# Learning Audio Attributes in 3D Gaussian Splatting

Tsung-Wei Huang, Guan-Ming Su
Dolby Laboratories, Inc., USA
tsung-wei.huang@dolby.com, guanmingsu@ieee.org

## ABSTRACT

3D Gaussian Splatting (3DGS) is an explicit 3D scene representation for novel view synthesis. While being computationally efficient, it can render high-quality and realistic images. To enable the 3DGS model for more immersive applications in virtual environments, in this paper, we propose augmenting the Gaussian primitives with audio attributes for sound generation. In the proposed framework, we first learn to predict audio signals from DINOv2 visual features through a low-dimensional embedding space. Then, given multi-view images and visual feature maps of a 3D object, we train the 3DGS model jointly with its audio attribute in the embedding space to ensure multi-view consistency in visual appearance and audio property. The learned 3DGS model with audio attributes can render both novel view images and generate audio signals in real-time. Experimental results on the ObjectFolder-Real dataset show the proposed framework can generate realistic audio signals of impact sounds of real-world household objects.

***Index Terms***— Gaussian splatting, audio attribute

## 1. INTRODUCTION

Recently, the 3D Gaussian Splatting (3DGS) [1] has gained increasing attention owing to its high efficiency and quality in novel view synthesis for 3D scenes [10]. Compared to implicit neural representations such as Neural Radiance Field (NeRF) [2] or Instant Neural Graphics Primitives (InstantNGP) [3] whose performance and quality are limited by the ray marching and choice of grid, 3DGS represents a 3D scene explicitly using GPU-friendly 3D Gaussians and thus achieves faster training and rendering speed as well as better image quality.

In view of the great success of 3DGS for novel view synthesis for 3D scenes and its potential in virtual environment applications, in this paper, we propose adding additional *audio attributes* to the 3DGS model for more user engagement and interactivities with a 3D scene in virtual environment. The additional audio attributes are aimed at providing audio signals to the user in real-time interaction with a 3D scene. As shown in Figure 1, when the user touches or interacts with a 3D object in virtual environment, the audio attributes in the corresponding 3D Gaussian can be retrieved and decoded into audio signals such as impact sounds of the object.

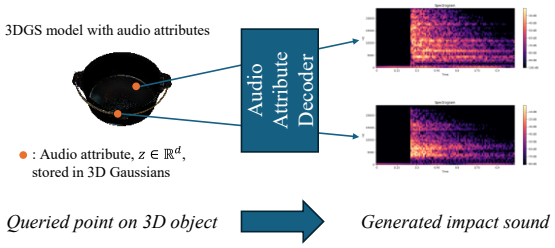



Figure 1. We propose augmenting 3DGS model with audio attributes. Given any queried points on the 3D object by user, the audio attribute decoder decodes the corresponding audio attributes in 3D Gaussians into audio signals such as impact sounds.

While we can learn the visual appearance for a 3D object using its multi-view captured images, learning the audio attributes for a 3D object is not trivial because densely recording the impact sounds for each surface point on an object is extremely expensive and need to be done in well-controlled environment. Therefore, we propose an efficient framework to learn the audio attributes of any 3D objects and generate the impact sounds in a zero-shot manner while rendering novel view images in real-time.

The high-level workflow of the proposed framework is shown in Figure 2. In the V2A training stage, the goal is to learn a low-dimensional embedding space where audio attributes are projected from visual features from pretrained vision foundation models, such as CLIP [5] or DINOv2 [6], and decoded into audio signals. The assumption is that audio signals such as impact sounds of a 3D object are correlated with its visual features. In the 3DGS-A training stage, given multi-view input images and corresponding visual feature maps of a 3D object, we can train a 3DGS model with multi-view consistent audio attributes. Then, the learned audio attributes can be decoded into impact sounds given any queried 3D points on the 3DGS model.

979-8-3315-7111-5/26/$31.00 ©2026 IEEE

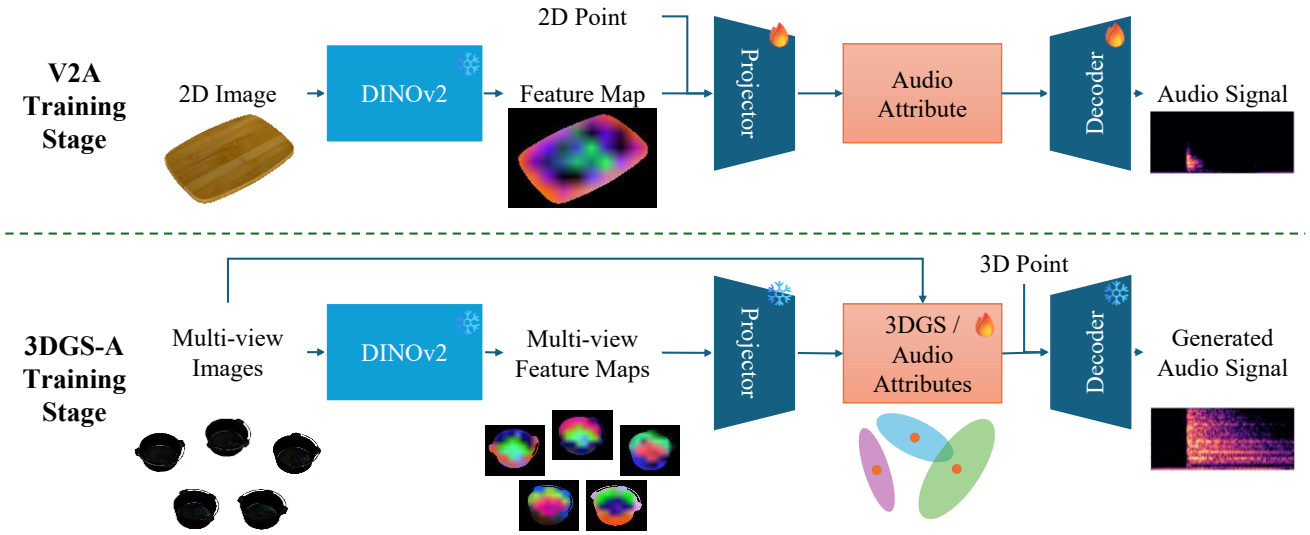



Figure 2. Proposed framework for learning audio attributes and generating audio signals in 3DGS models. In the V2A training stage, we train the projector and decoder to predict audio attribute from 2D image and decode it into audio signal. In the 3DGS training stage, we freeze the projector and decoder and optimize the 3DGS model jointly with audio attributes to generate audio signal in zero-shot manner.

The major contributions of this work are summarized as follows:

- We propose a framework to embed audio attribute into 3D Gaussians for providing more immersive experience for users in virtual environment.
- We propose the audio attribute projector and decoder to project pretrained visual feature into low-dimensional audio attribute embedding space, and decode the audio attribute into audio signal, such as impact sound.
- We propose the method to jointly train both visual and audio attributes of any 3D objects and generate the impact sounds given user-queried 3D points on objects in a zero-shot manner.
- Experimental results show the proposed framework can generate realistic audio signals of impact sounds of 3D objects when rendering in real-time.

The rest of the paper is organized as follows. Section 2 reviews the preliminaries of the 3DGS and defines the proposed audio attributes. Section 3 presents the proposed audio attribute training and testing framework. Section 4 shows the experimental results, and Section 5 concludes this paper.

## 2. BACKGROUND

### 2.1. 3D Gaussian Splatting

In this section, we briefly review the preliminaries of 3DGS [1]. The 3DGS represents a 3D scene as a set of 3D Gaussians $\mathcal{G} = \{G_k = (\mu_k, \Sigma_k, o_k, S_k) | k = 1, \dots, K\}$, where $K$ is the number of Gaussians, and each 3D Gaussian $G$ has the following properties: mean $\mu$, covariance $\Sigma$, opacity $o$, spherical harmonic (SH) coefficients $S$ for color. The 3D Gaussian at a 3D location $x$ can be represented as:

$$G(x) = \exp\left(-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)\right) \quad (1)$$

To render the 3D Gaussians into 2D image, each 3D Gaussians $G$ is first projected to 2D Gaussian $G^{2D}$ on the 2D image plane [4]. Then, a tile-based rasterizer is used to sort and render the 2D Gaussians by $\alpha$-blending. For a 2D pixel $x'$, the rendered image $\mathcal{I}$ is calculated as:

$$\mathcal{I}(x') = \sum_{i \in \mathcal{N}} c_i \alpha_i \prod_{j=1}^{i-1}(1-\alpha_j) \quad (2)$$

$c_i$ is the color of the $i$-th sorted Gaussian derived from SH coefficients, $\mathcal{N}$ is the set of Gaussians in the tile, and $\alpha_i = o_i G_i^{2D}(x')$ is the $\alpha$-blending weight.

Given a set of training images with known camera poses, the 3D Gaussians can be optimized using gradient descent methods. Commonly used loss functions include $\mathcal{L}_1$ loss and SSIM loss between rendered 2D images and ground truth images. Besides, the 3D Gaussians can be initialized using Structure-from-Motion (SfM) sparse point cloud and the number of 3D Gaussians are managed by adaptive density control (growing and pruning) during optimization.

### 2.2 Audio Attribute in 3D Gaussian

We define the additional audio attribute of each 3D Gaussian $G$ as a vector $z \in \mathbb{R}^d$, where $d$ is the dimension of the vector. The 3D Gaussians with additional attributes can be denoted as $\mathcal{G}^A = \{G_k^A = (\mu_k, \Sigma_k, o_k, S_k, z_k) | k = 1, \dots, K\}$. The concept of rendering and decoding 3D Gaussians with audio attributes into image and audio is illustrated in Figure 3.

In addition, unlike visual attributes which are rendered into images, audio attributes do not need pixel level spatial resolution. Therefore, in real-world applications, to reduce computational cost and storage, we only need to keep the audio attributes in a subset of $M$ Gaussians where $M \ll K$, and remove the audio attributes in the remaining Gaussians. The subset to keep audio attributes can be determined by farthest point sampling (FPS) or other point cloud sampling methods.

## 3. METHOD

In this section, we introduce the proposed framework in two stages: **V2A Training Stage** – learning to predict audio attributes from visual features; and **3DGS-A Training Stage** – optimizing audio attributes for unseen objects.

### 3.1. V2A Training Stage - Learning to Predict Audio Attributes from Visual Features

In V2A training stage, we train a projector to project visual features into audio attributes in a low-dimensional space and a decoder to decode the audio attributes into audio signals.

*3.1.1 Projector*

The audio attribute projector $h: \mathbb{R}^{d_F} \to \mathbb{R}^d$ projects the visual feature given by pretrained vision foundation model $\Psi$ such as DINOv2 [6]. For a 2D image $\mathcal{I} \in \mathbb{R}^{3 \times H \times W}$, the foundation model $\Psi$ gives a visual feature map $\mathcal{F} \in \mathbb{R}^{d_F \times H_F \times W_F}$, where $d_F$ is the dimension of visual feature. At a 2D pixel location $x'$, the projector takes the bilinearly sampled visual feature $F = \mathcal{F}(x')$ from feature map and outputs a low dimensional latent embedding as the audio attribute $z \in \mathbb{R}^d$, where $d \ll d_F$. For simplicity, we can write the operation as:

$$z = h(F) \tag{3}$$

*3.1.2 Decoder*

The audio attribute decoder $g: \mathbb{R}^d \to \mathbb{R}^L$ decodes each audio attribute $z \in \mathbb{R}^d$ into an audio signal $y \in \mathbb{R}^L$, where $L$ is the number of samples of the audio waveform. For simplicity, we can write the decoder operation as:

$$y = g(z) \tag{4}$$

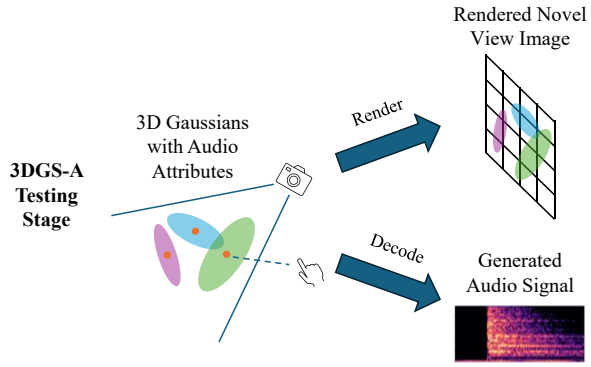



Figure 3. Given 3D Gaussians with audio attributes, we can render their visual attributes into image at given camera pose and decode their audio attributes into audio signal at queried 3D point (e.g., by hand touch).

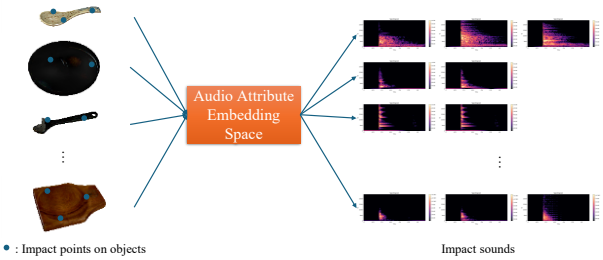



Figure 4. We train the audio attribute embedding space on objects of various categories and materials with point-impact sound pairs.

*3.1.3 Training*

Given the training dataset consisting of sparsely annotated pairs of points and audio signal $\{(x'_m, y^*_m) | m = 1, \dots, M\}$ where $M$ is the number of training data, we first use the feature extractor to extract the semantic feature $F_m$ from corresponding input image at each point location $x'_m$ to acquire the processed training data $\{(F_m, y^*_m) | m = 1, \dots, M\}$. Then, we can train the projector $h$ and decoder $g$ in an end-to-end manner using gradient descent methods to minimize the loss $\mathcal{L}_{V2A}$:

$$\begin{aligned} \mathcal{L}_{V2A} &= \frac{1}{M} \sum_{m=1}^{M} \mathcal{L}_{audio}(y_m, y^*_m) \\ &= \frac{1}{M} \sum_{m=1}^{M} \mathcal{L}_{audio}\big(g\big(h(F_m)\big), y^*_m\big) \end{aligned} \tag{5}$$

Here $\mathcal{L}_{audio}(y_m, y^*_m)$ is the loss function for audio signal prediction. The idea is illustrated in Figure 4.

### 3.2. 3DGS-A Training Stage - Optimizing Audio Attributes for Unseen Objects

After V2A training, given an unseen 3D object with $N$ multi-view images $\{\mathcal{I}^*_v | v = 1, \dots, N\}$ as input, we first encode the images into multi-view feature maps and then project into

audio attribute maps $\{z_v^* = h(\Psi(\mathcal{I}_v^*))|v = 1, \dots, N\}$ as the ground truth to train audio attributes. Similar to the rendering process for color images in original 3DGS models, we can render the audio attribute maps $z$ from the audio attributes of all 3D Gaussians by replacing the color $c_i$ with audio attributes $z_i$ in Eq. (2), similar to LangSplat [7].

$$z(x') = \sum_{i \in \mathcal{N}} z_i \alpha_i \prod_{j=1}^{i-1} (1 - \alpha_j) \tag{6}$$

Therefore, we can train the 3DGS model with audio attributes using the image-based rendering loss $\mathcal{L}_{image}$ of both rendered images and audio attribute maps.

$$\mathcal{L}_{3DGS} = \frac{1}{N} \sum_{v=1}^{N} \mathcal{L}_{image}(\mathcal{I}_v, \mathcal{I}_v^*) \tag{7}$$

$$\mathcal{L}_{3DGS-A} = \frac{1}{N} \sum_{v=1}^{N} \mathcal{L}_{image}(z_v, z_v^*) \tag{8}$$

Because the audio attribute maps are inferred from input images in each view individually and are not intrinsically multi-view consistent, we first train the visual attributes and freeze them before we train the audio attributes.

At testing time, given the queried 3D point $x_{query}$, we retrieve the nearest 3D Gaussian and decode its audio attribute into final output audio signal:

$$y = g\left(z_{\arg\min_k \|\mu_k - x_{query}\|}\right) \tag{9}$$

## 4. EXPERIMENTAL RESULTS

### 4.1 Implementation Details

We use the vision foundation model DINOv2-Large (ViT-L/14) to extract visual features. For the projector, we use a three-layer multilayer perceptron (MLP) to project the 1024-dimensional DINOv2 feature to 75-dimensional audio attributes. For the decoder, we use the audio decoder in RAVE [8] which directly outputs audio waveform.

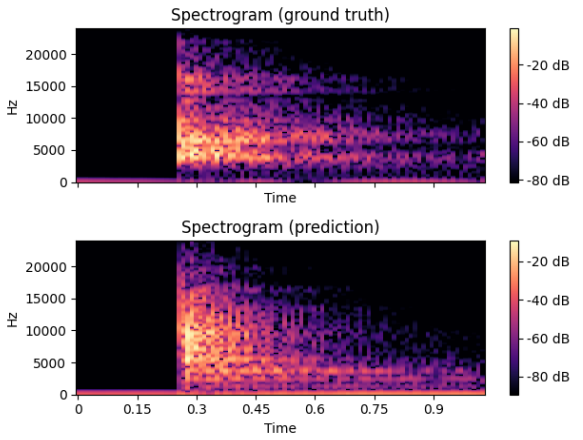
The loss function for training projector and decoder is a combination of L1 loss and multi-scale short-time Fourier transform (MS-STFT) spectrogram loss [11]:

$$\mathcal{L}_{audio} = (1 - \lambda_{MS-STFT})\mathcal{L}_{L1} + \lambda_{MS-STFT}\mathcal{L}_{MS-STFT} \tag{10}$$

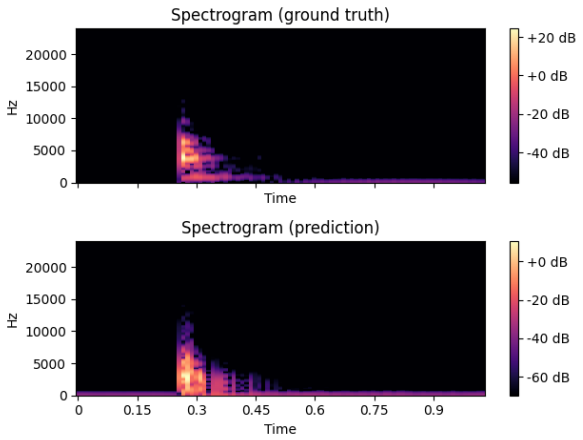
The loss function for training 3DGS with audio attributes is a combination of L1 loss and SSIM loss:

$$\mathcal{L}_{image} = (1 - \lambda_{SSIM})\mathcal{L}_{L1} + \lambda_{SSIM}\mathcal{L}_{SSIM} \tag{11}$$

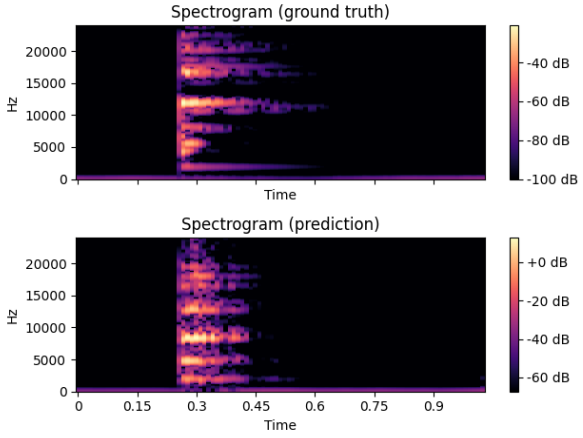
We use $\lambda_{MS-STFT} = 0.1$ and $\lambda_{SSIM} = 0.5$ in all our experiments.




(a) Dutch Oven, Iron




(b) Cutting Board Middle, Wood




(c) Beer Glass, Glass

Figure 5. Example STFT spectrograms of generated audio waveform of different objects and materials. For each object, top is spectrogram of ground truth waveform and bottom is spectrogram of predicted waveform.

### 4.2 Dataset

We use the ObjectFolder-Real dataset [9] to evaluate our method. The ObjectFolder-Real dataset contains 100 real-world household objects with high-quality 3D meshes. Each object has 30-50 impact sounds recorded with corresponding 3D impact locations. We split the dataset into 90 objects for training and 10 objects for testing. Besides, to simulate the real-world multi-view capture scenario, for each object, we use the mesh to synthetically generate 100 multi-view images to train the projector, decoder and 3DGS models.

### 4.3 Audio Generation Result

We evaluate the audio generation quality by comparing the STFT spectrograms of generated audio waveform with ground truth waveform. Example results in Figure 5 show inter-object variation of predicted STFT spectrograms in the testing set. Besides, we can see that the generated audios share the same trend as ground truth in distribution in time and frequency. For quantitative evaluation, we measure the L1 distance between predicted and ground truth STFT spectrograms. Because the magnitude of impact sound depends on the strength of impact, which is unknown in our framework, we scale the predicted spectrogram so that it has the same average magnitude as ground truth spectrogram before calculating L1 distance. For the testing set, the L1 distance of scaled spectrogram is 5.36dB.

The details of each object and the L1 distance are shown in Table 1. We can see that for objects whose material and type can be easily infer from the RGB image, such as (30, Cutting Board Middle, Wood) and (50, Mixing Bowl Small, Plastic), the predicted spectrograms are more accurate. On the other hand, for objects that have very distinct texture or shape, like (60, Beer Glass, Glass) and (80, Spoon Holder, Wood), the error of prediction is much higher. We think this is because the prediction of audio attributes is distracted by the unrelated visual features. We believe this issue can be alleviated by increasing the size and diversity of training dataset and applying data augmentation on RGB images.

Table 1. Objects in Testing Set

| Index<br>Name<br>Material | RGB Image | L1 dist. (dB) |
|---|---|---|
| 10<br>Soup Bowl<br>Ceramic |  | 8.15 |
| 20<br>Dutch Oven<br>Iron |  | 8.83 |
| 30<br>Cutting Board<br>Middle<br>Wood |  | 2.05 |
| 40<br>Wrench Middle<br>Steel |  | 3.09 |
| 50<br>Mixing Bowl<br>Small<br>Plastic |  | 1.47 |
| 60<br>Beer Glass<br>Glass |  | 7.05 |
| 70<br>Drop Funnel<br>Polycarbonate |  | 2.73 |
| 80<br>Spoon Holder<br>Wood |  | 9.78 |
| 90<br>Slotted Turner<br>Steel |  | 5.24 |
| 100<br>Trim Removal<br>Tool 2<br>Plastic |  | 5.15 |

## 5. CONCLUSION

In this paper, we propose a framework to generate audio attributes in 3DGS model and predict the audio signals created by the 3D object. A projector projects visual features from pretrained vision foundation models into low-dimensional audio attributes; and a decoder decodes the audio attributes into audio signals. With multi-view supervision, the audio attributes in 3DGS model and corresponding audio signals are 3D consistent. Experimental results show the proposed framework can generate realistic audio signals of impact sounds of real-world household objects. We believe this work can provide a better and more immersive experience to users in the virtual environment.

## 6. REFERENCES

[1] Kerbl, Bernhard, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. "3D Gaussian Splatting for Real-Time Radiance Field Rendering." ACM Trans. Graph. 42, no. 4 (2023): 139-1.

[2] Mildenhall, Ben, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. "Nerf: Representing scenes as neural radiance fields for view synthesis." Communications of the ACM 65, no. 1 (2021): 99-106.

[3] Müller, Thomas, Alex Evans, Christoph Schied, and Alexander Keller. "Instant neural graphics primitives with a multiresolution hash encoding." ACM transactions on graphics (TOG) 41, no. 4 (2022): 1-15.

[4] Zwicker, Matthias, Hanspeter Pfister, Jeroen Van Baar, and Markus Gross. "EWA volume splatting." In Proceedings Visualization, 2001. VIS'01., pp. 29-538. IEEE, 2001.

[5] Radford, Alec, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry et al. "Learning transferable visual models from natural language supervision." In International conference on machine learning, pp. 8748-8763. PMLR, 2021.

[6] Caron, Mathilde, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. "Emerging properties in self-supervised vision transformers." In Proceedings of the IEEE/CVF international conference on computer vision, pp. 9650-9660. 2021.

[7] Qin, Minghan, Wanhua Li, Jiawei Zhou, Haoqian Wang, and Hanspeter Pfister. "Langsplat: 3d language gaussian splatting." In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 20051-20060. 2024.

[8] Caillon, Antoine, and Philippe Esling. "RAVE: A variational autoencoder for fast and high-quality neural audio synthesis." arXiv preprint arXiv:2111.05011 (2021).

[9] Gao, Ruohan, Yiming Dou, Hao Li, Tanmay Agarwal, Jeannette Bohg, Yunzhu Li, Li Fei-Fei, and Jiajun Wu. "The objectfolder benchmark: Multisensory learning with neural and real objects." In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 17276-17286. 2023.

[10] Kathariya, Birendra, Dae Yeol Lee, Tsung-Wei Huang, Tong Shao, Peng Yin, and Guan-Ming Su. "Gaussian Splatting: State-of-The-Arts and Future Trends." In 2025 International Conference on Computing, Networking and Communications (ICNC), pp. 876-881. IEEE, 2025.

[11] Engel, Jesse, Lamtharn Hantrakul, Chenjie Gu, and Adam Roberts. "DDSP: Differentiable digital signal processing." arXiv preprint arXiv:2001.04643 (2020).