Audio Inpainting by Generative Decomposition

Zachary Shah[1], Neelesh Ramachandran, Mason L. Wang

Abstract

This report describes initial work exploring diffusion models to accomplish the task of inpainting in spectrogram image representations of audio. We build off of Riffusion [1], which is a text-to-spectrogram diffusion model fine-tuned from Stable Diffusion in which mel-spectrogram images can be generated from text descriptions of audio. We study two architectures to apply Riffusion to the task of audio inpainting: Riff-CNET, which applies a ControlNet [2] that adds an extra image condition to Riffusion; and Riff-P2P, a further fine-tune using the InstructPix2Pix approach which uses textual instructions to directly edit an input image [3]. Our goal is to generate sensible target instrument stems on top of the inputted background audio. We accomplish this by training these models to learn to identify individual instrument stems from text embeddings, which can be thought of as a form of musical decomposition learning in which the generation of a new instrument is dependent on learning the conditional distribution of that instrument given other classes of instruments. We demonstrate that the Riff-P2P approach is feasible for the sub-task of generating vocal melodies, while preserving the style, rhythm, harmony, and instrumentation of the background audio through forward sampling of a DDIM better than baseline approaches like diffusive seeding. Finally, we suggest methods to extend the Riff-P2P approach to the general audio inpainting task.

Index Terms

Stable Diffusion, ControlNet, Riffusion, audio inpainting, generative modeling, spectrogram processing

I. Introduction

IN generative music, audio inpainting is the task of adding additional musical stems or instruments into a piece of audio conditioned on a text prompt. For example, we may generate a singer's voice over a provided accompaniment or backing track containing other instruments. This task requires the model's generated stem to mesh harmoniously with provided incomplete audio across a number of dimensions, including style, tempo, rhythm, and harmony.

Currently, to generate audio from a text prompt, models like Riffusion have framed the problem instead as text-to-image generation, which applies the same architecture as Stable Diffusion to use text prompts and generate image representations of spectrograms (i.e., the magnitude of a complex spectrogram) [1]. To adapt Stable Diffusion for text-to-audio generation, the diffusive model samples an image of the spectral magnitude representation of a segment of audio that matches the conditions embedded into a given text prompt. The audio desired is then recovered through the inverse Fourier transform of the generated spectrogram image using the Griffin-Lim phase reconstruction algorithm.

This approach has been shown to sample audio from relatively vague text prompts; however, sampling with finer control is quite difficult due to size limitations of text prompts for standard text encoders. To provide finer control, we frame this task as a conditional image generation problem [4]; given a spectral representation of some background audio, we seek to generate a spectrogram image of the background audio superimposed onto a new vocal melody. This is a reduced version of the general audio inpainting problem, as we aim to only generate vocal melodies into a provided segment of background accompaniment audio. This spectrogram image should be sampled from the distribution of a dataset of songs from which the model is trained on, conditioned on both our given background audio and a text prompt. The generated audio can then be recovered in the same manner as before.

A. ControlNet (Riff-CNET)

Recently, L. Zhang and M. Agrawala at Stanford demonstrated supplying finer control over image generation through ControlNet, a neural network architecture that modifies pretrained generative networks to support adding extra conditions to them. Mainly, these conditions take the form of images detailing structural features like a Canny edge map or human pose, controlling structures in the network's output [2]. ControlNet has been shown to generate images conditional on these structural inputs with high precision, with or without an additional text prompt, as shown in Fig. 1.

The core idea behind ControlNet is the creation of two copies of the pre-trained network, as shown in Fig. 2. The first copy is locked, so that the parameters in the pre-trained network will not be modified. The second copy is not locked, and will be modified during the training process. The original conditions (text) are fed to the locked copy,

[1]Majority contribution

Fig. 1: Input and Output to using ControlNet on Stable Diffusion. Figure duplicated from [2].

and a new condition (spectral representations of background audio) are fed to the trainable copy. The two copies are connected at various stages through a "zero convolutional" layer, which is essentially a convolutional layer with a 1x1 kernel whose weights and biases are initialized to all zeros.

By adding ControlNet layers to the pre-trained Riffusion model, our goal is to generate output audio spectrograms (representing the music with the generated melody) conditioned not only on text prompts, but also on a reduced spectral representation of the background audio in the form of a Canny edge map. We see that for this new model, which we call Riff-CNET the generated audio will then follow the conditional distribution of full song segments on which it was trained.

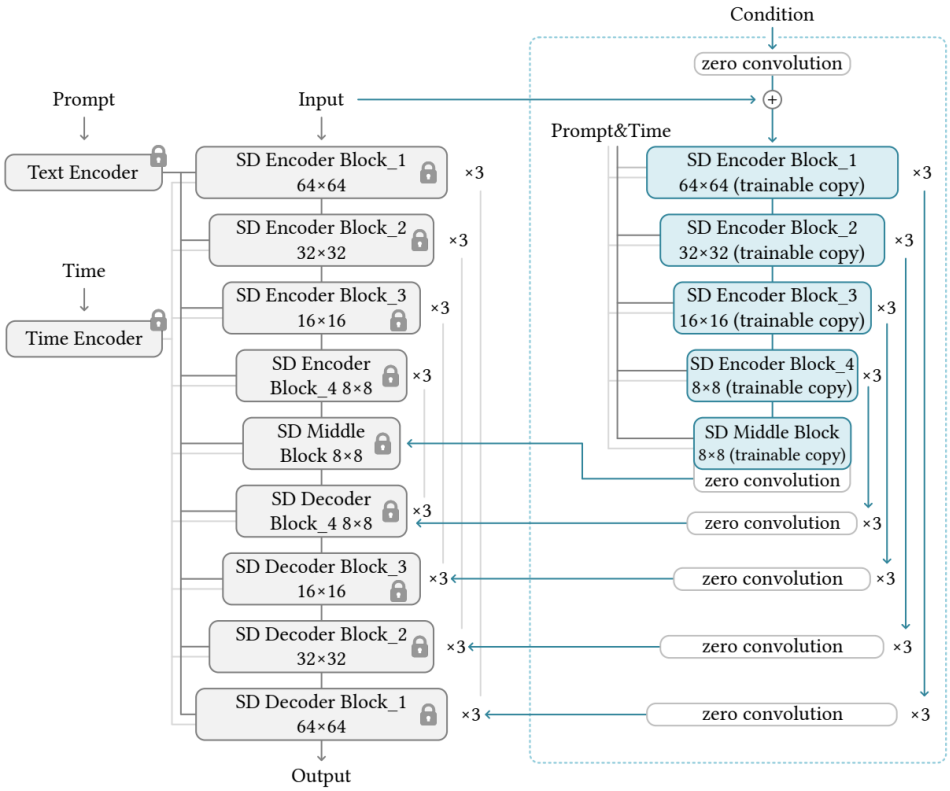


Fig. 2: Full ControlNet architecture. The gray blocks are the structure of Riffusion, while the blue blocks are ControlNet. Figure retrieved from [2].

Providing two copies of Riffusion into a ControlNet architecture, the locked copy will take in a text condition, while the trainable copy will receive in an image condition. Riffusion has already been trained to sample from thousands of

songs given text conditions, which we aim to preserve. Locking the first network allows us to preserve the full sampling knowledge already present within Riffusion, and prevents it from overfitting it to our training dataset. Since the zero-convolutional layers are initialized to kernels with zero weight and bias, as we train, the modifiable copy won't change the weights of the locked Riffusion model, preserving its learned styles and behavior. Thus, we aim only to train the modified copy and the zero-convolutional layers to learn embeddings for the conditioning spectral Canny edge maps. As the zero convolutional layers deviate further from their initialization, the trainable copy will begin to modify the values at different stages of the locked copy, to effectively add the new structural conditions to each sample.

B. InstructPix2Pix (Riff-P2P)

Though the ControlNet approach generally works well for preserving a designated shape of objects in the input, many of the finer patterns of a spectrogram are difficult to control with a reduced conditional input like an edgemap. One alternative to preserving input features is to reframe the task as image editing. In this case, given an image and an edit instruction, the goal is to learn an efficient shift of an image in the latent space towards its edited version, while preserving features not dictated in the edit instruction. This method requires a minimal architectural change to the diffusion model, which is to add new zero-initialized input channels for an input image where each text instruction is injected into the diffusion model (at the cross-attention layers of each block). These input embedding layers are learned during the fine-tuning training process, along with the rest of the model weights.

Figure 11. Applying our model recurrently with different instructions results in compounded edits.

Fig. 3: Pix2Pix Model applying edits (from original paper [3])

II. Experiments

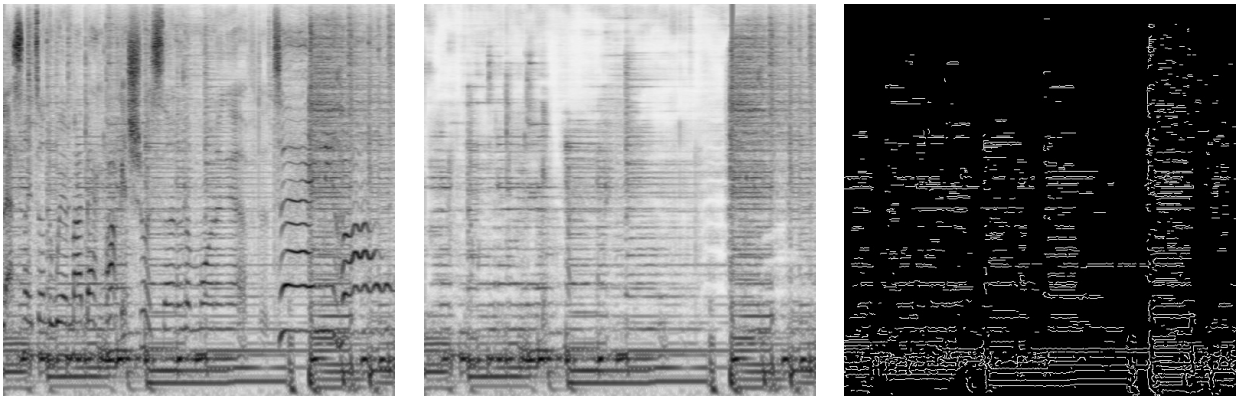
Below, several experiments are listed which detail attempts to train Riff-CNET and Riff-P2P models. We show potential for both approaches to learn the task of generating vocal melodies; however, Riff-CNET has thus far failed to learn the general audio inpainting task, while such an application for Riff-P2P has yet to be explored.

A. Riff-CNET for Vocal Melodies using Canny Edge Conditioning

1) Dataset: Our first experiment trained and evaluated Riff-CNET for vocal melody generation using GTZAN, a collection of 1000 30-second clips of songs from 10 different genres [5]. We manually selected 180 songs from GTZAN, which contained a diverse set of vocal stems, as our training set. This includes vocal songs from the Blues, Country, Reggae, Rock, and Pop genres. We then manually generated input text prompts for each of these 30-second clips, describing the characteristics of the vocal stem present.

2) Preprocessing: From our dataset of songs, we then prepared a set of three elements for each training example: two conditioning elements and a corresponding target for those conditions. The target is a 512×512 magnitude spectrogram representing a 5.12 second clip of full audio from our training dataset, and the conditions are a text prompt describing the vocal melody present in the full audio, as well as a 512×512 Canny edge map created from the spectrogram of the background extracted from the corresponding target. Given 180 audio-prompt pairs in our training dataset, prepared training examples as follows:

1) **Isolate the background audio** using Spleeter, a deep-learning based audio source separator. [6]. From this, we generated 180 30-second background-full audio pairs.
2) **Segment each of the** 30 **sec. clips** into 5.12 sec. segments, to generate a total of 1080 pairs of $\sim$ 5 sec. clips.
3) Augment the dataset by **shifting all clips into all** 12 **musical keys**, creating a total of 12960 clip pairs.
4) **Filter out segments associated with low vocal audio power**, determined by RMS ratio of vocals to background. A segment was filtered if the RMS power of the vocal stem was less than $0.1\times$ the RMS power of the background stem. This left us with 9869 pairs of 5 sec. clips with sufficient vocal power.



(a) Full Song Spectrogram (Target) **(b)** Bgnd Song Spectrogram **(c)** Bgnd Canny Edges (Source)

Fig. 4: Spectrograms for Data Processing. (a) A full audio spectrogram, which is our target image. (b) The spectrogram of the isolated background audio using Spleeter. (c) The Canny edges detected on the background audio spectrogram, which are the conditioning sources used in the ControlNet architecture.

5) **Generate magnitude spectrograms** for full and background audio segments. The set of full song spectrograms represents the distribution we are trying to sample from (9869 spec. pairs). See Figure 4(a),(b) for examples of these spectrogram representations.
6) **Perform Canny Edge detection** on background audio segments. These are input conditions to our network. See Figure 4(c) for an example.

This preprocessing is also summarized in Fig. 5. In total, we were able to generate about 9800 training examples.

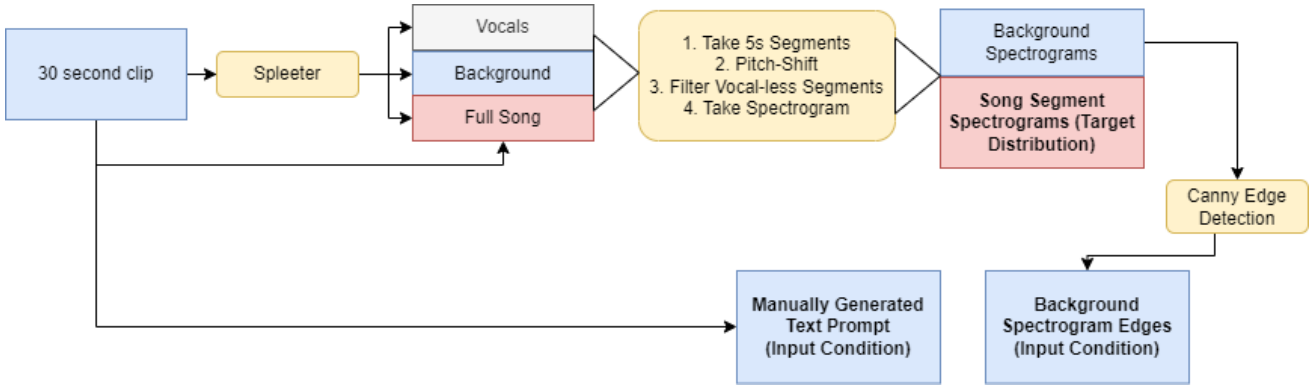


Fig. 5: Preprocessing pipeline for preparing Riff-CNET training examples.

3) Training: To prepare our model, we implemented a ControlNet architecture around the open-source Riffusion-v1 checkpoint retrieved from Huggingface, and trained exactly as decribed by [2]. Training was conducted for 12 hours on an NVIDIA Titan RTX for 8 epochs, with a batch size of 4 and total of 20,000 steps. Spectrograms were generated from a DDIM sampler with 50 denoising steps, as shown in Figure 6. The trained model checkpoint can is available for download on HuggingFace at [huggingface.co/zachary-shah/riffusion-cnet-v2].

4) Results: We provide a few example audio samples from Riff-CNET in the `README.md` in our Github Repo: [github.com/zachary-shah/riff-cnet]. In the section named "More Samples" in the `README.md`, we provide some examples of our target audio, along with the generated samples from Riff-CNET for each target. Preservation of our background audio is qualitatively evident from our examples. As compared to the existing methods of generating audio, we are notably more successful in generating audio that not only has a unique melody but also strongly preserves the input background audio.

Now, we also demonstrate that Riff-CNET quantitatively outperforms baselines in background preservation. To do so, we generated 20 samples from Riff-CNET and for each, extracted the background using Spleeter and compared it to the background used to generate the conditioning edge map. We repeated using three different metrics, each of which was computed pairwise between the background audio used to condition each sample and the isolated background of the sample. The metrics come from [7].

The first metric is L1 Spectral loss, which computes the average L1 distance over multiple scales of magnitude and log-magnitude spectrograms between audio clips. In eqs. (3) to (4), W is the background audio used to condition our

model, $\hat{W}$ is the background audio extracted from our generated audio, β is the window overlap ratio (we used 0.5), λ_1 and λ_2 are weights (we used 0.5), s_ω is the window size, and S is the short-time Fourier transform (STFT).

$$S_{\text{STFT}} = S\left(W, s_\omega, \beta s_\omega\right) \tag{1}$$

$$\hat{S}_{\text{STFT}} = S\left(\hat{W}, s_\omega, \beta s_\omega\right) \tag{2}$$

$$L_{\text{spec}}(W, \hat{W}) = \sum_{s_\omega = 128, \ldots, 2048} \lambda_1 \left| S_{\text{STFT}} - \hat{S}_{\text{STFT}} \right| \tag{3}$$

$$+ \lambda_2 \left| \log\left(S_{\text{STFT}}\right) - \log\left(\hat{S}_{\text{STFT}}\right) \right| \tag{4}$$

Second, we used CDPAM, a CNN trained to measure perceptual distance between audio clips, that is trained using contrastive methods and fine-tuned directly on human judgements[8]. Lastly, we measured the differences between the analytic audio envelopes of the target and sampled background audio. The formula for this is shown in eq. (5).

$$L_{\text{env}}(W, \hat{W}) = \sqrt{\frac{\left(|H(W)| - \left|H(\hat{W})\right|\right)^2}{N}} \tag{5}$$

Where the $|x|$ indicates the element-wise complex magnitude of x, H is the Hilbert transform, and N is the number of audio frames. Lower scores mean better preservation of the background audio for all three metrics.

Fig. 6: Diffusion process showing spectrogram sample generation from noise at denoising steps 0, 15, 30, 40, and 50. The output may be compared to the original audio features in fig. 4; several prominent features of the background are preserved, but the higher frequency content corresponding to vocals and melody are newly-formed.

As a baseline, we also compute these metrics for samples generated from Riffusion given the same Canny-edges of our background audio, used as seeds for the diffusive process (row 2 of Table I)). Riffusion fails to produce coherent music with this conditioning, so as another baseline, we seeded Riffusion instead with the corresponding full target spectrograms (row 3). In this case, Riffusion can generate music but does not preserve the original background audio well. These metrics are summarized in Table I, where we observe that Riff-CNET outperforms Riffusion for these two different conditioning settings in preserving background audio. This is especially significant since Riff-CNET is never given the background audio, just a Canny edge map, yet still preserves it.

Model	**Condition**	**Spectral**	**CDPAM**	**Envelope**
Riff-CNET	bkg. edges	**6.84**	**0.206**	**0.172**
Riffusion	bkg. edges	11.9	0.463	0.234
Riffusion	target Spec.	9.33	0.271	0.204

TABLE I: Metrics on Background Audio Preservation. Lower is better.

5) Discussion: The Riff-CNET model trained on Canny Edge maps is able to generate melodic components containing human-like lyrics which do not correspond to English words, but coherent lyrical generation is an open problem is audio generation. However, several problems occured in this training scheme which likely affected our results:

1) **Melody leakage from Spleeter**. Though spleeter is pre-trained to splice vocals from a song, it does not do its job perfectly. Some low-volume artifacts of vocals can be heard in certain background tracks (especially in rock and heavy metal examples), which may have biased the training process to learn and amplify existing melodies.
2) **Training example leakage**. After training, we found evidence of songs repeated in the train and test set, which may have led to unfair evaluation.
3) **Inconsistent background preservation**. Though melodies were often generated into similar backgrounds to the input, the style of these background instruments is often compressed. This is likely due the edge conditioning preserving only the major features of the background, but not subtle details like higher instrument harmonics or tamber encoded into low-power regions of the spectrogram.

B. Alternate Riff-CNET Training Experiments.

Several other experiments were performed to explore Riff-CNET's capabilities.

First, we examined alternative edge conditions. Beyond various Canny edge maps, we also explored Sobel Edge maps generated both from the true spectrogram and a non-local means denoised spectrogram. Sobel Edge maps have tendency to preserve much more information in the original image, as the kernel size of the edge detector is much smaller than that of a Canny edge detector; thus, we hypothesized Sobel edges would better preserve input audio detail. Variants of the Sobel Edge map are detailed in Figure 7.

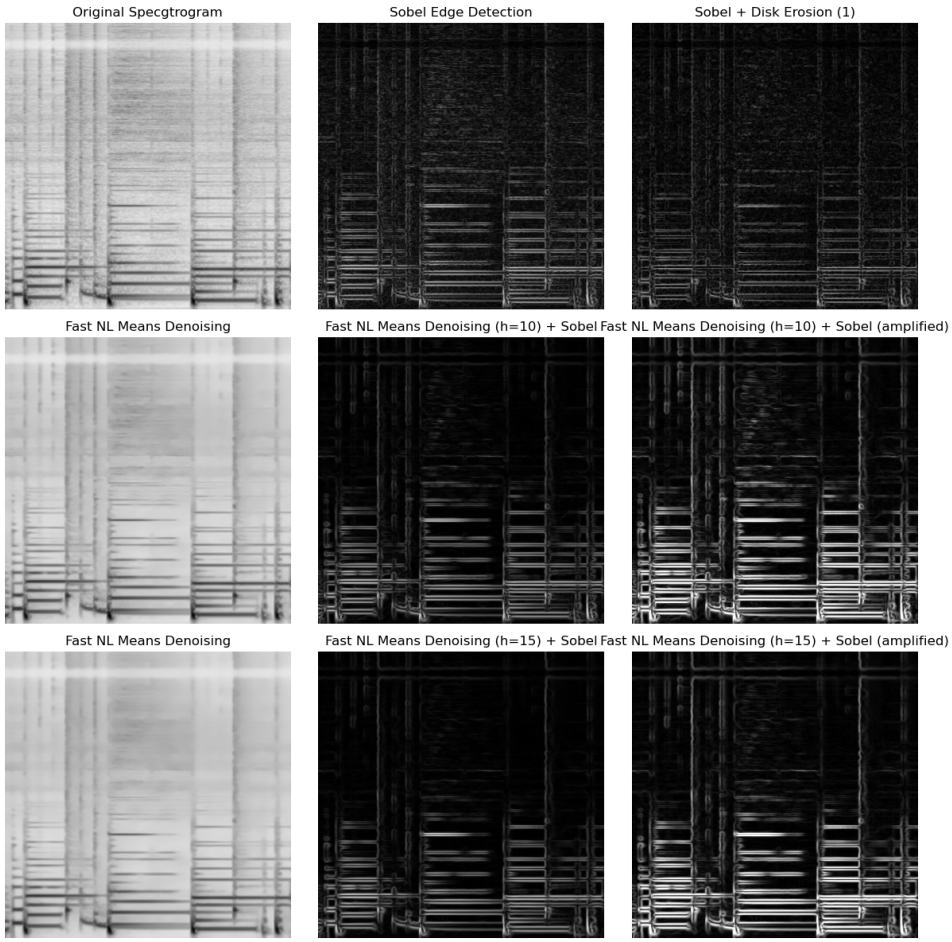


Fig. 7: Various edge condition maps in relation to an example spectrogram.

From our initial results, in which we repeated the training process for the original Riff-CNET above, we found no improvement in sample quality from these other various edgemaps. We even attempted to use the full original spectrogram as an input condition, but training on the full spectrogram did not result in any ability to generate sensible samples in the same time-frame.

Second, we attempted to train Riff-CNET for longer (3 days). This also did not seem to improve sample quality very much.

Finally, we looked into training on alternative datasets. Namely, we attempted to train on the general audio inpainting task using Slakh2100, a pre-stemmed music dataset with over 100 unique instruments represented. However, training such a task for 3 days did not converge for Riff-CNET under any edgemap configuration.

Details of these experiments can be found at the public notebook: [notion.site/AudioInpainting].

C. Riff-P2P for Vocal Melody Generation.

Given that ControlNet's intent is to preserve boundaries on objects, but not higher-frequency content, it may be that edge preservation is not the best method to enforce background audio preservation. This motivates the Riff-P2P model, where we trained an InstructPix2Pix model from the Riffusion v1 checkpoint.

To show proof-of-concept of this approach, we trained Riff-P2P to generate vocal melodies using the MUSDB-18 dataset, which contains 150 full-length pre-stemmed music tracks, in which the vocals and background are pre-separated in the dataset. We prepared training examples for the model by using the background audio spectrogram as an input, with manually written prompts to add in a vocal melody of a designated style. The target that the model tries to learn is the full audio spectrogram, which includes the vocal melody super-imposed onto the background. An example is shown in Figure 8.

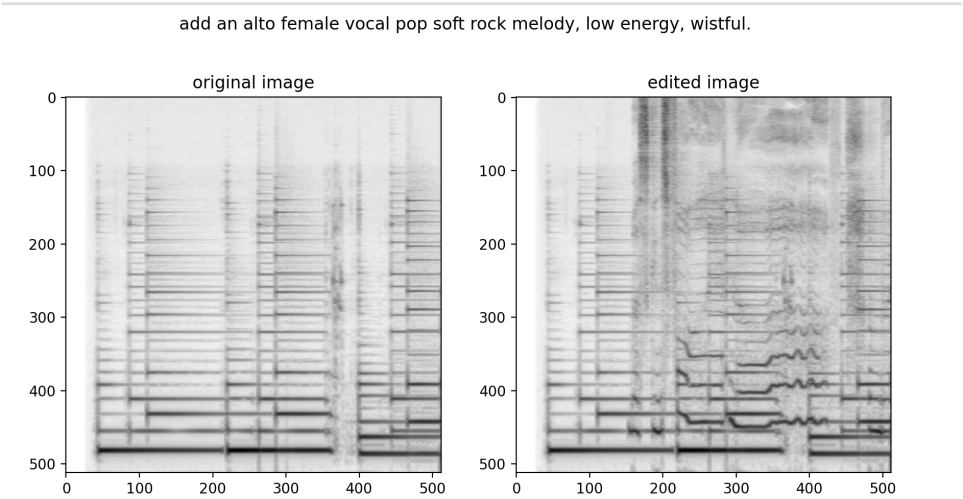


Fig. 8: Riff-P2P Training example. The original image (background audio spectrogram) is provided as the input image for training along with the edit instruction, with the edited image containing the full audio as the label / target

The training process is documented here: [notion.site/Riff-P2P-InstructPix2Pix-Training].

A few model samples along with all project code can also be found here:[github.com/zachary-shah/riff-pix2pix]

The pretrained model is located on huggingface: huggingface.co/zachary-shah/riff-pix2pix-v1

Overall, the results show good preservation of the background audio. However, melodies are not always generated based on the edit instruction. This is likely because the training dataset and training task are quite narrow; initial observations show that melodies are most likely to be generated when the input background has high similarity to some example song in the training dataset. Thus, the model does not generalize well to styles it is unfamiliar with, which is to be expected.

III. Recommendation for Future Work

We observe that with a relatively short training period, Riff-P2P vastly outperforms Riff-CNET in controlled melody generation. Further work can be done to apply this method to the general audio inpainting task.

First, training should be reproduced for Riff-P2P using the Slakh2100 dataset. This dataset has instruments pre-stemmed in more classes than MUSDB-18 and allows for preparation of training data in a combinatorial fashion. This was done for Riff-CNET in preprocess_slakh.py. This script should be re-written in the style of preprocessing for Riff-P2P according to training usage outlined by "Readme: How to Train". This will involve making an equivalent preprocessing script and dataset class for the Slakh2100 dataset, as was done for MUSDB18 in the riff-pix2pix repo. Edit instructions can be simply written in the form "add in {generated instrument}", which can be inferred and automatically written given the generated instrument selected during training example preparation.

To better assess the quantitative performance of Riff-P2P, the quantitative metrics in Table I shoudl be computed and compared to the performance observed for Riff-CNET. We expect Riff-P2P to perform much better quantitatively based on initial qualitative observations.

Finally, our current method is constrained to generating exactly 5.12 second audio clips, which is a constraint placed based on the input and output dimensions of each architecture. To sample longer audio clips, an interpolation between samples could be made in the latent space. Alternatively, sampling a set of sequential overlapping audio clips, with the overlapping region conditioned on the canny edges of the previously generated spectrogram, could produce a set of temporally coherent spectrograms which could be stitched together to form a longer sample. Initial work on this sample extension method is outlined for Riff-CNET here, which has yet to be applied to Riff-P2P.

Acknowledgment

The authors would like to thank Dr. Mert Pilanci and Rajarshi Saha for their guidance and assistance for this project, as well as for providing compute access on their NVIDIA Titan RTX GPU for model training.

References

[1] S. Forsgren and H. Martiros, *Riffusion - Stable diffusion for real-time music generation*, 2022. [Online]. Available: https://riffusion.com/about.

[2] L. Zhang and M. Agrawala, *Adding conditional control to text-to-image diffusion models*, 2023. arXiv: 2302.05543 [cs.CV].

[3] T. Brooks, A. Holynski, and A. A. Efros, "Instructpix2pix: Learning to follow image editing instructions," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 18 392–18 402.

[4] A. Van den Oord, N. Kalchbrenner, Espeholt, *et al.*, "Conditional image generation with pixelcnn decoders," *Advances in neural information processing systems*, vol. 29, 2016.

[5] A. Goyal and J. Bharadwaj, "Music genre classification,"

[6] R. Hennequin, A. Khlif, F. Voituret, and M. Moussallam, "Spleeter: A fast and efficient music source separation tool with pre-trained models," *Journal of Open Source Software*, vol. 5, no. 50, p. 2154, 2020.

[7] Anonymous, "Realimpact: A dataset of impact sound fields for real objects," in *Conference on Computer Vision and Pattern Recognition 2023*, 2023. [Online]. Available: https://openreview.net/forum?id=GZwFP1CcRO.

[8] P. Manocha, Z. Jin, R. Zhang, and A. Finkelstein, "Cdpam: Contrastive learning for perceptual audio similarity," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2021, pp. 196–200.