Sketch2Sound: Controllable Audio Generation via Time-Varying Signals and Sonic Imitations

Hugo Flores García[io], Oriol Nieto[i], Justin Salamon[i], Bryan Pardo[o], Prem Seetharaman[i]
[i]*Adobe Research,* [o]*Northwestern University*
hugofloresgarcia@u.northwestern.edu

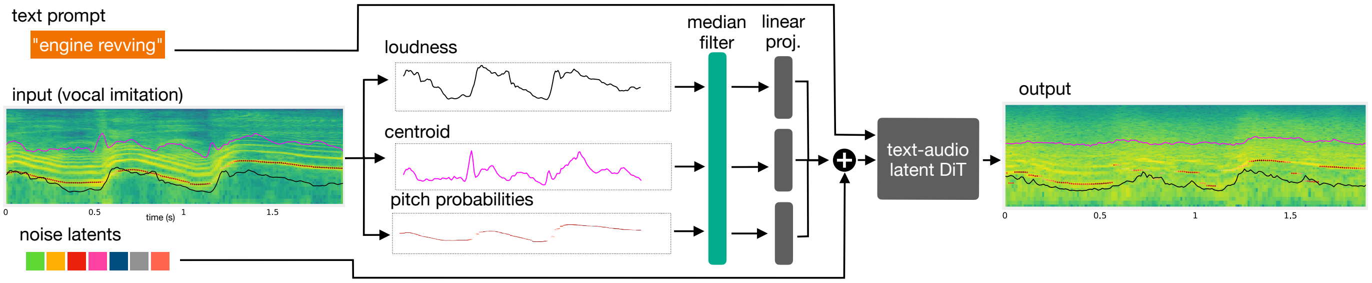

Fig 1. Overview of Sketch2Sound. We extract three control signals from any input sonic imitation: loudness, spectral centroid (i.e., brightness) and pitch probabilities. We encode the signals and add them to the latents used as input to a DiT text-to-sound generation system. **We *strongly* encourage the reader to listen to this example and many more at: https://hugofloresgarcia.art/sketch2sound**.

***Abstract*—We present Sketch2Sound, a generative audio model capable of creating high-quality sounds from a set of interpretable time-varying control signals: loudness, brightness, and pitch, as well as text prompts. Sketch2Sound can synthesize arbitrary sounds from sonic imitations (i.e., a vocal imitation or a reference sound-shape). Sketch2Sound can be implemented on top of any text-to-audio latent diffusion transformer (DiT), and requires only 40k steps of fine-tuning and a single linear layer per control, making it more lightweight than existing methods like ControlNet. To synthesize from sketchlike sonic imitations, we propose applying random median filters to the control signals during training, allowing Sketch2Sound to be prompted using controls with flexible levels of temporal specificity. We show that Sketch2Sound can synthesize sounds that follow the gist of input controls from a vocal imitation while retaining the adherence to an input text prompt and audio quality compared to a text-only baseline. Sketch2Sound allows sound artists to create sounds with the semantic flexibility of text prompts and the expressivity and precision of a sonic gesture or vocal imitation. Sound examples are available at https://hugofloresgarcia.art/sketch2sound/.**

***Index Terms*—audio systems, sound generation, auto foley**

I. Introduction

Sound design is the craft of storytelling through sonic composition. Within sound design, Foley sound is a technique where special sound effects are designed and performed in sync to a film during post-production [1]. These sound scenes are typically performed by a Foley artist on a stage equipped with abundant sound instruments and other soundmaking materials[1]. Foley sound is a skilled and gestural performance art: performing a sound scene with sound-making objects and instruments (instead of arranging pre-recorded samples post hoc) allows sound artists to create fluent and temporally aligned sounds with a "human" (i.e., gestural) touch. Adding this gestural touch to the resulting sound composition often results in a sonic product of great aesthetic and production value.

Recent research in generative modeling for sound has paved the way for text-to-sound systems [2]–[4], where a user can create sound samples from text descriptions of a sound (e.g., "explosion"). While the text-to-sound paradigm can help a sound designer find sounds more quickly (and, perhaps in the future, with a higher degree of specificity), a sound designer still has to painstakingly modify the temporal characteristics of the generated sound so that they can be in sync with the visuals in the editing timeline. This is in opposition to the natural way that Foley artists gesturally create sound effects by physically performing with physical soundmaking objects.

To overcome the drawbacks of a purely text-to-audio interaction, several works in the music domain sought to condition generative models on audio [5], parallel instrument stems [6], melody [7], sound event timestamps and frequency [8], or multiple structural control signals like song structure and dynamics [9]. Notably, [10] condition an audio VAE on control signals such as brightness and loudness, though their experiments are limited to models trained on narrow sound distributions (e.g., violin, darbouka, speech) and not a multi-distribution text-to-audio model. For speech, [11] proposes a fully interpretable and disentangled representation for speech generation and editing, which allows for fine-grained control over the pitch, loudness, and phonetic pronunciation of speech.

The human voice is a gestural sonic instrument [12]: it allows us to realize sounds without having to perform any symbolic abstraction (i.e., putting a sound into words) beforehand. When humans communicate audio concepts to other people, they typically combine language and vocal imitation [13]–[15], and recent work has shown its utility for query-by-example search of audio databases [16]–[18]. This is a more natural method than describing the evolution of pitch, timing, and timbre via pure text descriptions [13], and voice-driven sound synthesis interactions have been of interest long before modern generative modelling for their embodied capabilities [19]–[24].

We propose Sketch2Sound: a text-to-audio model that can create high-quality sounds from sonic imitation prompts by following interpretable, fine-grained time-varying control signals that can be easily extracted from any audio signal at different levels of temporal detail: loudness, brightness (*spectral centroid*) and pitch. We expand upon previous work [25] by developing a method capable of following the loudness, brightness *and* pitch of a vocal imitation, with the option to drop any of the three controls. Additionally, we propose a technique that varies the temporal detail of the control signals used during training by applying median filters

[1] Example of a Foley artist performing a scene: youtu.be/WFVLWo5B81w

of different window sizes to the control signals before using them as input. This allows sound artists to specify the degree of temporal precision to which a generative model should follow the specified control signals, which improves sound quality in sounds that may be too hard to perfectly imitate with one's voice.

This method is not limited to just vocal imitation: any kind of sonic imitation can be used to drive our proposed generative model – we place the focus on vocal imitation due to people's innate ability to imitate sounds with our voices. Vocal imitations can always be augmented through other sonic gestures like clapping, tapping, playing instruments, etc. Sketch2Sound can be added to any existing latent diffusion transformer (DiT) sound generation model with as little as 40k fine-tuning steps. Unlike ControlNet methods [26], [27] that require an extra trainable copy of the entire neural network encoder, Sketch2Sound requires only a single linear layer per control.

Our experiments show that Sketch2Sound can generate sounds that closely follow the input control signals (loudness, spectral centroid, and pitch/periodicity) from a vocal imitation while still achieving a high degree of adherence to a text prompt and an audio quality comparable to the text-only pre-trained model. We show that our median filtering technique leads to improved audio quality and text adherence when generating sounds from vocal imitations. We also show that, during inference, a user can arbitrarily specify a degree of temporal detail by choosing a median filter size, allowing them to navigate the trade-off between strict adherence to the vocal imitations and audio quality + text adherence.

To the best of our knowledge, this is the first sound generation model capable of following vocal imitations *and* text prompts by conditioning on a set of holistic control signals suitable for generating sound objects with fine-grained, gestural control of pitch, loudness, and brightness. We believe Sketch2Sound will give sound artists a more expressive, controllable, and gestural interaction for generating sound-objects than existing text-to-audio and other conditional sound generation systems. **We highly encourage the reader to listen to our expansive set of audio examples demonstrating Sketch2Sound.** [2]

II. Method

We propose a method for conditioning an audio latent diffusion model on a set of interpretable, time-varying control signals that are suitable tasks creating variations of sounds and generating new sounds expressively via text-prompted sonic imitations.

A. Time-varying control signals for sound objects

We use the following control signals as conditioning for Sketch2Sound:

- **Loudness:** We extract the per-frame loudness of an audio signal by performing an A-weighted sum across the frequency bins in a magnitude spectrogram [11] and taking the RMS of the result.
- **Pitch and Periodicity:** We use the raw pitch probabilities of the CREPE [28], [29] ("tiny" variant) pitch estimation model. To avoid leaking timbral information in this signal, we zero out all probabilities below 0.1 in the pitch probability matrix.
- **Spectral Centroid** is defined as the center of mass of the frequency spectrum for a given audio frame. Frames with a higher centroid will be perceived as having a brighter timbre. To preprocess the centroid, we convert the signal from linear frequency space (i.e., Hz) to a continuous MIDI-like representation, scaled to roughly a $(0, 1)$ range by dividing the input signal by 127 (note G9, roughly 12.5kHz), which we found to stabilize the first steps of training.

Other momentary time-varying control signals may be used as well.

B. Conditioning a latent audio DiT on time-varying control signals

Refer to Figure 1 for a visual overview of our approach. We use a large pre-trained text-to-sound latent diffusion transformer (DiT), similar to the one described in [2], [30] (text-conditioned only, no timing conditioning) and adapt it to generate sounds conditioned on the time-varying control signals mentioned above. The latent diffusion model for text-to-sound generation has two parts: first, a variational autoencoder (VAE) compresses 48kHz mono audio to a sequence of continuous vectors of dim 64 at a rate of 40Hz. Then, a transformer model is trained to generate new sequences of latents, which can be decoded into audio using the VAE decoder. This text-to-audio DiT was pre-trained on a large mix of proprietary, licensed sound effect datasets and publicly available CC-licensed general audio datasets. Once the model is pre-trained, we fine-tune it for 40k steps and adapt it to handle our time-varying control signals.

Because the time-varying control signals can be easily and efficiently extracted from any audio signal on the fly, we can fine-tune the pre-trained text-to-audio model in a self-supervised manner: Given any input audio signal, we extract the three control signals (loudness, centroid, pitch) from the audio signal and use them as conditioning for the model during fine-tuning. The model is then fine-tuned using the same recipe used during training: learning the reverse diffusion process from a set of noisy latents with text conditioning, along with our proposed control conditioning.

To align the time-varying control signals with the latents from our text-to-sound DiT, the control signals must be extracted at the same frame rate as the audio VAE latents or interpolated to this frame rate. This allows us to perform a simple conditioning method: condition a latent diffusion model ϵ_θ by simply adding a linear projection layer from our control signals to the noisy latents used as input to the diffusion model. Since these time-varying control signals are highly localized to their given time frame, a simple linear layer suffices to incorporate each time-varying signal as conditioning to the model.

Given the noisy latent vector sequence $\mathbf{z} \in \mathbb{R}^{D \times N}$ that is used as input to a latent diffusion model ϵ_θ with embedding dimension D and sequence length N, we introduce our time-varying conditioning signal $\mathbf{c}_{ctrl} \in \mathbb{R}^{K \times N}$ with dimension K and sequence length N to the latent $\mathbf{z}$ by applying a trainable linear projection layer $p_\theta(\mathbf{c}_{ctrl}) \in \mathbb{R}^{D \times N}$ to the input conditioning, and adding the result directly to the latents used as input to the diffusion model: $\mathbf{z}_{ctrl} = p_\theta(\mathbf{c_{ctrl}}) + \mathbf{z}$. We can repeat this process for any number of time-varying control signals that we'd like to condition our latent diffusion model.

During fine-tuning, the loss configuration for the model does not change from the one used in original training; that is, we do not apply any reconstruction losses for the control curves themselves, removing the need to pass through the VAE decoder during fine-tuning. Despite not measuring loss on the control signals, providing them as input during fine-tuning is sufficient for the model to condition generation on them. As a result, these signals become useful for control at inference time. To ensure we can generate without requiring all the control signals, we perform dropout during fine tuning on the control signals by zero-ing out the control embeddings $p_\theta(\mathbf{c_{ctrl}})$ before they are added to the diffusion model's latents $\mathbf{z}$. We drop each control signal (as well as the text conditioning) individually with a 20% probability, with an added 20% probability of dropping out all signals together. Likewise, these signals can be dropped out at inference if a user wishes to control only one or two controls while omitting others.

At inference time, we follow the two-conditioning classifier-free guidance setup described in [31]. We use guidance scales s_{ctrl} and

[2] https://hugofloresgarcia.art/sketch2sound

TABLE I: Control Signal Evaluation and Sketch Type Ablation. **Control Signals**: Introducing more control signals improves control adherence while trading off text adherence and audio quality. **Sketch Types:** Using median filters to create sketch curves achieves the best trade-off between text adherence/audio quality and control adherence, compared to a no filtering baseline and a low pass filtering approach.

	Control Signal	**Sketch Type**	**Text Adherence ↑**	**Control Adherence (Error) ↓**				**Audio Quality (FAD) ↓**	
			CLAP Score	**RMS (dB)**	**Centroid (st)**	**Pitch (st)**	**Chroma (st)**	**VGGish**	**CLAP**
control signals	text-only	median (sz 10)	0.2̲73	13.41	10.34	13.91	2.96	2.57	0.27
	loudness (ldns)	median (sz 10)	0.230	3.88	10.37	12.45	2.96	2.69	0.296
	ldns+centroid	median (sz 10)	0.219	3.60	4.39	11.17	2.87	2.67	0.306
	ldns+centroid+pitch (ours)	median (sz 10)	0.211	3.60	4.43	1.49	0.48	2.51	0.312
sketch types	ldns+centroid+pitch	low pass	0.166	2.19	3.33	0.44	0.23	3.30	0.363
	ldns+centroid+pitch	no filters	0.152	1.87	3.21	0.45	0.21	3.53	0.379

s_{text}, which can be used to trade off the guidance strength for the control signal conditioning and text conditioning, respectively. We find that using a single guidance scale for all three control signals together (s_{ctrl}) is a sufficient approach, but future work may explore the effect of applying guidance strengths to each time-varying control independently of each other. Anecdotally, we find that setting s_{text} to a value of 5 and s_{ctrl} to 1 achieves results with good text adherence while following the contour provided in the sonic imitation controls.

C. Creating sketchlike controls via control-rate filtering

Even though the human voice is remarkable at imitating sounds, there will still be a mismatch between the control signals of the target sound and the control signals of the vocal imitation. To overcome this issue, we propose a technique to make the controls sketchlike by applying random median filters to the control signals at different window sizes (1-25 control frames) before they are used as input to the model. This filtering technique can help mitigate the mismatch in temporal specificity between the vocal imitation and target sound and help the model produce higher-quality sounds from sketchlike vocal imitations. Our experiments show that during inference, a sound artist is free to adjust the control rate of their input control signals, giving them an interpretable control over the trade-off between text-prompt adherence and fine temporal precision.

III. Experimental Design

Our experiments evaluate Sketch2Sound's ability to synthesize high-quality sounds from vocal imitations. Except for the text-only baseline (which doesn't need fine-tuning), we fine-tune every model for 40k steps with the same configuration used for training. For all fine-tunings, we use the text-only baseline as the starting checkpoint.

The main dataset used for evaluation is VimSketch [32], which consists of approximately 12k vocal imitations, each with a text description and reference sound. For each model variant, we generate 10k examples with durations of up to 5 seconds using the vocal imitation and text description as conditioning. We evaluate Sketch2Sound along the following characteristics:

Audio Quality: To measure a model's ability to synthesize high-quality sound effects, we compute the Frechét Audio Distance (FAD) [33] using a proprietary dataset of 40k high-quality sound effects as the reference set, and 10k sounds generated from vocal imitations from the VimSketch dataset as the evaluation set, as suggested by [34]. We report the FAD for VGGish [35] and LAION-CLAP [36] embeddings. **Text Adherence:** We measure how well our generated audio adheres to the target text prompt by computing the CLAP embedding cosine similarity [36] between audio generated from a sonic imitation and the target text prompt for every example.

Control Signal Adherence: Finally, to measure the adherence of the generated audio to the vocal imitations, we measure the error (L1) between the input and generated control signals (loudness, centroid, pitch) only on non-silent (loudness $> -40dB$) frames. We measure loudness error in dBFS RMS and centroid error in semitones (st). We report the following pitch metrics: pitch error (st), chroma error (from the predicted pitch), and periodicity error, as estimated by torchcrepe [29]. We only measure pitch and chroma error on *voiced* frames, i.e., where the periodicity predicted by torchcrepe is greater than a threshold of 0.5 for both the vocal imitation and the model output.

IV. Experiments

A. Control signals

First, we validate that Sketch2Sound can synthesize sounds from a reference vocal imitation and a text prompt while achieving a competitive audio quality, text adherence, and adherence to the vocal imitation. We fine-tune three models, each with a different set of control signals (loudness only, loudness+centroid, loudness+centroid+pitch), and compare the performance of these models, along with a text-only baseline, using the metrics discussed in Section III. For all model variants, we use a fixed median filter window size of 10 at inference.

B. Sketch type ablation

To observe the effect of our random median filtering as a way of creating sketchlike controls, we compare our approach to a no-filter baseline, as well as an alternative approach using low-pass filters instead of median filters to remove fine temporal detail from the controls during training. We train our model with random median filters with window sizes ranging between 1 and 25 control frames (1 frame = 25 ms). At inference, we use a fixed median filter size of 10. For our low-pass filter approach, we apply random low-pass filters ranging from 5Hz-20Hz and use a fixed cutoff (10Hz) at inference. Likewise, we compare each variant in this experiment in terms of audio quality, text and control adherence, following Section III.

C. Inference-time control rates

To verify whether our median filtering approach allows for an inference-time trade-off between text adherence and fine-temporal control, we observe the performance of our model trained with random median filters at different inference-time temporal resolutions. Specifically, we generate samples using our model at different inference-time median filter sizes of $\{1, 5, 10, 15, 20, 25\}$ observe their performance on the metrics (Section III).

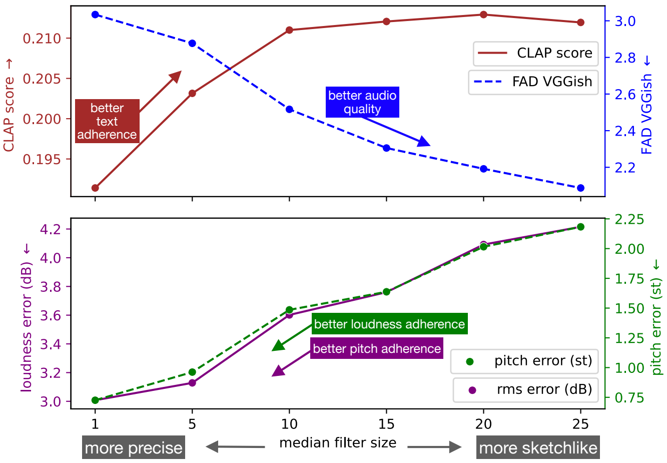

Fig. 2: At inference, larger median filters are more sketchlike and can lead to higher audio quality, while smaller filters are more precise and may lead to lower audio quality if the vocal imitations aren't precise enough, giving the sound artist a choice over this trade-off.

V. Results and Discussion

A. *Control signals*

Table I validates Sketch2Sound's ability to synthesize sounds using control signals extracted from a vocal imitation + a text prompt while achieving comparable audio quality to a text-only baseline. We incrementally add each control (loudness, centroid, pitch), and observe each model's performance in terms of text adherence (CLAP score), control signal adherence, and audio quality (FAD).

Conditioning a model on a time-varying control signal (e.g., loudness, centroid or pitch) significantly improves the adherence to that control signal compared to when the model is not conditioned on that control. Since loudness, centroid, and pitch are often correlated in natural sounds, incorporating a single conditioning (i.e., loudness) also slightly improves the control adherence for other control signals.

In terms of text adherence and audio quality, introducing control signals produces a slight decrease in audio quality and text adherence. Empirically, we find that this difference is near negligible when compared to the text-only baseline in most cases. We also find that the quality of the generated audio can be a function of how well the user imitates the characteristics of the target sound: better-performed vocal imitations lead to higher-quality generated sound effects.

B. *Sketch type ablation*

Table I shows that our median filter method makes Sketch2Sound robust to generating high-quality sound effects from sketchlike vocal imitation control signals, improving the audio quality and text adherence over a no-filter baseline and a low-pass method. Notably, our median filter method is able to achieve a higher CLAP score (text adherence) and lower FAD (audio quality), while trading off the control adherence to vocal imitations. **This trade-off, where lowering the control adherence to vocal imitations improves the audio quality is more desirable than a strict adherence to the controls since most vocal imitations cannot perfectly mimic the fine temporal behavior of target sounds. Generating sounds that exactly follow the vocal imitation controls (i.e. *"no filtering"*) results in audio that does not sound like the text, but "speechlike".**

C. *Inference-time control rates*

Our control-rate filtering method lets a sound artist use different-size median filters at inference time, allowing users to choose the desired amount of temporal detail needed for a particular voice-to-sound example. The results in Figure 2 show that Sketch2Sound can be used with different control-rate resolutions at inference time by using median filters of different sizes. Smaller filters achieve higher control adherence, at the cost of a lower audio quality and text adherence. We hypothesize that the decrease in text adherence and audio quality is due to the mismatch between vocal control signals and the target control signals suitable for generating Foley sounds. However, this flexibility allows one to use smaller filters (i.e., higher temporal resolution) when the vocal imitations are well-performed, and larger filters (i.e., lower temporal resolution) when the vocal imitations are impossible to precisely imitatate with the human voice.

D. *The semantics of control curves are implicitly modeled*

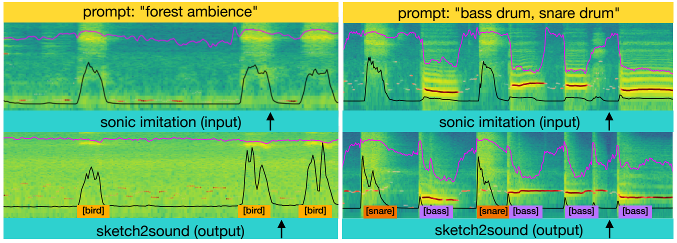

Fig. 3: (left) When prompted with "forest ambience", bursts of loudness in the controls become of birds without prompting the model to do so. (right) With "bass drum, snare drum", the model places snares in unpitched areas and bass drums in pitched areas.

We find that the control signals can manipulate the semantics of the generated signals. For example, using the text prompt "forest ambience" with a sonic imitation containing random bursts of loudness in it, we can synthesize bird sounds into those loudness bursts (see Figure 3), without having to use the prompt "birds" at all. The model follows the correlations between bursts of loudness and the presence of birds in recordings of forest ambience in the training data, and so generates bird sounds when prompted with loudness bursts in a "forest ambience" prompt. Likewise, with the prompt "snare drum, bass drum, drum beat", performing a sonic imitation with a series of pitched (bass drum) and unpitched (snare drum) sounds will successfully apply bass drum sounds on pitched regions, and snare drum on unpitched regions. Both of these examples are available on our accompanying website.

E. *Limitations*

We found that the centroid control tends to entangle the room tone of the input sonic imitation onto the generated audio. We believe this is because the room tone of the input audio is encoded by the centroid when no sound events are occurring in the input audio. A potential solution is to drop the controls when the signal is quiet, though this may fail with overcompressed inputs or loud room tones.

VI. Conclusion

Sketch2Sound is a generative sound model capable of generating sounds from text prompts and time-varying controls: loudness, brightness (spectral centroid), and pitch. Sketch2Sound is capable of generating sounds from sonic imitations and sketchlike control curves. Sketch2Sound is a lightweight adapter method, requiring only 40k steps of fine-tuning and a single linear adapter layer per control signal. We showed that Sketch2Sound can create arbitrary sounds via sonic imitations with a flexible amount of temporal specificity while retaining the semantics of a text prompt, making it a controllable, gestural, and expressive tool for sound artists.

Acknowledgments

H.F.G. would like to thank Zachary Novack for many meaningful conversations on diffusion, generative audio models, statistics, and median filters. The authors thank Rithesh Kumar for his insight into conditional generative modeling and Adolfo Hernandez Santisteban for valuable discussions in interface design and interaction for sound design practices.

References

[1] V. Ament, *The Foley Grail: The Art of Performing Sound for Film, Games, and Animation*, 3rd ed. Routledge, 2021. [Online]. Available: https://doi.org/10.4324/9781003008439

[2] Z. Evans, J. D. Parker, C. Carr, Z. Zukowski, J. Taylor, and J. Pons, "Long-form music generation with latent diffusion," 2024.

[3] D. Yang, J. Yu, H. Wang, W. Wang, C. Weng, Y. Zou, and D. Yu, "Diffsound: Discrete diffusion model for text-to-sound generation," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 31, pp. 1720–1733, 2023.

[4] 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," *Proceedings of the International Conference on Machine Learning*, 2023.

[5] H. F. Garcia, P. Seetharaman, R. Kumar, and B. Pardo, "Vampnet: Music generation via masked acoustic token modeling," in *International Society for Music Information Retrieval (ISMIR)*, 2023.

[6] J. D. Parker, J. Spijkervet, K. Kosta, F. Yesiler, B. Kuznetsov, J.-C. Wang, M. Avent, J. Chen, and D. Le, "Stemgen: A music generation model that listens," in *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP*, 2024.

[7] J. Copet, F. Kreuk, I. Gat, T. Remez, D. Kant, G. Synnaeve, Y. Adi, and A. Défossez, "Simple and controllable music generation," in *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

[8] Z. Xie, X. Xu, Z. Wu, and M. Wu, "Picoaudio: Enabling precise timestamp and frequency controllability of audio events in text-to-audio generation," *arXiv preprint arXiv:2407.02869*, 2024.

[9] Z. Novack, J. McAuley, T. Berg-Kirkpatrick, and N. J. Bryan, "DITTO: Diffusion inference-time t-optimization for music generation," in *International Conference on Machine Learning (ICML)*, 2024.

[10] N. Devis, N. Demerlé, S. Nabi, D. Genova, and P. Esling, "Continuous descriptor-based control for deep audio synthesis," in *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2023, pp. 1–5.

[11] M. Morrison, C. Churchwell, N. Pruyne, and B. Pardo, "Fine-grained and interpretable neural speech editing," in *Interspeech 2024*, 2024.

[12] T. Wishart, "The composition of" vox-5"," *Computer Music Journal*, vol. 12, no. 4, pp. 21–27, 1988.

[13] G. Lemaitre and D. Rocchesso, "On the effectiveness of vocal imitations and verbal descriptions of sounds," *The Journal of the Acoustical Society of America*, vol. 135, no. 2, pp. 862–873, 2014.

[14] G. Lemaitre, O. Houix, F. Voisin, N. Misdariis, and P. Susini, "Vocal imitations of non-vocal sounds," *PloS one*, vol. 11, no. 12, p. e0168167, 2016.

[15] P. S. Guillaume Lemaitre, Arnaud Dessein and K. Aura, "Vocal imitations and the identification of sound events," *Ecological Psychology*, vol. 23, no. 4, pp. 267–307, 2011.

[16] Y. Zhang, J. Hu, Y. Zhang, B. Pardo, and Z. Duan, "Vroom! a search engine for sounds by vocal imitation queries," in *Proceedings of the 2020 Conference on Human Information Interaction and Retrieval*, 2020. [Online]. Available: /assets/papers/zhang-et-al-2020.pdf

[17] D. S. Blancas and J. Janer, "Sound retrieval from voice imitation queries in collaborative databases," in *Audio Engineering Society Conference: 53rd International Conference: Semantic Audio*. Audio Engineering Society, 2014.

[18] Y. Zhang and Z. Duan, "Imisound: An unsupervised system for sound query by vocal imitation," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2016, pp. 2269–2273.

[19] S. Delle Monache, D. Rocchesso, F. Bevilacqua, G. Lemaitre, S. Baldan, and A. Cera, "Embodied sound design," *International Journal of Human-Computer Studies*, vol. 118, pp. 47–59, 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S107158191830260X

[20] A. Hazan, "Billaboop: Real-time voice-driven drum generator," in *Audio Engineering Society Convention 118*. Audio Engineering Society, 2005.

[21] J. Janer and M. De Boer, "Extending voice-driven synthesis to audio mosaicing," in *5th Sound and Music Computing Conference, Berlin*, vol. 4. Citeseer, 2008.

[22] A. D. Piccolo and D. Rocchesso, "Non-speech voice for sonic interaction: a catalogue," *Journal on Multimodal User Interfaces*, vol. 11, pp. 39–55, 2017.

[23] S. Fasciani and L. Wyse, "Mapping the voice for musical control," 2013.

[24] S. Baldan, S. Delle Monache, D. Rocchesso, H. Lachambre *et al.*, "Sketching sonic interactions by imitation-driven sound synthesis," in *Proceedings of the 13. Sound & Music Computing Conference*. Zentrum für Mikrotonale Musik und Multimediale Komposition (ZM4) Hochschule ..., 2016, pp. 47–54.

[25] Y. Chung, J. Lee, and J. Nam, "T-foley: A controllable waveform-domain diffusion model for temporal-event-guided foley sound synthesis," 2024. [Online]. Available: https://arxiv.org/abs/2401.09294

[26] S.-L. Wu, C. Donahue, S. Watanabe, and N. J. Bryan, "Music controlnet: Multiple time-varying controls for music generation," *IEEE/ACM Trans. Audio, Speech and Lang. Proc.*, vol. 32, p. 2692–2703, may 2024. [Online]. Available: https://doi.org/10.1109/TASLP.2024.3399026

[27] S. Zhao, D. Chen, Y.-C. Chen, J. Bao, S. Hao, L. Yuan, and K.-Y. K. Wong, "Uni-controlnet: All-in-one control to text-to-image diffusion models," *Advances in Neural Information Processing Systems*, 2023.

[28] J. W. Kim, J. Salamon, P. Li, and J. P. Bello, "Crepe: A convolutional representation for pitch estimation," in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018.

[29] M. Morrison, "torchcrepe," 2022. [Online]. Available: https://github.com/maxrmorrison/torchcrepe

[30] Z. Evans, C. Carr, J. Taylor, S. H. Hawley, and J. Pons, "Fast timing-conditioned latent audio diffusion," in *International Conference on Machine Learning (ICML)*, 2024.

[31] T. Brooks, A. Holynski, and A. A. Efros, "Instructpix2pix: Learning to follow image editing instructions," in *CVPR*, 2023.

[32] B. Kim, M. Cartwright, and B. Pardo, "Vimsketch dataset," 2019. [Online]. Available: https://zenodo.org/records/2596911

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

[34] A. Gui, H. Gamper, S. Braun, and D. Emmanouilidou, "Adapting frechet audio distance for generative music evaluation," in *ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2024, pp. 1331–1335.

[35] S. Hershey, S. Chaudhuri, D. P. W. Ellis, J. F. Gemmeke, A. Jansen, C. Moore, M. Plakal, D. Platt, R. A. Saurous, B. Seybold, M. Slaney, R. Weiss, and K. Wilson, "Cnn architectures for large-scale audio classification," in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017. [Online]. Available: https://arxiv.org/abs/1609.09430

[36] Y. Wu*, K. Chen*, T. Zhang*, Y. Hui*, T. Berg-Kirkpatrick, and S. Dubnov, "Large-scale contrastive language-audio pretraining with feature fusion and keyword-to-caption augmentation," in *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP*, 2023.