DIFF-A-RIFF: MUSICAL ACCOMPANIMENT CO-CREATION VIA LATENT DIFFUSION MODELS

Javier Nistal[1] **Marco Pasini**[2] **Cyran Aouameur**[1]
Maarten Grachten[1] **Stefan Lattner**[1]
[1]Sony Computer Science Laboratories, Paris, France
[2]Queen Mary University of London, UK
javier.nistal@sony.com

ABSTRACT

Recent advancements in deep generative models present new opportunities for music production but also pose challenges, such as high computational demands and limited audio quality. Moreover, current systems frequently rely solely on text input and typically focus on producing complete musical pieces, which is incompatible with existing workflows in music production. To address these issues, we introduce Diff-A-Riff, a Latent Diffusion Model designed to generate high-quality instrumental accompaniments adaptable to any musical context. This model offers control through either audio references, text prompts, or both, and produces 48kHz pseudo-stereo audio while significantly reducing inference time and memory usage. We demonstrate the model's capabilities through objective metrics and subjective listening tests, with extensive examples available on the accompanying website. [1]

1. INTRODUCTION

Deep generative modeling has recently made significant strides, greatly expanding the toolbox for synthesizing visual and auditory art [1–6] and signaling a new era of enhanced creative expression. These technologies promise more intuitive, high-level control over digital creations, yet their deployment in music production comes with inherent challenges. Generative music systems frequently rely solely on text inputs for control and typically focus on generating complete musical pieces rather than individual sounds or instruments. This approach can limit their integration into existing musical workflows and may compromise the artist's control over the final product. Furthermore, the computational demands of these advanced models often necessitate access to specialized hardware or online services. Additionally, they often fail to meet professional audio standards, such as true stereo output at 48 kHz.

In this paper, we introduce Diff-A-Riff, a novel Latent Diffusion Model designed for generating single-instrument accompaniments. A distinct feature of our approach is the ability to condition on musical audio contexts. This specific form of control crucially allows the music to dynamically adapt to the artist's style, enabling a more personalized creation process. Additionally, the model supports conditioning using joint text-and-audio embeddings from CLAP [7], which can be derived from either textual descriptions or audio references, providing versatile input options for directing the music generation.

At the core of Diff-A-Riff are two pivotal technological elements. First, the efficiency of a Consistency Autoencoder with a high compression rate enhances the system's performance in terms of inference time and memory usage [8]. Second, the model employs the expressiveness of Elucidated Diffusion Models (EDMs), known for their robust handling of complex data distributions and improved efficiency in model parameterization and inference [9].

We validate Diff-A-Riff through comprehensive evaluations, assessing its performance in ablation studies using objective metrics, and we compare it to other models and estimate context adherence using subjective listening tests. The results, detailed in Sections 5 and 6, demonstrate that our model not only achieves state-of-the-art audio quality (statistically not distinguishable from real audio) but also effectively adapts to various conditional settings confirming its potential for practical applications in music production.

The paper is organized as follows: after a review of related work in Section 2 and background in Section 3, we describe our methodology in Section 4. We then present our results in Section 5 and conclude with a discussion and potential future research directions in Section 6.

2. RELATED WORK

End-to-end models. The landscape of generative models for music has undergone transformative advancements in recent years. End-to-end Autoregressive Models (AMs) have traditionally been at the forefront of sound fidelity, diversity, and long-term coherence [10, 11]. Nonetheless, their high computational demands render AMs unsuitable for music production settings (i.e., sample rate $\geq$ 44.1 kHz, stereo). In contrast, Generative Adversarial Networks [12]

[1] sonycslparis.github.io/diffariff-companion/

and Variational Autoencoders [13] exhibit exceptional generation speed at high sampling rates [14–16], positioning them as valuable assets for commercial music production technologies [14, 17]. However, these strategies typically require simple datasets with reduced diversity [14, 15], and often restrict generation to fixed lengths [16, 17]. Recently, diffusion models showed a balanced equilibrium between generation quality, diversity, and efficiency [18–21]. Nevertheless, these rely on an iterative denoising process that, while faster than AMs, still demands long and heavy computations.

Latent models. To address the challenges inherent in end-to-end modeling, generative models have recently pivoted towards operating on compressed representation spaces learned via autoencoders [3–6, 22]. By doing so, generative systems can allocate representational capacity separately for learning immediate auditory characteristics of sound and longer-term music structure. Additionally, they facilitate the interpretation and integration of multi-modal control data, such as text [3, 5, 6], audio [23, 24], or melody [4]. Within this evolved framework, AMs leverage discrete representation spaces crafted through vector-quantized variational autoencoders [25, 26], resulting in faster models with better long-term structure [3, 4]. Recent developments have equipped AMs with parallel decoding using masked token modeling techniques [23, 27, 28], enabling sample rates as high as 44.1 kHz with acceptable inference speed.

Latent Diffusion Models (LDMs) also operate on compressed representation spaces, which are typically continuous [5, 6, 29]. This evolution has catalyzed the development of various LDMs capable of generating high-resolution musical audio with long-term structure [5, 6, 29, 30]. Notably, some works can generate audio at sampling rates as high as 48 kHz [6] and stereo [29, 30]. Other works like Stable Audio [5] improve inference efficiency, enabling the generation of 44.1 kHz sampling rate and stereo audio at an unprecedented speed.[2] Following this spirit, our system leverages a pre-trained Consistency Autoencoder [8] which enables Diff-A-Riff to function within a highly compressed representation space, allowing faster generation than previous systems. Further, our LDM employs the framework of Elucidated Diffusion Models (EDMs) [9, 20], a departure from the Denoising Diffusion Implicit Models (DDIMs) [31] used in previous approaches [5, 6, 29].

Control mechanisms. As evidenced by the state-of-the-art, text prompts currently serve as the most common interface for users to guide audio generative models [3–6]. To facilitate finer control, Jukedrummer [32] and Music ControlNet [33] utilize time-varying controls such as rhythmic and dynamic envelopes and melodic lines. By exploiting the semantic properties of multi-modal text-and-audio spaces, recent works propose zero-shot solutions to music editing via latent space manipulations [34] and inversion methods [35]. Alternative approaches to control pretrained models include inference-time optimization [36] or guidance [37]. Another method for influencing audio output involves conditioning on audio signals, a technique primarily used in style transfer and accompaniment tasks. In style transfer, the objective is to emulate specific aspects of the source audio, e.g., melody [4], timbre [24]. For accompaniment, the focus is on generating musical content that complements or enhances the conditioning audio [23, 24, 38–40]. Recent works attempting joint music generation and source separation also exhibit compositional capabilities such as accompaniment generation without requiring paired data [41, 42]. Inspired by these control mechanisms, our system introduces conditioning on audio and textual features derived from CLAP [7] alongside audio signals that serve as music context, widening the scope of generative capabilities, e.g., accompaniment generation, text-driven generation, and style transfer.

3. BACKGROUND

In this section, we provide a brief overview of Consistency Models and Denoising Diffusion Models. For an in-depth explanation, we encourage the reader to review the corresponding references.

Consistency Models (CMs) [43, 44] are a novel class of generative models that can produce high-quality samples in a single forward pass without adversarial training. CMs learn a mapping between noisy and clean data samples via a probability flow Ordinary Differential Equation (ODE) [31]. Given a noise level t, the consistency function f transforms a noisy sample $x_t \sim p_t(x)$ to a clean sample $x \sim p_{data}(x)$ by mapping $f(x_t, t) \mapsto x$. This consistency function is approximated by a neural network $f_\theta(x_t, t)$ with parameters θ. It must satisfy the boundary condition $f_\theta(x, t_{\text{min}}) = x$ and is trained by minimizing the discrepancy between its output and a teacher CM at adjacent noise levels t_i and t_{i+1}.

Denoising Diffusion Models (DDMs) [45] are generative models originally inspired by the concept of thermodynamic diffusion [46]. DDMs first add noise to data in a *forward* diffusion process and then use a neural network to *reverse* this process by removing the noise iteratively. The forward diffusion process is detailed by a Stochastic Differential Equation (SDE), introducing noise to the original data x_0 over T steps, resulting in a noisy version x_T. This process is defined by $dx_t = f(x_t, t)dt + g(t)dB_t$. Here, dB_t is the increment of a Wiener process (the random noise), $f(x_t, t)$ is the drift term, $g(t)$ is the diffusion term, and t represents the diffusion time step. The reverse process aims to reconstruct the original data from its noisy version by removing the noise. This is achieved by modeling the score of the data distribution, i.e., the gradient of the log probability density function of the noisy data with respect to the data itself, $\nabla_x \log p(x|t)$. The reverse process is described by another SDE, which guides the denoising $dx_t = [f(x_t, t) - g(t)^2 \nabla_x \log p(x_t|t)]dt + g(t)dB_t$, where a neural network g_θ, with parameters θ, is trained to estimate this score function, i.e., $g_\theta(x_t, t) \approx \nabla_x \log p(x_t|t)$. During inference, by performing this process iteratively, we can progressively transform pure noise inputs into data points following the training data distribution.

[2] 95 seconds of audio in 8 seconds on an A100 GPU

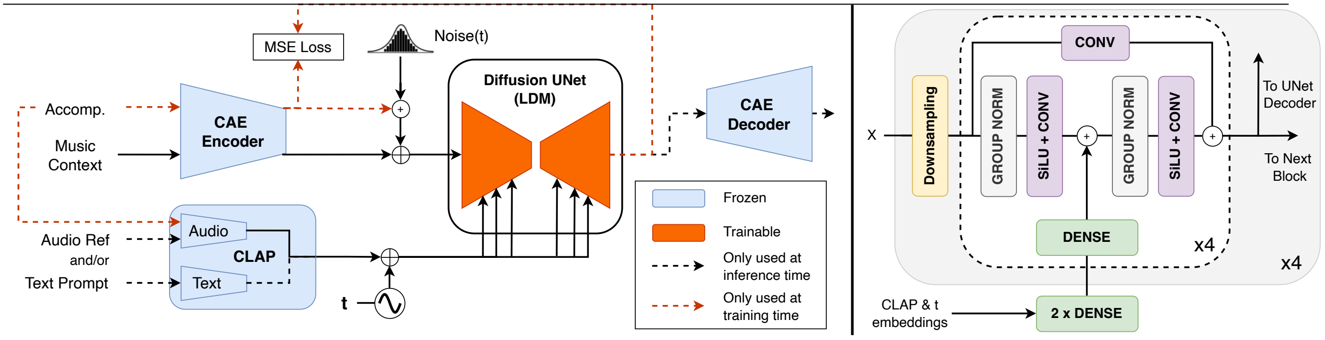

Figure 1. Overview of Diff-A-Riff. **Left**: The CAE Encoder transforms the music context into a compressed representation, concatenated with a noisy sample, and further processed through a multi-scale U-Net. At each scale, conditional CLAP and time-step embeddings are integrated through a feature-wise linear transformation. The generated latent sequence is decoded via the CAE Decoder. We highlight frozen components in blue and trainable elements in orange. Text prompting is only used at inference. **Right**: The encoder architecture comprises four down-sampling blocks with four convolutional and group norm layers with skip connections. The decoder mirrors this architecture.

4. METHODOLOGY

4.1 Dataset

We train our model on a proprietary dataset comprising 12,000 multi-track recordings of diverse music genres (e.g., pop/rock, R&B, rap, country). Each multi-track has various instrument tracks, including bass, guitars, pianos, vocals, and more. We resample each track to 48 kHz, convert it to mono and segment it into overlapping windows of approximately 10 seconds with a 3-second hop size. In training, we randomly select a target *accompaniment* track (excluding vocals) and construct the music *context* by mixing a random subset of the remaining tracks for each segment. We apply this data segmentation and sampling strategy offline to obtain 1M training pairs of audio segments. Following the same methodology, we derive a validation set from 1,200 multi-tracks.

4.2 Diff-A-Riff

4.2.1 Consistency Autoencoder

In this work, we employ a consistency model-based Autoencoder (CAE). We use it pre-trained and freeze its parameters to train a generative model on its latent embeddings (see Fig. 1). The CAE encodes audio samples into a continuous representation space with a 64× compression ratio. It operates on complex Short-Time Fourier Transform (STFT) spectrograms, with real and imaginary components as separate channels. The architecture uses convolutional residual blocks interleaved with down/up-sampling layers. The *CAE Encoder* produces 64-dimensional encodings in the range $(-1, 1)$ with a sample rate of 12 Hz for 48 kHz input audio. The model has $\sim$ 58 million parameters and is trained following the consistency training framework [44]. For a detailed description of the architecture and training procedure, we refer the reader to the original reference [8].

4.2.2 Latent Diffusion Model

We train a Latent Diffusion Model (LDM) on the latent space learned by the CAE. The proposed LDM follows the framework of Elucidated Diffusion Models (EDMs) [9], a departure from DDIMs [31] for improved model parametrization and inference. The architecture follows DDPM++ [47], an upgraded version of the originally proposed Diffusion Probabilistic Model [45]. We only adapt the network's input dimensionality to that of the CAE's latent space (64 channels, see Sec. 4.2.1). Also, we increase the dimensionality of the conditional embedding input with that of CLAP [7] (i.e., 512 dimensions). Our UNet is composed of four down/up-sampling blocks with convolutional layers and skip connections, both for the encoder and the decoder (see Fig. 1 Right). Self-attention is employed in the penultimate resolution layer. We use 512 base channels and double their number at each resolution block. Additionally, the model relies on two dense layers to project the concatenation of CLAP embeddings and the sinusoidal denoising step embeddings into a joint representation. The resulting embedding is used in all down/up-sampling blocks to condition the denoising process as illustrated in Fig 1.

4.2.3 Training

Fig. 1 provides a high-level overview of Diff-A-Riff's setup. Given a pair of input *context* and target *accompaniment* audio segments, the model is trained to reconstruct the *accompaniment* given the *context* and a CLAP embedding derived from a randomly selected sub-segment of the target itself. This prevents the model from relying on CLAP for temporal alignment. In order to use Classifier-Free Guidance (see Sec. 4.3.1) and allow the model to optionally operate unconditionally, we drop the audio context and clap embeddings both with a 50% probability. We train Diff-A-Riff over 1M iterations using a batch size of 256 (2 weeks on a single RTX 3090 GPU). We use AdamW [48] as the optimizer and a base learning rate of 10^{-4}. We use a learning rate schedule with an initial warm-up phase and a reduce-in-plateau process that decreases the learning rate to a minimum value of 10^{-6}. We keep an Exponential Moving Average (EMA) on the weights with a momentum of 0.9999. The resulting model has 500M parameters (including the CAE and not CLAP) and occupies 3 GB of memory.

4.3 Evaluation

Objective comparison of Diff-A-Riff with existing state-of-the-art models [4–6] is challenging as these are generally trained to perform a substantially different task (generation of fully mixed music with no accompaniment conditioning). Even though StemGen [23] and SingSong [39] are trained to generate accompaniments, their implementation and pretrained weights are not publicly available. Therefore, we focus on subjective listening tests to compare with available music generation models. Additionally, we perform objective evaluations to analyze different inference parameters (e.g., number of diffusion steps, conditioning information; see Section 4.3.1) to understand which configurations of our proposed model perform the best on our set of metrics. We then apply the gained insights in order to generate the samples that are proposed in the user studies. In the following sections, we describe how we generate the samples used for evaluation, the objective metrics, the listening test methodology, and the baselines we compare against.

4.3.1 Inference Configurations

In this section, we describe the inference configurations that we use for the objective and subjective evaluations.

Conditioning Signals: Different conditioning signals are evaluated. $\textit{CLAP}_A$ refers to the audio-derived CLAP embeddings, which are obtained by using CLAP to encode real audio from a track of the evaluation set. We can also condition the model on text-derived CLAP embeddings, despite them never being fed during training to the model, since CLAP offers a joint embedding space for both modalities. Because our dataset does not contain audio/text pairs, we create $\textit{CLAP}_T$ embeddings by asking ChatGPT to write text descriptions of single-stem tracks. Finally, *Context* refers to the conditioning signal obtained by solely encoding the music context into the CAE Encoder.

Classifier-Free Guidance (CFG) [49] allows to improve generation quality by increasing the influence of conditioning signals in the sampling process. Given the guidance strength CFG, we implement guidance as $x_{t-1} = f_\theta(x_t) + \text{CFG} \cdot (f_\theta(x_t, c) - f_\theta(x_t))$. At inference time, we can use different guidance strengths for *Context* and *CLAP* embeddings, denoted as $\text{CFG}_{\textit{Context}}$ and $\text{CFG}_{\textit{CLAP}}$, respectively.

Number of Diffusion Steps: At inference time, the number of denoising steps T allows to trade between audio quality and generation speed.

Pseudo-Stereo Generation: We generate pseudo-stereo audio by denoising until a given diffusion time step, and then by independently concluding a stochastic denoising process twice, one for each audio channel. We define the *stereo width* as the proportion of denoising steps used for stereo generation over the total number of steps. In the user study, we set this parameter to 0.4.

4.3.2 Objective Metrics

We evaluate Diff-A-Riff through objective metrics to assess various aspects of the generated audio. These include the standard *Squared Maximum Mean Discrepancy* (MMD2) [50] and *Fréchet Audio Distance* (FAD) [51] for audio quality as well as *Density* and *Coverage* [52] for evaluating fidelity and diversity. To study the system's responsiveness to text prompts, we employ the *Clap Score* (CS) [53], which calculates the cosine similarity between text and audio embeddings. In order to evaluate the alignment of the generated accompaniment with the context, we employ the *Audio Prompt Adherence* (APA) [54], a metric based on FAD tailored to evaluate accompaniment systems. All metrics are calculated by averaging five batches of 500 candidate samples. We use CLAP [7] as the embedding space for metrics that compare distributions (like MMD2 and FAD) using a reference set of 5,000 real audio examples.

4.3.3 Listening Tests

Subjective Audio Quality (SAQ): We perform a Mean Opinion Score (MOS) test to assess audio quality. Participants were presented with 5-second audio segments from real data as well as generations from the baselines and the proposed system. Their task is to rate the audio quality of these segments on a 5-level Likert scale ranging from poor (1) to excellent quality (5). For all items (real data, Diff-A-Riff, and baselines), we compare both complete music pieces as well as solo instruments.

We generate solo instruments with Diff-A-Riff by conditioning the model on text or audio-derived CLAP embeddings ($\textit{CLAP}_{A \text{ or } T}$) and without an input context (for a fair comparison with the baselines, which do not rely on contextual audio inputs). Despite the model not being trained for this task, we can also generate *complete* music pieces using *CLAP* and *Context* embeddings, following an iterative approach: First, we create sets of $\textit{CLAP}_{A \text{ or } T}$ embeddings as described in 4.3.1. Then, from an initially empty context, we generate new tracks from those *CLAP* embeddings, iteratively summing the resulting generation into the input context from which we derive the next *Context* embeddings.

We compare Diff-A-Riff against three state-of-the-art text-to-music baselines: AudioLDM2 [6], MusicGen [4], and Stable Audio [5]. [3] For each baseline, we generate 20 5-second excerpts of complete music and solo instruments using text prompts generated by ChatGPT.

Subjective Audio Prompt Adherence (SAPA): We also conduct a subjective assessment of audio-prompt adherence. Participants are provided with a reference 10-second music segment and are asked to rate the compatibility of five distinct accompaniments on a scale from 0 (indicating no adherence) to 100 (perfect adherence), according to harmonic, rhythmic, and overall music style compatibility.

[3] We use the open-source AudioLDM2 model 'AudioLDM2-48kHz' operating at 48 kHz. For MusicGen, we use the open-source model 'MusicGen-large' operating at 32 kHz, and for Stable Audio, we use their public API.

	Cond. Signal	↓ MMD2^a	↓ FAD	↑ *Coverage*	↑ *Density*	↑ APA	↑ CS
Real	Original acc.	0.00	0.02	0.18	1.03	0.93	1.00
Lower bound	-	64.32^b	1.67^b	0.00^b	0.00^b	0.11^c	-0.07^c
Diff-A-Riff $^{\text{CFG}=1.25}_{T=30}$	$CLAP_A$ + *Context*	**0.22**	**0.03**	**0.17**	**1.00**	**0.92**	-
	$CLAP_T$ + *Context*	3.96	0.17	0.05	0.32	0.20	**0.25**
	Context only	4.87	0.24	0.05	0.37	0.23	-
	$CLAP_A$ only	0.36	0.03	0.14	0.84	-	-
	$CLAP_T$ only	4.38	0.20	0.05	0.41	-	0.24
	No Conditioning	6.70	0.27	0.03	0.25	-	-
Diff-A-Riff $^{\text{CFG}=1}_{T=10}$	$CLAP_A$ + *Context*	1.50	0.06	0.09	0.54	0.54	-
	$CLAP_T$ + *Context*	6.23	0.19	0.03	0.17	0.00	0.23
	Context only	6.59	0.25	0.03	0.24	0.00	-
	$CLAP_A$ only	1.57	0.06	0.09	0.52	-	-
	$CLAP_T$ only	6.26	0.22	0.03	0.20	-	0.22
	No Conditioning	7.67	0.28	0.03	0.20	-	-

$^a\times 10^{-4}$, ^{b} obtained from white noise, ^{c} obtained by using a random accompaniment from the dataset

Table 1. Objective metrics using two configurations, *Diff-A-Riff*$^{\text{CFG}=1.25}_{T=30}$ and *Diff-A-Riff*$^{\text{CFG}=1}_{T=10}$, and different conditional settings (see Sec. 4.3.1). We compare against higher bounds obtained from the *real* validation set, and lower bounds obtained from random *noise* or random pairs (*Real*, *Random acc.*). Some cells are empty for APA and CS in the case of context and text-free generation respectively.

The reference segments are derived from music pieces within the evaluation set by summing all tracks in a multitrack, excluding one track, which is reserved to serve as the original accompaniment. Each accompaniment is presented mixed with the reference segment, with slight panning applied to the right to aid in distinguishing between them. The five accompaniments include the original accompaniment, a randomly selected one from the evaluation set, and three generated by our model under different conditional setups: ($CLAP_A$ + *Context*), ($CLAP_T$ + *Context*) and (*Context* only). To remove a potential bias toward better-quality audio, the original and random segments are encoded and decoded through the CAE.

For both studies, we used the GoListen platform [55]. All audio segments are normalized to a loudness of -20 dB LUFS and not cherry-picked. Sample questions are available on the accompanying website.

5. RESULTS & DISCUSSION

5.1 Objective Evaluation

Fig. 2 shows the MMD2 score of our model as a function of the number of denoising steps, with each line corresponding to a different conditional setting (see Sec. 4.3.1), all without classifier-free guidance (CFG). Note that MMD2 compares the distributions of embeddings of generated and real audio in CLAP's latent space. This means it indicates not only audio quality but also how well the distributions of generated instrument types and timbres match the test data distribution. For this reason, when using audio CLAP conditioning ($CLAP_A$), the results are considerably better, as we force the timbre distribution to be equal to the test data distribution (by using embeddings of that distribution as conditioning). However, the improvement of the results when increasing the number of denoising steps can be considered independent of the timbre distribution.

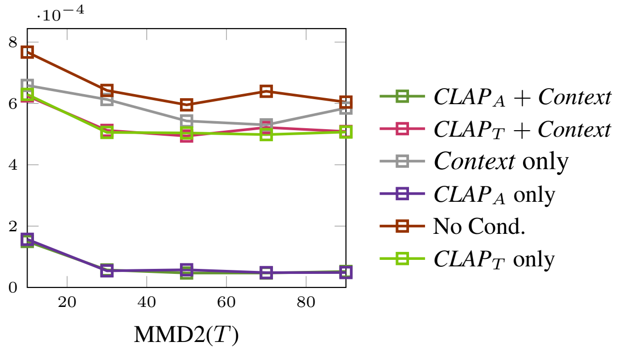

Figure 2. MMD2 as a function of the number of denoising steps T for various conditional settings (see Sec. 5.1).

Based on the results described above we perform a grid search over diffusion steps (T) and multi-source classifier-free guidance strength ($CFG_{<source>}$). $T = 30$ steps and $\text{CFG}_{Context} = \text{CFG}_{CLAP} = 1.25$ yields the best results. We denote this configuration as *Diff-A-Riff*$^{\text{CFG}=1.25}_{T=30}$ in Tab. 1. In addition, we compare to a specific configuration that achieves real-time performance[4] on a CPU with acceptable quality using $T = 10$ steps and CFG $= 1$, denoted as *Diff-A-Riff*$^{\text{CFG}=1}_{T=10}$. For reference, we also include metrics computed on *real* data and a *lower bound*, calculated from white noise in the case of quality metrics or random real pairs for input adherence metrics (APA and CS). The overall trend suggests that dense conditioning information helps the model in all metrics: audio quality, *coverage* and *density*, as well as APA. Also, the results could suggest a slight dependency of the model on $CLAP_A$ embeddings, with metrics close to real data, even in the absence of music context. Only *density* exhibits a minor drop without con-

[4] 95 seconds of audio in 73 seconds on an AMD EPYC 7502P

text conditioning, suggesting that generating from a silent context leads to samples that sometimes fall in low-density regions of the CLAP space.

The impact of Classifier-free Guidance (CFG) on MMD2 can be estimated by comparing Fig. 2, that displays results without CFG, with Tab. 1, where CFG was used. In Fig. 2, the metrics for both ($CLAP_A$ + *Context*) and ($CLAP_A$ only) converge towards an MMD2 of 0.5, while the corresponding values in Tab. 1 show a reduction to about half this figure. For ($CLAP_T$ + *Context*) and ($CLAP_T$ *only*), the MMD2 drops from approximately 5 to about 4 with CFG, and for *Context only*, it decreases from around 6 to approximately 5.

Finally, we calculate the Clap Score (CS) for text-conditioned generation ($CLAP_T$ + *Context*, $CLAP_T$ *only*). We compare *Diff-A-Riff*$^{\text{CFG}=1.25}_{T=30}$ against random pairs of text and real audio (CS=-0.07), suggesting that the model is only somewhat responsive to text prompts (CS=0.25).

5.2 Subjective Evaluation

Table 2 presents the outcomes of the Subjective Audio Quality (SAQ) test based on 74 users who each rated 32 audio segments, resulting in 2368 ratings. In this test, we compare results against leading baselines (see Sec. 4.3.3) and *real* audio data. Our analysis includes a comparison between these benchmarks and the audio generated by Diff-A-Riff in both mono (ch=1) and pseudo-stereo (ch=2) formats (see Sec. 4.3.1). Results show that the pseudo-stereo samples generated by Diff-A-Riff received ratings that are statistically indifferent from *real* audio ratings (p-value=0.79), indicating that participants found the audio quality of the generations indistinguishable from real data. This outcome is particularly remarkable given that Diff-A-Riff was not explicitly trained on complete musical pieces nor stereo music generation, but is still competitive to other models. Further, it highlights the influence of stereo imaging on the perceived audio quality.

Tab. 3 shows the results for the Subjective Audio Prompt Adherence (SAPA) listening test based on 35 users, each rating 25 accompaniments. The results include ratings scored by *real* accompaniments , *random* accompaniments , and the various conditional settings described in Sec. 4.3.1. Following the trend of previous results, the default setting ($CLAP_A$ + *Context*) scores the closest to *real* accompaniments, suggesting that the model can effectively adapt to the context under this setting. When conditioned on (*Context* only) , Diff-A-Riff is rated worse but still significantly better than *random* accompaniments. Further, for ($CLAP_T$+*Context*), the accompaniments are rated the lowest. A reason could be that the overall quality is worse because CLAP embeddings of text prompts have not been shown during training. Another problem could be that the randomly chosen text prompt is incompatible with the provided music context (e.g., "A drum machine with electronic textures" with an acoustic blues context), which reduces perceived adherence due to conflicting styles.

Overall, SAPA results are interesting given that APA (see Tab. 1) suggested rather pessimistic results for ($CLAP_T$ + *Context*) and (*Context* only). This could potentially be attributed to APA's sensitivity to audio quality and timbre differences between reference and candidate sets.

	SR/Ch	Params	RTF^a	Solo	Songs
Real data	44.1/**2**	-	-	3.5 ± 0.2	3.8 ± 0.2
MusicGen	32/1	3.3B	0.4	3.1 ± 0.2	3.2 ± 0.2
StableAudio	44.1/**2**	1B	11.8	2.5 ± 0.2	3.0 ± 0.2
AudioLDM2	**48**/1	712M	0.4	2.6 ± 0.2	2.0 ± 0.2
Diff-A-Riff	**48/2**	**500M**	**13.5** (0.57)	**3.4 ± 0.1**	**3.8 ± 0.1**
	48/1		**19** (1.3)	2.8 ± 0.1	3.2 ± 0.1

[a] NVIDIA A100 (CPU : AMD EPYC 7502P)

Table 2. Comparison of Diff-A-Riff to baselines. We include sampling rate in kHz and number of channels (*SR/Ch*), the total number of parameters *Params* (without CLAP), the Real Time Factor (*RTF*, the ratio of generated time over inference time, for 95 second-long audios) on GPU (and CPU for our model), as well as the SAQ32 values and 95% confidence intervals for the subjective audio quality assessment of *Solo* instruments and complete *Songs*.

	Cond. Signal	SAPA
Real	-	70.1 ± 4.5
Random	-	12.3 ± 3.0
Diff-A-Riff	$CLAP_A$ + *Context*	**62.4 ± 4.4**
	$CLAP_T$ + *Context*	37.6 ± 4.2
	Context only	42.3 ± 4.3

Table 3. Results for SAPA (see Sec. 4.3.3). The table includes results of Diff-A-Riff using different conditional settings, with 95% confidence intervals.

5.3 Control Mechanisms

In the accompanying website, we show examples of *Diff-A-Riff* generations for different inference settings (see Sec. 4.3.1). We also showcase other controls that naturally emerge from the denoising process, such as *in/out-painting* or the generation of *variations* and *loops*, as well as controls derived from the manipulation of CLAP embeddings, e.g., text-audio *Interpolations*.

6. CONCLUSION

This work introduced Diff-A-Riff, a Latent Diffusion Model capable of generating instrumental accompaniments adapted to a user-provided musical audio context. It can be controlled based on style audio references, text prompts, or both. We also proposed a simple method for producing pseudo-stereo audio. By exploiting the efficiency of a Consistency Autoencoder, Diff-A-Riff can generate 48 kHz sample rate pseudo-stereo audio with unprecedented speed and quality. Through extensive objective and subjective evaluation, we showed that our model achieves state-of-the-art audio quality, adapts to various conditional settings, and generates content that adheres to pre-existing musical audio contexts. We believe this work represents a significant step towards AI-assisted music production tools that prioritize artist-centric interactions, enriching the landscape of human-machine music co-creation.

7. ETHICS STATEMENT

Sony Computer Science Laboratories is committed to exploring the positive applications of AI in music creation. We collaborate with artists to develop innovative technologies that enhance creativity. We uphold strong ethical standards and actively engage with the music community and industry to align our practices with societal values. Our team is mindful of the extensive work that songwriters and recording artists dedicate to their craft. Our technology must respect, protect, and honour this commitment.

Diff-A-Riff supports and enhances human creativity and emphasises the artist's agency by providing various controls for generating and manipulating musical material. By generating a stem at a time, the artist remains responsible for the entire musical arrangement.

Diff-A-Riff has been trained on a dataset that was legally acquired for internal research and development; therefore, neither the data nor the model can be made publicly available. We are doing our best to ensure full legal compliance and address all ethical concerns.

8. REFERENCES

[1] U. Singer, A. Polyak, T. Hayes, X. Yin, J. An, S. Zhang, Q. Hu, H. Yang, O. Ashual, O. Gafni, D. Parikh, S. Gupta, and Y. Taigman, "Make-A-Video: Text-to-video generation without text-video data," in *Proc. of the 11th International Conference on Learning Representations, ICLR*, 2023.

[2] J. Betker, G. Goh, L. Jing, TimBrooks, J. Wang, L. Li, LongOuyang, JuntangZhuang, JoyceLee, YufeiGuo, WesamManassra, PrafullaDhariwal, CaseyChu, YunxinJiao, and A. Ramesh, "Improving image generation with better captions," in *CoRR*, 2023.

[3] A. Agostinelli, T. I. Denk, Z. Borsos, J. H. Engel, M. Verzetti, A. Caillon, Q. Huang, A. Jansen, A. Roberts, M. Tagliasacchi, M. Sharifi, N. Zeghidour, and C. H. Frank, "MusicLM: Generating Music From Text," in *CoRR*, 2023.

[4] J. Copet, F. Kreuk, I. Gat, T. Remez, D. Kant, G. Synnaeve, Y. Adi, and A. Défossez, "Simple and controllable music generation," in *Advances in Neural Information Processing Systems 36 NeurIPS*, 2023.

[5] Z. Evans, C. Carr, J. Taylor, S. H. Hawley, and J. Pons, "Fast timing-conditioned latent audio diffusion," in *CoRR*, 2024.

[6] H. Liu, Q. Tian, Y. Yuan, X. Liu, X. Mei, Q. Kong, Y. Wang, W. Wang, Y. Wang, and M. D. Plumbley, "AudioLDM 2: Learning holistic audio generation with self-supervised pretraining," in *arXiv*, 2023.

[7] B. Elizalde, S. Deshmukh, M. A. Ismail, and H. Wang, "CLAP learning audio concepts from natural language supervision," in *Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing ICASSP*, 2023.

[8] M. Pasini, S. Lattner, and G. Fazekas, "Music2latent: Consistency autoencoders for latent audio compression," in *Proc. of the International Society for Music Information Retrieval (ISMIR)*, 2024.

[9] T. Karras, M. Aittala, T. Aila, and S. Laine, "Elucidating the design space of diffusion-based generative models," in *Proc. of the 36th Conference on Neural Information Processing Systems (NeurIPS)*, 2022.

[10] A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. W. Senior, and K. Kavukcuoglu, "WaveNet: A generative model for raw audio," in *Proc. of the 9th ISCA Speech Synthesis Workshop*, 2016.

[11] S. Mehri, K. Kumar, I. Gulrajani, R. Kumar, S. Jain, J. Sotelo, A. C. Courville, and Y. Bengio, "SampleRNN: An unconditional end-to-end neural audio generation model," in *Proc. of 5th International Conference on Learning Representations, ICLR*, 2017.

[12] I. J. Goodfellow, J. Pouget-Abadie *et al.*, "Generative adversarial nets," in *Advances in Neural Information Processing Systems 27*, Dec. 2014, pp. 2672–2680.

[13] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," in *2nd International Conference on Learning Representations (ICLR)*, Apr. 2014.

[14] A. Caillon and P. Esling, "RAVE: A variational autoencoder for fast and high-quality neural audio synthesis," in *CoRR*, 2021.

[15] M. Pasini and J. Schlüter, "Musika! Fast infinite waveform music generation," in *Proc. of the 23rd International Society for Music Information Retrieval Conference, ISMIR*, 2022.

[16] J. Nistal, S. Lattner, and G. Richard, "DrumGAN: Synthesis of drum sounds with timbral feature conditioning," in *Proc. of the 21st International Society for Music Information Retrieval Conference, ISMIR*, 2020.

[17] J. Nistal, C. Aouameur, I. Velarde, and S. Lattner, "DrumGAN VST: A plugin for drum sound analysis/synthesis with autoencoding generative adversarial networks," in *Proc. of International Conference on Machine Learning ICML, Workshop on Machine Learning for Audio Synthesis, MLAS*, 2022.

[18] S. Rouard and G. Hadjeres, "CRASH: raw audio score-based generative modeling for controllable high-resolution drum sound synthesis," in *Proc. of the 22nd International Society for Music Information Retrieval Conference, ISMIR*, 2021.

[19] Q. Huang, D. S. Park, T. Wang, T. I. Denk, A. Ly, N. Chen, Z. Zhang, Z. Zhang, J. Yu, C. H. Frank, J. H. Engel, Q. V. Le, W. Chan, and W. Han, "Noise2Music: Text-conditioned music generation with diffusion models," in *CoRR*, 2023.

[20] G. Zhu, Y. Wen, M. Carbonneau, and Z. Duan, "EDMSound: Spectrogram based diffusion models for efficient and high quality audio synthesis," in *CoRR*, 2023.

[21] F. Schneider, "ArchiSound: Audio generation with diffusion," in *CoRR*, 2023.

[22] P. Dhariwal, H. Jun, C. Payne, J. W. Kim, A. Radford, and I. Sutskever, "Jukebox: A generative model for music," in *CoRR*, 2020.

[23] J. D. Parker, J. Spijkervet, K. Kosta, F. Yesiler, B. Kuznetsov, J. Wang, M. Avent, J. Chen, and D. Le, "StemGen: A music generation model that listens," in *CoRR*, 2023.

[24] M. Pasini, M. Grachten, and S. Lattner, "Bass accompaniment generation via latent diffusion," in *IEEE International Conference on Acoustics, Speech and Signal Processing ICASSP*. IEEE, 2024.

[25] N. Zeghidour, A. Luebs, A. Omran, J. Skoglund, and M. Tagliasacchi, "SoundStream: An end-to-end neural audio codec," in *IEEE ACM Trans. Audio Speech Lang. Process.*, 2022.

[26] A. Défossez, J. Copet, G. Synnaeve, and Y. Adi, "High fidelity neural audio compression," in *CoRR*, 2022.

[27] H. F. García, P. Seetharaman, R. Kumar, and B. Pardo, "VampNet: Music generation via masked acoustic token modeling," in *Proc. of the 24th International Society for Music Information Retrieval Conference, ISMIR*, 2023.

[28] A. Ziv, I. Gat, G. L. Lan, T. Remez, F. Kreuk, A. Défossez, J. Copet, G. Synnaeve, and Y. Adi, "Masked audio generation using a single non-autoregressive transformer," in *CoRR*, 2024.

[29] F. Schneider, O. Kamal, Z. Jin, and B. Schölkopf, "Moûsai: Text-to-music generation with long-context latent diffusion," in *CoRR*, 2023.

[30] P. Li, B. Chen, Y. Yao, Y. Wang, A. Wang, and A. Wang, "JEN-1: text-guided universal music generation with omnidirectional," in *CoRR*, 2023.

[31] J. Song, C. Meng, and S. Ermon, "Denoising diffusion implicit models," in *Proc. of the 9th International Conference on Learning Representations, ICLR*, 2021.

[32] Y. Wu, C. Chiu, and Y. Yang, "Jukedrummer: Conditional beat-aware audio-domain drum accompaniment generation via transformer VQ-VAE," in *Proc. of the 3rd International Society for Music Information Retrieval Conference, ISMIR*, 2022.

[33] S.-L. Wu, C. Donahue, S. Watanabe, and N. J. Bryan, "Music controlnet: Multiple time-varying controls for music generation," in *CoRR*, 2023.

[34] Y. Zhang, Y. Ikemiya, G. Xia, N. Murata, M. A. M. Ramírez, W. Liao, Y. Mitsufuji, and S. Dixon, "Musicmagus: Zero-shot text-to-music editing via diffusion models," *CoRR*, 2024.

[35] H. Manor and T. Michaeli, "Zero-shot unsupervised and text-based audio editing using DDPM inversion," *CoRR*, 2024.

[36] Z. Novack, J. J. McAuley, T. Berg-Kirkpatrick, and N. J. Bryan, "DITTO: diffusion inference-time t-optimization for music generation," *CoRR*, 2024.

[37] M. Levy, B. D. Giorgi, F. Weers, A. Katharopoulos, and T. Nickson, "Controllable music production with diffusion models and guidance gradients," *CoRR*, 2023.

[38] M. Grachten, S. Lattner, and E. Deruty, "BassNet: A variational gated autoencoder for conditional generation of bass guitar tracks with learned interactive control," in *Applied Sciences*, 2020.

[39] C. Donahue, A. Caillon, A. Roberts, E. Manilow, P. Esling, A. Agostinelli, M. Verzetti, I. Simon, O. Pietquin, N. Zeghidour, and J. H. Engel, "Singsong: Generating musical accompaniments from singing," in *CoRR*, 2023.

[40] S. Lattner and M. Grachten, "High-level control of drum track generation using learned patterns of rhythmic interaction," in *2019 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics, WASPAA*. IEEE, 2019.

[41] E. Postolache, G. Mariani, L. Cosmo, E. Benetos, and E. Rodolà, "Generalized multi-source inference for text conditioned music diffusion models," *CoRR*, 2024.

[42] G. Mariani, I. Tallini, E. Postolache, M. Mancusi, L. Cosmo, and E. Rodolà, "Multi-source diffusion models for simultaneous music generation and separation," *CoRR*.

[43] Y. Song, P. Dhariwal, M. Chen, and I. Sutskever, "Consistency models," in *International Conference on Machine Learning, ICML*, 2023.

[44] Y. Song and P. Dhariwal, "Improved techniques for training consistency models," *arXiv preprint arXiv:2310.14189*, 2023.

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

[46] J. Sohl-Dickstein, E. A. Weiss *et al.*, "Deep unsupervised learning using nonequilibrium thermodynamics," in *Proceedings of the 32nd International Conference on Machine Learning, ICML*, 2015.

[47] T. Karras, M. Aittala, T. Aila, and S. Laine, "Score-based generative modeling through stochastic differential equations," in *Proc. of the International Conference on Learning Representations (ICLR)*, 2021.

[48] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," in *7th International Conference on Learning Representations, ICLR*, 2019.

[49] J. Ho and T. Salimans, "Classifier-free diffusion guidance," *arXiv preprint arXiv:2207.12598*, 2022.

[50] M. Binkowski, D. J. Sutherland, M. Arbel, and A. Gretton, "Demystifying MMD gans," in *Proc. of the 6th International Conference on Learning Representations, ICLR*, 2018.

[51] K. Kilgour, M. Zuluaga, D. Roblek, and M. Sharifi, "Fréchet audio distance: A reference-free metric for evaluating music enhancement algorithms," in *Proc. of the 20th Conference of the International Speech Communication Association (InterSpeech)*, 2019.

[52] M. F. Naeem, S. J. Oh, Y. Uh, Y. Choi, and J. Yoo, "Reliable fidelity and diversity metrics for generative models," in *Proc. of the 37th International Conference on Machine Learning, ICML*, 2020.

[53] R. Huang, J. Huang, D. Yang, Y. Ren, L. Liu, M. Li, Z. Ye, J. Liu, X. Yin, and Z. Zhao, "Make-an-audio: Text-to-audio generation with prompt-enhanced diffusion models," in *Proc. of the International Conference on Machine Learning, ICML*, 2023.

[54] M. Grachten and J. Nistal, "Audio Prompt Adherence: A measure for evaluating musical accompaniment systems," in *CoRR*, 2024.

[55] D. Barry, Q. Zhang, P. W. Sun, and A. Hines, "Go listen: An end-to-end online listening test platform." *Journal of Open Research Software*, vol. 9, no. 1, 2021.