Audio Engineering Society

Conference Paper 2

Presented at the 6th International Conference on Audio for Games
2024 April 27–29, Tokyo, Japan

A Machine learning method to evaluate and improve sound effects synthesis model design

Yisu Zong[1], Nelly Garcia-Sihuay[1], and Joshua Reiss[1]

[1] *Centre for Digital Music, Queen Mary University of London*

Correspondence should be addressed to Yisu Zong (`y.zong@qmul.ac.uk`)

ABSTRACT

Procedural audio models have great potential in sound effects production and design, they can be incredibly high quality and have high interactivity with the users. However, they also often have many free parameters that may not be specified just from an understanding of the phenomenon, making it very difficult for users to create the desired sound. Moreover, their potential and generalization ability are rarely explored fully due to their complexity. To address these problems, this work introduces a hybrid machine learning method to evaluate the overall sound matching performance of a real sound dataset. First, we train a parameter estimation network using synthesis sound samples. Through the differentiable implementation of the sound synthesis model, we use both parameter and spectral loss in this self-supervised stage. Then, we perform adversarial training by spectral loss plus adversarial loss using real sound samples. We evaluate our approach for an example of an explosion sound synthesis model. We experiment with different model designs and conduct a subjective listening test. We demonstrate that this is an effective method to evaluate the overall performance of a sound synthesis model, and its capability to speed up the sound model design process.

1 Introduction

The demand for immersive sound effects in games is increasing with the development of game content richness, and sound synthesis techniques are widely used in games' sound effects design and production processes. The most common techniques are sample-based, but their usability and variation are usually quite limited because it is too expensive to get the full range of a sound event, and its memory consumption is high. High-quality audio files can be data-heavy, and creating a diverse sound library for every possible player interaction and the game state is labor-intensive. This static approach can also be rigid, unable to adapt in real-time to the player's actions and choices.

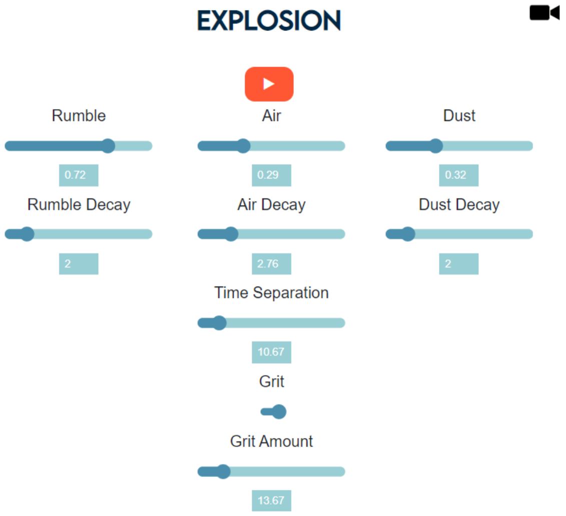
Procedural audio [1] offers a solution to these challenges by using algorithms to generate sounds in real-time based on player interactions and game states [2]. This dynamic method can not only reduce the need for large, pre-defined audio files but also provide a more adaptive and interactive gaming experience. A typical sound designing process goes through several analyses from different angles including waveform, spectrum and physical phenomenon, followed by the process of model parameterization [3]. This is generally challenging work because it requires a solid understanding of the physical principles of sound generation. Moreover, it is necessary to take into account the trade-off between sound realism and computational complexity of the model, which usually requires a lot of assumptions and simplification. In practice, sound designers may focus more on the sonic properties rather than ensuring the physical feasibility [4].

Based on these considerations, the inner structure of procedural audio models is generally complex, with many free parameters that may not be specified just from an understanding of the phenomenon, making it very difficult for users to create the desired sound. Also, procedural audio requires a large number of real-time parameter combinations in many different sound events [2], which poses an even more significant challenge for parameter selection. Moreover, a sound model's potential and generalization ability are rarely explored fully because of its complexity. Thus, the model design may be insufficient, which will lead to a worse synthesis quality than other methods [5, 6].

Therefore, the research on the exploration of sound models' overall performance and limitations can facilitate the improvement of model design and usage. One effective way is to conduct the sound matching experiment for the model with a dataset consisting of the desired sounds. A traditional way to do the sound matching is to use Evolutionary Algorithms (EA), including Evolution Strategies [7, 8], Genetic Algorithm [9, 10] or Particle Swarm Optimization [11, 12] to find some spectral representations' similarity of query and the target sound. EA is a problem-independent optimization method, it treats the model as a ”black-box” model and directly optimizes the parameter space. However, the computational cost of these methods is too expensive, so the requirements for a real-time query for users can rarely be satisfied.

Machine learning offers a promising and efficient approach for the sound matching task. After the model training, sound matching through the model could satisfy the requirement of a real-time query. In relevant studies [13, 14, 15], the typical approach is synthesizing a substantial amount of data and subsequently utilizing audio information to train a machine learning model for predicting the parameters of synthesized audio. The primary objective is to optimize this process by minimizing the discrepancy between predicted and ground-truth parameters.

Recently, the method Differentiable Digital Signal Processing (DDSP) [16] integrated classical DSP components into the deep learning workflow, enabling backpropagation of gradients in the audio domain. This idea is a significant source of inspiration for a lot of research and greatly enhances the applicability of deep learning techniques in sound matching [17, 18, 19].



Rumble 0.72

Air 0.29

Dust 0.32

Rumble Decay 2

Air Decay 2.76

Dust Decay 2

Time Separation 10.67

Grit

Grit Amount 13.67

Fig. 1: User control interface of the explosion sound synthesis model.

In this work, we conduct a machine learning method to perform the sound matching experiment for an explosion sound synthesis model and its variants. The explosion sound synthesis model and machine learning method are introduced in Section 2. Objective evaluation results are shown in Section 3, and results of the listening test are shown in Section 4. The discussion of results, limitations, and future research direction is provided in Section 5. Finally, conclusions are shown in Section 6.

2 Methods

2.1 Explosion Sound Synthesis

An explosion is a sudden and intense release of energy that leads to a rapid expansion of gases. The sound typically begins with a sharp, percussive shock wave, followed by a big blast and a descending rumble.

Previous work in explosion sound synthesis was mainly physical-based or sample-based methods. Due to the complexity of an explosion, physical-based synthesis

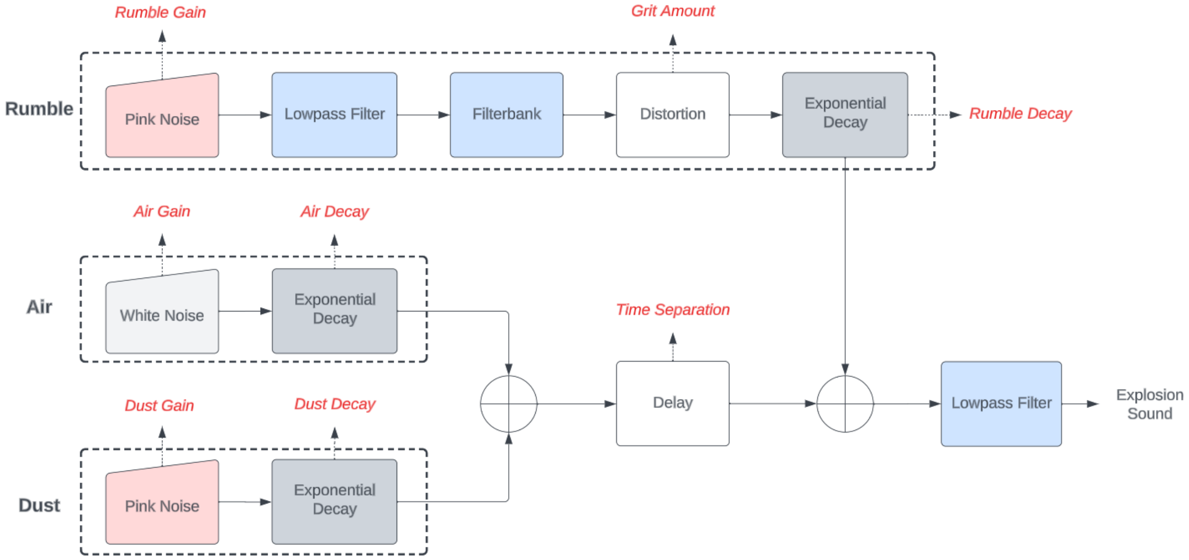


Fig. 2: High-level block diagram of the explosion sound synthesis model structure.

methods require expensive computation, and simplifying the physical model may lead to a degradation in sound quality [20]. Sample-based methods [21, 22] can serve as a valuable complement to physical models. However, they still encounter challenges related to heavy computation and limited variability.

In this work, the model we use is based on a physically-inspired synthesis method by Andy Farnell's design [3], and is available on the Nemisindo website[1]. It uses digital signal processing (DSP) components to approximate the physical phenomenon of explosion. The model is divided into three parts: "Rumble", "Air", and "Dust". All three parts start from a noise signal, where the Rumble and Dust use pink noise, and Air uses white noise, each with a user-controllable gain value. The explosion rumble mainly consists of low frequency components, so the Rumble is then connected to a low-pass filter where the cutoff frequency is 100 Hz, and a filter bank with five band-pass filters is used to shape the timbre. A distortion function is then applied to the Rumble,

$$y = \frac{(3+k)\cdot x \cdot 20 \cdot \frac{\pi}{180}}{\pi + k|x|} \tag{1}$$

where k is user-controllable to change the amount of distortion. The distortion enhances the granularity of the sound, aiming to simulate fragmented and shattered effects. An exponential decay envelope is then applied to Rumble, Air, and Dust respectively,

$$y(t) = Ae^{-\frac{t}{\tau}} \tag{2}$$

where A is a gain value, and τ is the user-controllable decay constant. For Rumble, the gain value is very high (A=3000) to achieve a dull thump effect at the start. Air and Dust are delayed to simulate the pressure wave effect, where the delay time is user-controllable. Finally, the three parts are put together and passed to a low-pass filter where the cutoff frequency is 10,000 Hz. Figure 1 shows the high-level structure of this model.

In our preliminary experiments, we observed that the model exhibited certain limitations, namely inadequate initial shock and the absence of high-frequency components in short bursts. We attributed these shortcomings to the narrow range of gain and decay constant values, and the limited coverage of explosion sound events by the filter design in Rumble. Based on the analysis, we experimented with the following settings of the controllable model parameters. First, we used the original setting of the synthesis model (Original) as the baseline. Second, we extended the parameter value range, and this model is denoted by Original-EV. The value range was extended from [0,1] to [0,5] for Rumble

[1] https://nemisindo.com/models/explosion

Gain, Air Gain, and Dust Gain. For Rumble Decay, Air Decay, and Dust Decay, the value range was extended from (0,10] to (0,20]. Third, based on Original-EV, we added one control band-pass filter (gain, center frequency, and Q factor as control parameters) in the filter bank of Rumble, and the cutoff frequency of the 100 Hz low-pass filter in Rumble was exposed in the range of [50, 500]. This model is denoted by Control-F. Fourth, based on Original-EV, we exposed the gains, center frequencies, and Q factors as control parameters of the five band-pass filters in the filter bank of Rumble, and the cutoff frequency of the 100 Hz low-pass filter in Rumble. This model is denoted by Control-FB. Table 1 summarizes these model configurations.

2.1.1 Differentiable Implementation

In our experiments, the machine learning pipeline requires a fully differentiable synthesis model allowing gradient backpropagation. Our model requires a differentiable biquad filter, where the transfer function is:

$$H(z) = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2}}{a_0 + a_1 z^{-1} + a_2 z^{-2}} \tag{3}$$

Direct implementation of an infinite impulse response (IIR) filter is possible, but the computation is expensive [23]. Based on implementation efficiency considerations, we adopted a finite impulse response (FIR) approximation method [24]. With an FIR filter, we can approximate the frequency response by evaluating $H(e^{j\omega})$ across a set of uniformly spaced frequencies at the frequency ω, thereby obtaining the output in the frequency domain, $Y = HX$, where $X = DFT(x)$. The time domain output is obtained through inverse DFT, $y = IDFT(Y)$.

2.2 Training Procedure

2.2.1 Data

We used two different kinds of sound data: synthesized sounds and real-world sounds.

For each version of the explosion model, we generated the sound with random parameter settings within the explosion model's range. All sounds are 3 seconds at the sample rate of 24,000 Hz. 30,000 data were generated and split into training, validation and test sets, each accounting for 80/10/10 respectively.

Then, we collected 72 high-quality real explosion sound samples from Pro Sound Effects[2] and BBC Sound Effects[3]. Our explosion model is designed to generate a single explosion, without including any environmental reflections or interactive effects such as glass shattering or impact by the explosion. Therefore, our data collection process adheres to the standard of avoiding obvious echoes and other interactive effects. All the samples were cut or zero-padded in the tail to 3 seconds at the sample rate of 24,000 Hz.

2.2.2 Training with Synthesized Sounds

The first step of our training procedure was to train a control parameter estimation network using synthesized sounds. This stage is depicted in Figure 3(a). We adopted a CNN-GRU architecture [16, 17] for the parameter estimation network. Three 1D convolution with normalization layers were employed for extracting deep embedding of the Mel spectrogram of the input sound data. Then, the output was fed into a 512-unit gated recurrent unit (GRU) layer, and subsequently, the GRU output was passed through a linear layer. As all control parameters are positive, we applied a sigmoid function to normalize the output to (0,1) and a linear function was employed to map the sigmoid output from (0,1) to the range of control parameters. The estimated parameters were then fed into the differentiable synthesis model for reconstruction.

The training objective in traditional deep learning sound matching was parameter loss [14, 15], and the differentiable implementation of the sound synthesis model allows gradient backpropagation of spectral loss. Recent work [17] has proposed that employing a mixed training strategy incorporating parameter and spectral loss could lead to better match quality. Therefore, we used both parameter loss and spectral loss in this step,

$$L_p = \sum_i |P_i - \hat{P}_i| \tag{4}$$

$$L_s = \sum_i (\|S_i - \hat{S}_i\|_1 + \|\log S_i - \log \hat{S}_i\|_1) \tag{5}$$

$$L = \alpha L_s + (1-\alpha) L_p \tag{6}$$

where parameter loss L_p is L1 loss of normalized control parameters, and the spectral loss L_s is a multi-resolution spectral loss [16], L is the total training objective. During the first 200 epochs, α was set to 0.5.

[2] https://www.prosoundeffects.com/hybrid-library/

[3] https://sound-effects.bbcrewind.co.uk/

Table 1: Summary of variations of the explosion synthesis model.

Model Type	Main Improvement	Number of Parameters
Original	None	8
Original-EV	Extended value range	8
Control-F	A control filter	12
Control-FB	Controllable filter bank	24

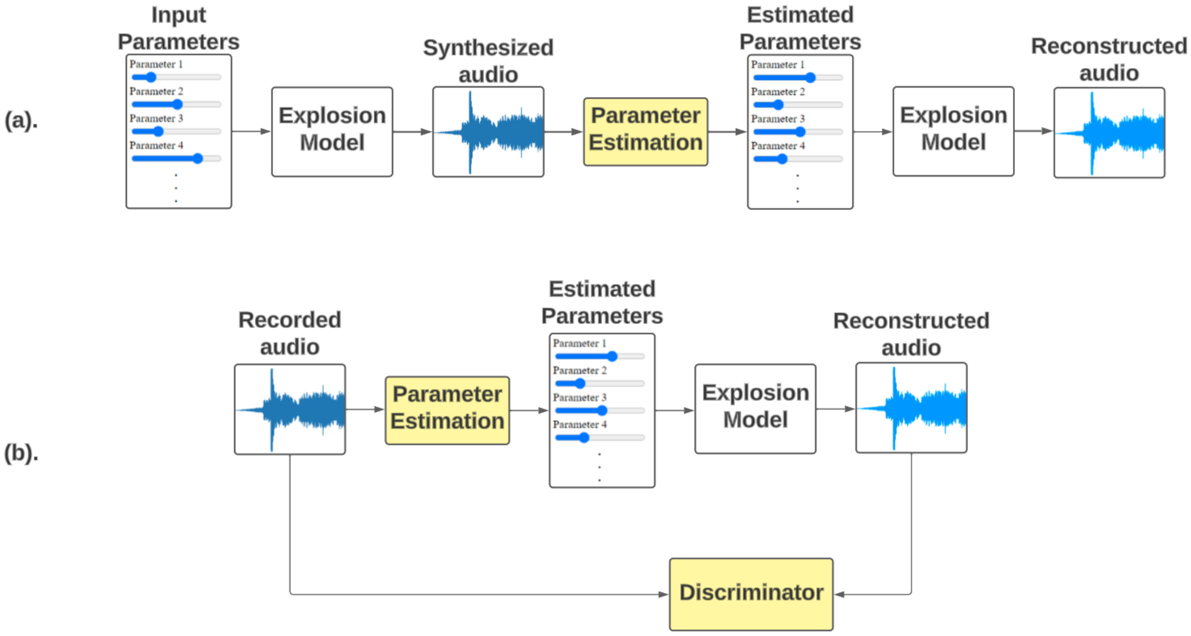


Fig. 3: Flow diagram of the training procedure.

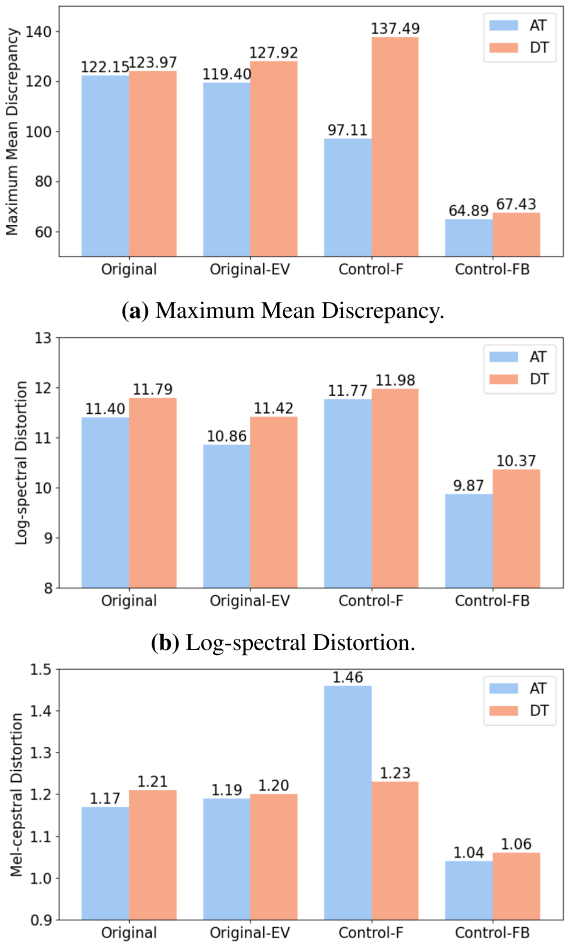
Subsequently, in the following 50 epochs, α linearly increased to 1 in order to complete the transition from a mixed loss to a spectral loss. Finally, the model continued training for an additional 50 epochs.

2.2.3 Adversarial Training

We aimed for the synthesised sound to be indistinguishable from the real sounds. However, there was a clear distribution gap between the audio space of the synthesized and the real datasets. To improve the realism of the matched output, we adopted the discriminator in Mel-GAN [25] for adversarial training—a multi-scale architecture containing three discriminators with different resolutions. As shown in Figure 3(b), the audio reconstruction process can be regarded as the generator of a GAN, while the goal of the discriminator is to identify whether the input is a reconstruction or a real-world sound. Following the method employed in Mel-GAN, the hinge version of discriminator loss in GAN is denoted by $L_{adv}(D;G)$. The hinge version of generator loss in GAN $L_{adv}(G;D)$ and a feature matching loss L_{FM} were incorporated into the generation training objective,

$$L = L_s + L_{adv}(G;D) + \lambda L_{FM} \tag{7}$$

where L_{FM} is L1 loss of distance between discriminator feature maps, and $\lambda = 10$. After training with synthesized data, the model was trained with the discriminator using real-world data for 100 epochs.



(a) Maximum Mean Discrepancy.

(b) Log-spectral Distortion.

(c) Mel-cepstral Distortion.

Fig. 4: Bar plots of objective evaluation results across different metrics. Blue bars represent adversarial training (AT), and red bars represent direct training using spectral loss (DT).

3 Objective Evaluation

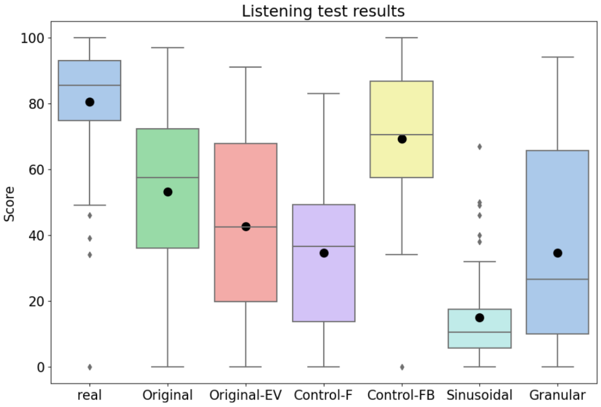


Fig. 5: Boxplot of subjective evaluation results. Black dots represent the average scores of each method.

For the evaluation metrics of overall sound matching quality, we used Maximum Mean Discrepancy (MMD) [26], a distribution distance measurement; log-spectral distortion (LSD), the root mean square log spectra distance; and mel-cepstral distortion (MCD) [27], the distance between Mel-frequency cepstral coefficients (MFCCs). We compared the reconstructions obtained by different model settings with the real sounds across the metrics above. Additionally, we conducted ablation studies to demonstrate the effectiveness of adversarial training. We compared the real sound training effectiveness of adversarial training with direct training using spectral loss across all model configurations. The results are shown in Figure 4.

Adversarial training shows superior overall match quality compared to solely employing spectral loss across various models and evaluation criteria, except the MCD score for the Control-F model. The Control-F model shows distinct characteristics of MMD and MCD compared to other models, potentially indicating the inconsistency between these evaluation metrics. Original-EV performs better than the Original model of MMD and LSD, and Control-FB outperforms all the other models across all the evaluation metrics. We observe that Control-F shows worse performance in LSD and MCD than Original-EV and even worse than the Original model, even though it introduces more control ability. This implies that the selection of appropriate control mode and control parameters is more crucial than just augmentation, as it not only enhances user control difficulty but also diminishes the synthetic quality of the model.

4 Subjective Evaluation

Following the methods in [6], we performed a listening test to compare 4 configurations of the explosion sound synthesis model with 3 real sounds. Besides, we incorporated two additional synthesis methods into the evaluation as benchmarks: Sinusoidal Modelling (Sinusoidal) [28] and Granular Synthesis (Granular) [29].

Table 2: P-values of pairwise comparison of synthesis methods using Tukey's HSD. The red numbers represent the null hypothesis has not been rejected ($p > 0.05$).

	Real	Original	Original-EV	Control-F	Control-FB	Sinusoidal	Granular
Real	-	<0.001	<0.001	<0.001	0.238	<0.001	<0.001
Original	<0.001	-	0.315	<0.01	<0.05	<0.001	<0.01
Original-EV	<0.001	0.315	-	0.642	<0.001	<0.001	0.633
Control-F	<0.001	<0.01	0.642	-	<0.001	<0.01	1.0
Control-FB	0.238	<0.05	<0.001	<0.001	-	<0.001	<0.001
Sinusoidal	<0.001	<0.001	<0.001	<0.01	<0.001	-	<0.01
Granular	<0.001	<0.01	0.633	1.0	<0.001	<0.01	-

Subjective evaluation similar to the MUlti Stimulus test with Hidden Reference and Anchor (MUSHRA) was conducted on the Go Listen platform [30] to evaluate the real sound sample and its reconstructions using six different methods. Participants rated the samples on a scale ranging from 0 to 100, where 1-20 is completely unrealistic, 20-40 is very unrealistic, 40-60 is somewhat unrealistic, 60-80 is good, and 80-100 is realistic. In total, 21 participants engaged in the online test, each providing 21 ratings. The participants were asked to use headphones or speakers in a quiet environment. Fourteen used headphones, and seven used speakers in a quiet environment. The participants' ages range from 21 to 63 years old, with an average of 29.3 years old. Of the total, eight were female, twelve were male, and one identified as other. Seventeen participants had a background related to music or audio production. Five of the participants did not rate any real audio samples above 80, so their ratings were removed.

The results are shown in Figure 5. We performed the one-way analysis of variance (ANOVA), showing significant differences among the synthesis methods ($p < 0.001$). To analyse the significance between the methods, we performed a post-hoc analysis using Tukey's Honest Significant Difference (Tukey's HSD), and the results are shown in Table 2. Control-FB outperformed all the other methods and was the only synthesis method for which the ratings did not differ significantly from the real samples. Control-F performed worse than the Original, which is the same result as the objective evaluation. The median and average scores of the Original-EV were lower than those of the Original but higher than those of the Control-F. However, there were no significant differences in ratings between the Original-EV, Original, and Control-F ($p > 0.05$).

5 Discussion

Control-FB has the best performance among all the models, and it can produce nearly indistinguishable explosion sounds from real ones. However, it also has the most parameters, thereby posing challenges in terms of model control ability when creating a sound from scratch. The parameters in the filter bank could be controlled by a two-dimensional interface, where users can draw a frequency curve or gain curve. Further investigation is required to assess the user-friendly method of control.

A limitation of this work is that the reconstruction quality is subject to the combined influence of both sound matching accuracy and expressive capacity of the synthesis model. Given the relatively straightforward structure of our model, the impact of sound matching accuracy is considerably less pronounced compared to the model's expressive capacity. However, this approach might become challenging when applied to other procedural models with more intricate structures.

Also, a precise objective metric is important to guide the design direction. Control-F evaluation results highlight the potential impact of incorrect parameterization on model quality, emphasizing the need for a precise metric to guide step-by-step design due to the substantial cost of conducting a series of listening tests. There is an evaluation difference within the selected objective metrics, and their correlation to subjective perception is not determined. Further perceptual quality metrics, such as [31, 32], can be evaluated.

6 Conclusion

We presented a machine learning method to evaluate the overall quality of different designs of an explosion sound synthesis model. We used synthesized data to train the parameter estimation network for the sound matching task, incorporating adversarial training using real audio samples to enhance the fidelity of synthesized sounds. The proposed explosion synthesis model demonstrates a significant enhancement and can generate authentic explosion sounds. This approach shows potential for expansion to applications of diverse procedural audio models.

References

[1] Menexopoulos, D., Pestana, P., and Reiss, J., "The State of the Art in Procedural Audio," *Journal of the Audio Engineering Society*, (12), pp. 826–848, 2023.

[2] Sinclair, J.-L., *Principles of game audio and sound design: sound design and audio implementation for interactive and immersive media*, CRC Press, 2020.

[3] Farnell, A., *Designing sound*, Mit Press, 2010.

[4] Moffat, D., Ronan, D., Reiss, J. D., et al., "Unsupervised taxonomy of sound effects," in *Proceedings of the 20th International Conference on Digital Audio Effects (DAFx-17)*, 2017.

[5] Böttcher, N., Serafin, S., et al., "Design and evaluation of physically inspired models of sound effects in computer games," in *Audio Engineering Society Conference: 35th International Conference: Audio for Games*, 2009.

[6] Moffat, D. and Reiss, J. D., "Perceptual evaluation of synthesized sound effects," *ACM Transactions on Applied Perception (TAP)*, 15(2), pp. 1–19, 2018.

[7] Mitchell, T. J. and Creasey, D. P., "Evolutionary sound matching: A test methodology and comparative study," in *Sixth International Conference on Machine Learning and Applications (ICMLA 2007)*, pp. 229–234, 2007.

[8] Mitchell, T., "Automated evolutionary synthesis matching," *Soft Computing*, 16(12), pp. 2057–2070, 2012.

[9] Riionheimo, J. and Välimäki, V., "Parameter estimation of a plucked string synthesis model using a genetic algorithm with perceptual fitness calculation," *EURASIP Journal on Advances in Signal Processing*, 2003(8), pp. 1–15, 2003.

[10] Lai, Y., Jeng, S.-K., Liu, D.-T., and Liu, Y.-C., "Automated optimization of parameters for FM sound synthesis with genetic algorithms," in *International Workshop on Computer Music and Audio Technology*, p. 205, 2006.

[11] Zúñiga, J. and Reiss, J. D., "Realistic procedural sound synthesis of bird song using particle swarm optimization," in *Audio Engineering Society Convention 147*, 2019.

[12] Cámara, M., Xu, Z., Zong, Y., Blanco, J. L., and Reiss, J. D., "Optimization Techniques for a Physical Model of Human Vocalisation," in *Proceedings of the 26th International Conference on Digital Audio Effects (DAFx23)*, 2023.

[13] Itoyama, K. and Okuno, H. G., "Parameter estimation of virtual musical instrument synthesizers," in *40th International Computer Music Conference (ICMC)*, 2014.

[14] Yee-King, M. J., Fedden, L., and d'Inverno, M., "Automatic programming of VST sound synthesizers using deep networks and other techniques," *IEEE Transactions on Emerging Topics in Computational Intelligence*, 2(2), pp. 150–159, 2018.

[15] Barkan, O. and Tsiris, D., "Deep synthesizer parameter estimation," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 3887–3891, 2019.

[16] Engel, J., Gu, C., Roberts, A., et al., "DDSP: Differentiable Digital Signal Processing," in *International Conference on Learning Representations*, 2019.

[17] Masuda, N. and Saito, D., "Synthesizer Sound Matching with Differentiable DSP." in *22nd International Society for Music Information Retrieval Conference (ISMIR)*, pp. 428–434, 2021.

[18] Ganis, F., Knudesn, E. F., Lyster, S. V., Otterbein, R., Südholt, D., Erkut, C., et al., "Real-time timbre transfer and sound synthesis using ddsp," in

Proceedings of the 18th Sound and Music Computing Conference, 2021.

[19] Clarke, S., Heravi, N., Rau, M., Gao, R., Wu, J., James, D., and Bohg, J., "DiffImpact: Differentiable Rendering and Identification of Impact Sounds," in *Conference on Robot Learning*, pp. 662–673, 2022.

[20] Dobashi, Y., Yamamoto, T., and Nishita, T., "Synthesizing sound from turbulent field using sound textures for interactive fluid simulation," in *Computer Graphics Forum*, volume 23, pp. 539–545, 2004.

[21] Liu, S. and Gao, S., "Automatic synthesis of explosion sound synchronized with animation," *Virtual Reality*, 24(3), pp. 469–481, 2020.

[22] Liu, S., Gao, S., and Xu, S., "Animating explosion with exploding sound and rigid-body sound," *Computer Animation and Virtual Worlds*, 32(1), p. e1970, 2021.

[23] Kuznetsov, B., Parker, J. D., and Esqueda, F., "Differentiable IIR filters for machine learning applications," in *Proceedings of the 23rd International Conference on Digital Audio Effects (DAFx2020)*, pp. 297–303, 2020.

[24] Nercessian, S., "Neural parametric equalizer matching using differentiable biquads," in *Proceedings of the 23rd International Conference on Digital Audio Effects (DAFx2020)*, pp. 265–272, 2020.

[25] Kumar, K., Kumar, R., De Boissiere, T., Gestin, L., Teoh, W. Z., Sotelo, J., De Brebisson, A., Bengio, Y., and Courville, A. C., "Melgan: Generative adversarial networks for conditional waveform synthesis," *Advances in neural information processing systems*, 32, 2019.

[26] Gretton, A., Borgwardt, K. M., Rasch, M. J., Schölkopf, B., and Smola, A., "A kernel two-sample test," *The Journal of Machine Learning Research*, 13(1), pp. 723–773, 2012.

[27] Kubichek, R., "Mel-cepstral distance measure for objective speech quality assessment," in *Proceedings of IEEE pacific rim conference on communications computers and signal processing*, volume 1, pp. 125–128, 1993.

[28] Serra, X. and Smith, J., "Spectral modeling synthesis: A sound analysis/synthesis system based on a deterministic plus stochastic decomposition," *Computer Music Journal*, 14(4), pp. 12–24, 1990.

[29] O'Leary, S. and Röbel, A., "A montage approach to sound texture synthesis," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 24(6), pp. 1094–1105, 2016.

[30] Barry, D., Zhang, Q., Sun, P. W., and Hines, A., "Go listen: an end-to-end online listening test platform," *Journal of Open Research Software*, 9(1), 2021.

[31] Thiede, T., Treurniet, W. C., Bitto, R., Schmidmer, C., Sporer, T., Beerends, J. G., and Colomes, C., "PEAQ-The ITU standard for objective measurement of perceived audio quality," *Journal of the Audio Engineering Society*, 48(1/2), pp. 3–29, 2000.

[32] Chinen, M., Lim, F. S., Skoglund, J., Gureev, N., O'Gorman, F., and Hines, A., "ViSQOL v3: An Open Source Production Ready Objective Speech and Audio Metric," in *2020 Twelfth International Conference on Quality of Multimedia Experience (QoMEX)*, 2020.