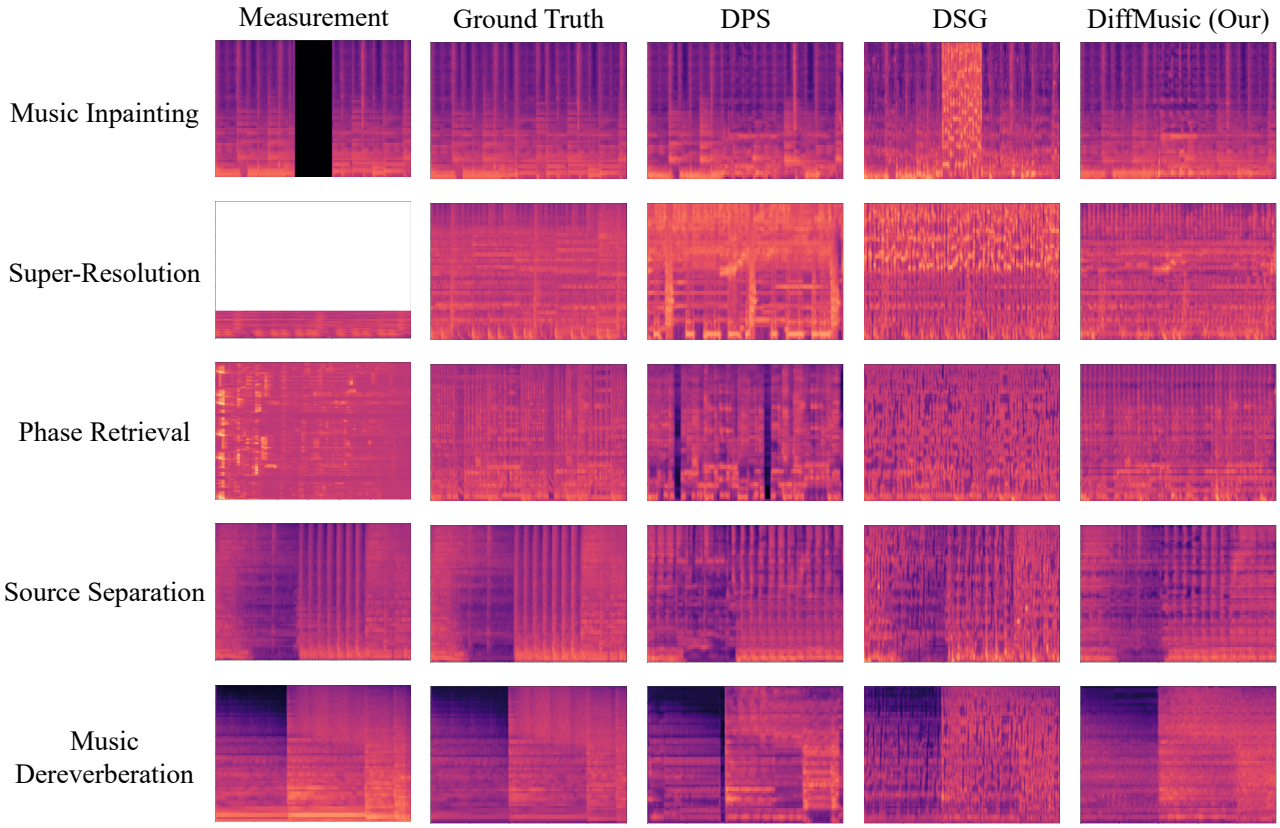

Figure 4: Qualitative results of music inverse problems.

support iterative diffusion processes within the time constraints emphasized in the proposed zero-shot paradigm.

5.3 Evaluation Metrics

5.3.1 *Log Spectral Distance (LSD)*

The Log Spectral Distance (LSD) [29] evaluates the perceptual similarity between reconstructed and ground-truth spectrograms by measuring the logarithmic deviation of their magnitude spectra. This metric is defined as:

$$\text{LSD} = \frac{1}{N}\sum_{n=1}^{N}\left(\frac{1}{K}\sum_{k=1}^{K}(\log|\boldsymbol{X}_{\text{rec}}(n,k)| - \log|\boldsymbol{X}_{\text{gt}}(n,k)|)^2\right)^{\frac{1}{2}}.$$

In this equation, N represents the total number of time frames in the spectrogram, while K denotes the number of frequency bins per frame. $X_{\text{rec}}(n,k)$ and $X_{\text{gt}}(n,k)$ orrespond to the magnitude values of the reconstructed and ground-truth spectrograms at time frame n and frequency bin k.

5.3.2 *Fréchet Audio Distance*

The Fréchet Audio Distance (FAD) [30] quantifies the similarity between the reconstructed and ground-truth audio distributions in a perceptually meaningful feature space. The feature representations are typically extracted using a pre-trained model designed for audio analysis. The FAD is computed as:

$$\text{FAD} = \|\boldsymbol{\mu}_{\text{rec}} - \boldsymbol{\mu}_{\text{gt}}\|_2^2 + \text{trac}(\boldsymbol{\Sigma}_{\text{rec}} + \boldsymbol{\Sigma}_{\text{gt}} - 2(\boldsymbol{\Sigma}_{\text{rec}}\boldsymbol{\Sigma}_{\text{gt}})^{\frac{1}{2}}).$$

In this formulation, $\boldsymbol{\mu}_{\text{rec}}$ and $\boldsymbol{\mu}_{\text{gt}}$ are the mean vectors of the feature distributions for the reconstructed and ground-truth audio. $\boldsymbol{\Sigma}_{\text{rec}}$ and $\boldsymbol{\Sigma}_{\text{gt}}$ are their covariance matrices. The first term measures the squared Euclidean distance between the mean vectors, while the second term accounts for the discrepancy in the covariance structures, incorporating both the individual covariances and their interaction.

Problems	Methods	VMC	AudioLDM2 LSD ↓	AudioLDM2 FAD ↓
Music Inpainting	DPS [14]		1.3881	7.6662
	DPS [14]	✓	**0.6207**	6.4334
	DiffMusic (Our)		1,1407	7.0905
	DiffMusic (Our)	✓	0.6341	**4.9896**
Phase Retrieval	DPS [14]		1.6784	22.9290
	DPS [14]	✓	**0.8180**	7.7626
	DiffMusic (Our)		1.1041	10.4053
	DiffMusic (Our)	✓	0.8323	**6.2551**
Music Dereverberation	DPS [14]		1.0613	14.9935
	DPS [14]	✓	0.6837	7.8759
	DiffMusic (Our)		1.0656	11.7008
	DiffMusic (Our)	✓	**0.6604**	**7.1838**

Table 2: Ablation study in music inverse problems. The best is marked in **bold**.

5.4 Quantitative Results

The performance of the proposed DiffMusic framework was evaluated quantitatively across five music inverse problems: music inpainting, super-resolution, phase retrieval, source separation, and music dereverberation. These evaluations utilized AudioLDM2 and MusicLDM as the pretrained models and considered two key metrics: Log Spectral Distance (LSD) and Fréchet Audio Distance (FAD). Across all tasks in Tab. 1, DiffMusic demonstrated superior performance compared to baseline methods, namely Dif-

fusion Posterior Sampling (DPS) and Diffusion Spherical Gaussian Constraint (DSG). For instance, in the music inpainting task with AudioLDM2, DiffMusic achieved an LSD of 0.6341 and a FAD of 4.9896. These results represent significant improvements over DPS, which obtained an LSD of 0.6207 and a FAD of 6.4334, and DSG, which recorded an LSD of 0.7699 and a FAD of 13.8478. Similarly, in the super-resolution task, DiffMusic also outperformed the baseline methods. When paired with AudioLDM2, DiffMusic achieved an LSD of 0.9678 and a FAD of 8.9296, compared to DSG, which recorded an LSD of 1.3427 and a FAD of 15.0559. The consistent reduction in FAD across all tasks underscores the capability of DiffMusic to generate perceptually and statistically accurate audio reconstructions. These findings highlight the versatility and robustness of DiffMusic in addressing a diverse set of music inverse problems.

5.5 Abation Study

The ablation study demonstrates the effectiveness of the proposed Vocoder Mel-spectrogram Constraint (VMC) strategy in enhancing the performance of both the DPS [14] method and our DiffMusic framework across three critical music inverse problems: music inpainting, phase retrieval, and music dereverberation. As shown in Tab. 2, the integration of VMC consistently improves both LSD and FAD across all tasks. These results highlight the versatility and effectiveness of VMC, which consistently enhances perceptual quality and reduces distortion in reconstructed audio.

5.6 Qualitative Analysis

The qualitative analysis involved inspecting Mel-spectrograms (Fig. 4) and listening to reconstructed music samples to assess perceptual quality. For tasks like music inpainting, DiffMusic exhibited superior results, seamlessly filling missing segments while maintaining tonal and stylistic consistency. In contrast, baseline methods often introduced noticeable artifacts or abrupt discontinuities.

In the inpainting task, DiffMusic demonstrated superior performance by restoring missing segments with remarkable continuity and coherence. The reconstructed music seamlessly blended with the original content, preserving both tonal quality and stylistic consistency. In contrast, baseline methods, such as DPS and DSG, frequently introduced abrupt transitions or unnatural artifacts, especially in cases with large missing regions, leading to a noticeable loss in continuity. For music dereverberation, DiffMusic outperformed baselines by significantly reducing reverberation while preserving the tonal balance of the original music. Baseline outputs often suffered from residual echoes or loss of detail, compromising the overall quality. These observations demonstrate that DiffMusic exhibits strong capability and robustness in addressing various inverse problems.

6 Conclusion

This paper introduces DiffMusic, a zero-shot diffusion-based framework designed to solve a wide range of music inverse problems. By leveraging pretrained diffusion models, DiffMusic eliminates the need for task-specific training or fine-tuning, offering a flexible and efficient solution across various applications. To enhance the quality of reconstructed music, we proposed the Vocoder Mel-spectrogram Constraint (VMC) and integrated it with an Iterative Refinement Sampling (IRS) algorithm, which incorporates task-specific operators to optimize music reconstruction. Comprehensive experiments demonstrated that DiffMusic achieves significant improvements in Log Spectral Distance (LSD) and Fréchet Audio Distance (FAD), surpassing existing methods.

7 References

[1] J. Sohl-Dickstein, E. Weiss, N. Maheswaranathan, and S. Ganguli, "Deep unsupervised learning using nonequilibrium thermodynamics," in *International Conference on Machine Learning (ICML)*, 2015.

[2] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[3] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2014.

[4] P. Dhariwal and A. Nichol, "Diffusion models beat gans on image synthesis," *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

[5] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-resolution image synthesis with latent diffusion models," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[6] D. P. Kingma and M. Welling, "Auto-encoding variational bayes." in *International Conference on Learning Representations (ICLR)*, 2014.

[7] L. Zhang, A. Rao, and M. Agrawala, "Adding conditional control to text-to-image diffusion models," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023.

[8] C. Qin, S. Zhang, N. Yu, Y. Feng, X. Yang, Y. Zhou, H. Wang, J. C. Niebles, C. Xiong, S. Savarese, S. Ermon, Y. Fu, and R. Xu, "Unicontrol: A unified diffusion model for controllable visual generation in the wild," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

[9] D. Zavadski, J.-F. Feiden, and C. Rother, "Controlnet-xs: Designing an efficient and effective architecture for controlling text-to-image diffusion models," *arXiv preprint arXiv:2312.06573*, 2023.

[10] H. Liu, Z. Chen, Y. Yuan, X. Mei, X. Liu, D. Mandic, W. Wang, and M. D. Plumbley, "Audioldm: Text-to-audio generation with latent diffusion models," in *International Conference on Machine Learning (ICML)*, 2023.

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

[12] K. Chen, Y. Wu, H. Liu, M. Nezhurina, T. Berg-Kirkpatrick, and S. Dubnov, "Musicldm: Enhancing novelty in text-to-music generation using beat-synchronous mixup strategies," in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2024.

[13] H. Chung, B. Sim, D. Ryu, and J. C. Ye, "Improving diffusion models for inverse problems using manifold constraints," *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

[14] H. Chung, J. Kim, M. T. Mccann, M. L. Klasky, and J. C. Ye, "Diffusion posterior sampling for general noisy inverse problems," in *The Eleventh International Conference on Learning Representations*, 2023.

[15] Y. He, N. Murata, C.-H. Lai, Y. Takida, T. Uesaka, D. Kim, W.-H. Liao, Y. Mitsufuji, J. Z. Kolter, R. Salakhutdinov, and S. Ermon, "Manifold preserving guided diffusion," in *International Conference on Learning Representations (ICLR)*, 2024.

[16] L. Yang, S. Ding, Y. Cai, J. Yu, J. Wang, and Y. Shi, "Guidance with spherical gaussian constraint for conditional diffusion," in *International Conference on Machine Learning (ICML)*, 2024.

[17] G. Daras, H. Chung, C.-H. Lai, Y. Mitsufuji, J. C. Ye, P. Milanfar, A. G. Dimakis, and M. Delbracio, "A survey on diffusion models for inverse problems," *arXiv preprint arXiv:2410.00083*, 2024.

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

[19] E. M. Juanpere and V. Välimäki, "Diffusion-based audio inpainting," *AES: Journal of the Audio Engineering Society*, 2024.

[20] H. Liu, K. Chen, Q. Tian, W. Wang, and M. D. Plumbley, "Audiosr: Versatile audio super-resolution at scale," in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2024.

[21] S. Shoushtari, J. Liu, and U. S. Kamilov, "Dolph: Diffusion models for phase retrieval," *arXiv preprint arXiv:2211.00529*, 2022.

[22] S. Lutati, E. Nachmani, and L. Wolf, "Separate and diffuse: Using a pretrained diffusion model for improving source separation," *arXiv preprint arXiv:2301.10752*, 2023.

[23] C.-Y. Yu, E. Postolache, E. Rodolà, and G. Fazekas, "Zero-shot duet singing voices separation with diffusion models," *arXiv preprint arXiv:2311.07345*, 2023.

[24] Y. Song, S. Garg, J. Shi, and S. Ermon, "Sliced score matching: A scalable approach to density and score estimation," in *Uncertainty in Artificial Intelligence*, 2020.

[25] J. Song, C. Meng, and S. Ermon, "Denoising diffusion implicit models," in *International Conference on Learning Representations (ICLR)*, 2021.

[26] B. Efron, "Tweedie's formula and selection bias," *Journal of the American Statistical Association*, 2011.

[27] Z. Rafii, A. Liutkus, F.-R. Stöter, S. I. Mimilakis, and R. Bittner, "The MUSDB18 corpus for music separation," 2017. [Online]. Available: https://doi.org/10.5281/zenodo.1117372

[28] P. von Platen, S. Patil, A. Lozhkov, P. Cuenca, N. Lambert, K. Rasul, M. Davaadorj, D. Nair, S. Paul, W. Berman, Y. Xu, S. Liu, and T. Wolf, "Diffusers: State-of-the-art diffusion models," 2022. [Online]. Available: https://github.com/huggingface/diffusers

[29] A. Gray and J. Markel, "Distance measures for speech processing," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 1976.

[30] K. Kilgour, M. Zuluaga, D. Roblek, and M. Sharifi, "Fr\'echet audio distance: A metric for evaluating music enhancement algorithms," *arXiv preprint arXiv:1812.08466*, 2018.

Appendix

A Algorithms for Solving Inverse Problem

A.1 Diffusion Posterior Sampling

Algorithm 2 DPS [14]

Input: Measurement $\boldsymbol{y}$, inverse problem operator $\mathcal{T}$, UNet $\boldsymbol{\epsilon}_\theta(\cdot)$, VAE decoder $\mathcal{D}(\cdot)$, sequence of noise schedule $\{\bar{\alpha}_t\}_{t=1}^T$, learning rate $\gamma > 0$.
$\boldsymbol{z}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$.
for $t = T$ to 1 **do**
  $\hat{\boldsymbol{z}}_{0|t} \leftarrow \frac{1}{\sqrt{\bar{\alpha}_t}}\left(\boldsymbol{z}_t - \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}_\theta(\boldsymbol{z}_t, t)\right)$.
  $\hat{\boldsymbol{z}}_{t-1} \leftarrow \sqrt{\bar{\alpha}_{t-1}}\hat{\boldsymbol{z}}_{0|t} + \sqrt{1-\bar{\alpha}_{t-1}}\boldsymbol{\epsilon}_\theta(\boldsymbol{z}_t, t)$.
  $\hat{\boldsymbol{x}}_{0|t} \leftarrow \mathcal{D}(\hat{\boldsymbol{z}}_{0|t})$.
  $\boldsymbol{z}_{t-1} \leftarrow \hat{\boldsymbol{z}}_{t-1} - \gamma\nabla_{\hat{\boldsymbol{z}}_t}\|\mathcal{T}(\hat{\boldsymbol{x}}_{0|t}) - \boldsymbol{y}\|_F^2$.
end for
return $(\mathcal{V} \circ \mathcal{D})(\boldsymbol{z}_0)$.

Algorithm 3 DPS [14] w/ VMC

Input: Measurement $\boldsymbol{y}$, inverse problem operator $\mathcal{T}$, UNet $\boldsymbol{\epsilon}_\theta(\cdot)$, VAE decoder $\mathcal{D}(\cdot)$, wave to mel-spectrogram transformation $\psi(\cdot)$, vocoder $\mathcal{V}(\cdot)$, sequence of noise schedule $\{\bar{\alpha}_t\}_{t=1}^T$, learning rate $\gamma > 0$.
$\boldsymbol{z}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$.
for $t = T$ to 1 **do**
  $\hat{\boldsymbol{z}}_{0|t} \leftarrow \frac{1}{\sqrt{\bar{\alpha}_t}}\left(\boldsymbol{z}_t - \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}_\theta(\boldsymbol{z}_t, t)\right)$.
  $\hat{\boldsymbol{z}}_{t-1} \leftarrow \sqrt{\bar{\alpha}_{t-1}}\hat{\boldsymbol{z}}_{0|t} + \sqrt{1-\bar{\alpha}_{t-1}}\boldsymbol{\epsilon}_\theta(\boldsymbol{z}_t, t)$.
  $\hat{\boldsymbol{x}}_{0|t} \leftarrow (\psi \circ \mathcal{V} \circ \mathcal{D})(\hat{\boldsymbol{z}}_{0|t})$.
  $\boldsymbol{z}_{t-1} \leftarrow \hat{\boldsymbol{z}}_{t-1} - \gamma\nabla_{\hat{\boldsymbol{z}}_t}\|\mathcal{T}(\hat{\boldsymbol{x}}_{0|t}) - \boldsymbol{y}\|_F^2$.
end for
return $(\mathcal{V} \circ \mathcal{D})(\boldsymbol{z}_0)$.

A.2 Diffusion Spherical Gaussian Constraint

Algorithm 4 DSG [16]

Input: Measurement $\boldsymbol{y}$, inverse problem operator $\mathcal{T}$, UNet $\boldsymbol{\epsilon}_\theta(\cdot)$, VAE decoder $\mathcal{D}(\cdot)$, sequence of noise schedule $\{\bar{\alpha}_t\}_{t=1}^T$, learning rate $\gamma > 0$.
$\boldsymbol{z}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$.
for $t = T$ to 1 **do**
  $\hat{\boldsymbol{z}}_{0|t} \leftarrow \frac{1}{\sqrt{\bar{\alpha}_t}}\left(\boldsymbol{z}_t - \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}_\theta(\boldsymbol{z}_t, t)\right)$.
  $\hat{\boldsymbol{\mu}}_{t-1} \leftarrow \sqrt{\bar{\alpha}_{t-1}}\hat{\boldsymbol{z}}_{0|t} + \sqrt{1-\bar{\alpha}_{t-1}-\sigma_t^2}\boldsymbol{\epsilon}_\theta(\boldsymbol{z}_t, t)$.
  $\boldsymbol{L} \leftarrow \|\mathcal{T}(\hat{\boldsymbol{x}}_{0|t}) - \boldsymbol{y}\|_F^2$.
  $r \leftarrow \eta\sqrt{\frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}D}$.
  $\boldsymbol{d} \leftarrow -r\frac{\nabla_{\hat{\boldsymbol{z}}_{t-1}}\boldsymbol{L}}{\|\nabla_{\hat{\boldsymbol{z}}_{t-1}}\boldsymbol{L}\|_F}$
  $\boldsymbol{d}_{\text{sample}} \leftarrow \boldsymbol{z}_t - \hat{\boldsymbol{\mu}}_{t-1}$
  $\boldsymbol{d}_{\text{mix}} \leftarrow \boldsymbol{d}_{\text{sample}} + w(\boldsymbol{d} - \boldsymbol{d}_{\text{sample}})$
  $\boldsymbol{z}_{t-1} \leftarrow \hat{\boldsymbol{\mu}}_{t-1} + \gamma\frac{\boldsymbol{d}_{\text{mix}}}{\|\boldsymbol{d}_{\text{mix}}\|_F}$
end for
return $(\mathcal{V} \circ \mathcal{D})(\boldsymbol{z}_0)$.

B Team Contribution

The following outlines the individual contributions of the team members to this project:

- Jia-Wei Liao
  1. Implemented the diffusion model pipeline using the diffuser library.
  2. Implemented the DSG algorithm.
  3. Proposed and implemented the DiffMusic Iterative Refinement Sampling (IRS) algorithm.
  4. Developed and refactored the codebase.
  5. Assisted in conducting experiments.
  6. Prepared the presentation sections on diffusion models overview.
  7. Contributed to writing the final report, including the Abstract, Methods, Experiments, and Conclusion sections.
- Pin-Chi Pan
  1. Defined and implemented inverse problem operations in music.
  2. Implemented the DPS algorithm.
  3. Proposed and implemented the Vocoder Mel-spectrogram Constraint (VMC) methods.
  4. Developed and maintained the codebase.
  5. Assisted in conducting experiments.
  6. Prepared the presentation sections on inverse problems.
  7. Contributed to writing the final report, including the Introduction, Methods, and Experiments sections.
- Sheng-Ping Yang
  1. Prepared experimental data.
  2. Created sample demo music files.
  3. Contributed to writing the Experiments section in the final report.