Audio Engineering Society

Conference Paper

Presented at the 2022 International Conference on
Audio for Virtual and Augmented Reality
2022 August 15–17, Redmond, WA, USA

A Variational Y-Autoencoder for Disentangling Gesture and Material of Interaction Sounds

Simon Schwär[1], Meinard Müller[1], and Sebastian J. Schlecht[2]

[1] *International Audio Laboratories, Erlangen, Germany*
[2] *Dept. of Signal Processing and Acoustics and Dept. of Art and Media, Aalto University, Espoo, Finland*

Correspondence should be addressed to Simon Schwär (simon.schwaer@audiolabs-erlangen.de)

ABSTRACT

Appropriate sound effects are an important aspect of immersive virtual experiences. Particularly in mixed reality scenarios it may be desirable to change the acoustic properties of a naturally occurring interaction sound (e.g., the sound of a metal spoon scraping a wooden bowl) to a sound matching the characteristics of the corresponding interaction in the virtual environment (e.g., using wooden tools in a porcelain bowl). In this paper, we adapt the concept of a Y-Autoencoder (YAE) to the domain of sound effect analysis and synthesis. The YAE model makes it possible to disentangle the gesture and material properties of sound effects with a weakly supervised training strategy where only an identifier label for the material in each training example is given. We show that such a model makes it possible to resynthesize sound effects after exchanging the material label of an encoded example and obtain perceptually meaningful synthesis results with relatively low computational effort. By introducing a variational regularization for the encoded gesture, as well as an adversarial loss, we can further use the model to generate new and varying sound effects with the material characteristics of the training data, while the analyzed audio signal can originate from interactions with unknown materials.

1 Introduction

One goal of audio analysis and synthesis tasks is to identify and allow control of interpretable underlying factors of variation in the analyzed sound before it is resynthesized. A prominent example of this *disentanglement* task is timbre transfer, where some factors of variation (the instrument) are modified, while others (pitch, expression, etc.) should remain constant. In this paper, we combine methods for learning a disentangled representation and adapt them to a particular use case similar to timbre transfer: modifying the acoustic properties of an object (the *material*) while the interaction itself, i.e., the strength, duration, etc. of the impacts that produced the sound (the *gesture*) remains constant. This can be particularly useful in mixed reality scenarios, where an interaction may be performed with a real object that is replaced by a virtual representation of the object with different characteristics.

As an example, Figure 1 shows how a user may use a metal spoon to interact with a wooden bowl.

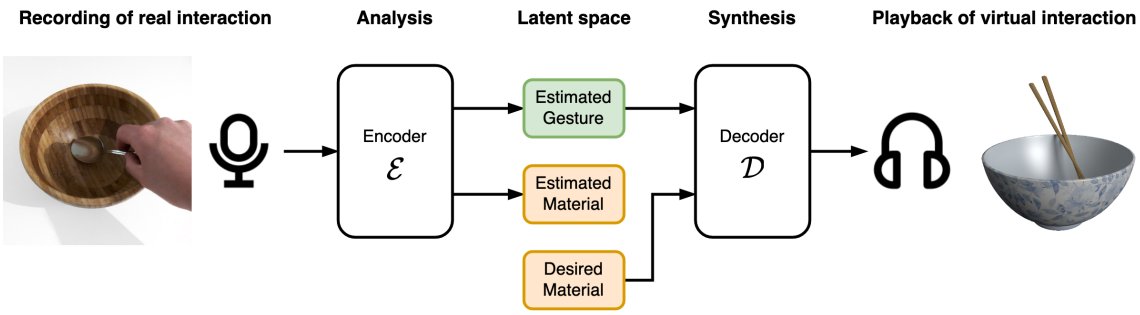

Fig. 1: Overview of the proposed system: An audio signal is recorded and analyzed by the encoder $\mathcal{E}$. The disentangled latent representation consisting of an estimate for gesture and material can be modified (e.g. by changing the material). A signal is resynthesized using the decoder $\mathcal{D}$ and played back over headphones.

The sound of this real interaction is analyzed, split up into a part that represents the material characteristics and a part that represents the gesture (i.e., how the bowl was struck or scraped). Before resynthesizing, the material representation is replaced to better match the characteristics of the virtual object – for example, chopsticks in a porcelain bowl. The resynthesized sound is then played back to the user, contributing to the immersive experience that the interaction actually took place with the virtual bowl.

To achieve this disentangling analysis and synthesis, we adapt the Y-Autoencoder [1] neural network training strategy to learn a representation of short audio clips where gesture and material are separated and can be individually modified. Using a single class label for each material (e.g., different combinations of spoons and bowls), we can train a model with weakly labeled training data. As opposed to other generative methods that achieve a similar disentanglement, like conditional generative adversarial networks (GANs) [2], an autoencoder (AE) can learn to analyze the disentangled representation as well. By adding a variational constraint on the learned gesture representation, we show that the gesture can also be encoded from recordings of interactions with objects that have not been part of the training set of materials. Furthermore, we add an adversarial loss to improve the resynthesis after changing the material class of an encoded example. With this, we combine advantages of generative methods and a conditional autoencoder in a single model, while achieving reasonable audio quality with small models whose computational complexity allows real-time execution.

This paper is organized as follows. In Section 2, we introduce our notation, review the concept of the Y-Autoencoder and discuss related work in the field of sound effect synthesis with neural networks. Section 3 gives an overview of our model and training strategy. In Section 4, we introduce our datasets and the experimental setups. Results are presented and discussed in Section 5. We provide audio examples and code on a supplemental website[1].

2 Background

An autoencoder (AE) consists of an encoder $\mathcal{E} : \mathcal{X} \to \mathbb{R}^L$ and a decoder $\mathcal{D} : \mathbb{R}^L \to \mathcal{X}$, so that for any $x \in \mathcal{X}$, $x \approx \mathcal{D}\big(\mathcal{E}(x)\big)$, where $\mathcal{X}$ is the set of possible inputs (e.g., images or audio frames). The vector $z = \mathcal{E}(x)$ is called an L-dimensional *latent representation* of x, where L is typically much smaller than the dimensionality of $\mathcal{X}$. The encoding and decoding can be learned when $\mathcal{E}$ and $\mathcal{D}$ are neural networks and the reconstruction loss function

$$\mathcal{L}_{\mathrm{R}}(x) = \big\| x - \mathcal{D}\big(\mathcal{E}(x)\big) \big\|_2^2 \tag{1}$$

is minimized on a training set $\mathcal{X}_{\mathrm{T}} \subset \mathcal{X}$.

[1] https://audiolabs-erlangen.de/resources/2022-AVAR-InteractionSounds

A variant of this method is the *variational* autoencoder (VAE) [3], where the encoder outputs parameters μ and σ of a multivariate normal distribution with independent dimensions from which z is sampled. The objective is extended to learn a continuous and compact latent space by regularizing the encoder output distribution for any input to be close to $\mathcal{N}(\mathbb{0}, \mathbb{I})$, with $\mathbb{0}$ as the all-zero vector and $\mathbb{I}$ as the identity matrix. This is achieved by minimizing the Kullback-Leibler (KL) divergence $\mathrm{KL}\big(\cdot \,\|\, \cdot\big)$ between the distributions, yielding a loss function with two terms:

$$\begin{aligned}\mathcal{L}_{\mathrm{VAE}} &= \mathcal{L}_{\mathrm{R}} + \mathrm{KL}\big(\mathcal{N}(\mu, \mathrm{diag}(\sigma)) \,\|\, \mathcal{N}(\mathbb{0}, \mathbb{I})\big) \\ &= \mathcal{L}_{\mathrm{R}} + \frac{1}{2}\big(1 + \log(\sigma^2) - \sigma^2 - \mu^2\big).\end{aligned} \tag{2}$$

With the constraint on the latent space organization, the VAE learns a representation where the distance between two encodings reflects the similarity of the original data points.

Adversarial losses have been used successfully to improve audio quality of generative models [4] and autoencoders [5]. The loss involves a separate *discriminator* neural network $\mathcal{A} : \mathcal{X} \to \{0, 1\}$ that is trained to discriminate a real input $x \in \mathcal{X}_{\mathrm{T}}$ from an output $\hat{x} = \mathcal{E}\big(\mathcal{D}(x)\big)$. The gradient of the loss to maximize the chance of correctly identifying x and $\hat{x}$ is used to train $\mathcal{A}$, whereas the gradient of the loss to maximize the chance of misclassifying $\hat{x}$ as real is used to train $\mathcal{E}$ and $\mathcal{D}$.

2.1 Y-Autoencoder

The Y-Autoencoder (YAE) [1] introduces a partly supervised method to disentangle a known class label $y \in \{0, 1\}^C$ (a *one-hot* encoding with C classes) from the rest of the latent representation. We denote the class estimate as the *explicit part* $z_{\mathrm{e}} \in [0, 1]^C$ and the remaining latent representation as the implicit part $z_{\mathrm{i}} \in \mathbb{R}^L$. The encoder and decoder of the YAE process z_{e} and z_{i} separately, so that $\mathcal{E}_{\mathrm{Y}} : \mathcal{X} \to \mathbb{R}^C \times \mathbb{R}^L$ with $(z_{\mathrm{e}}, z_{\mathrm{i}}) = \mathcal{E}_{\mathrm{Y}}(x)$, and $\mathcal{D}_{\mathrm{Y}} : \mathbb{R}^C \times \mathbb{R}^L \to \mathcal{X}$ with $x \approx \mathcal{D}_{\mathrm{Y}}(z_{\mathrm{e}}, z_{\mathrm{i}})$.

To learn the disentanglement, i.e., the independence of z_{e} and z_{i}, the latent representation is decoded twice with $\hat{x} = \mathcal{D}_{\mathrm{Y}}(y, z_{\mathrm{i}})$ and $\tilde{x} = \mathcal{D}_{\mathrm{Y}}(y_{\mathrm{r}}, z_{\mathrm{i}})$, where $y_{\mathrm{r}} \in \mathbb{R}^C$ is the one-hot label of a random class. The YAE loss

$$\mathcal{L}_{\mathrm{YAE}} = \mathcal{L}_{\mathrm{R}} + \mathcal{L}_{\mathrm{L}} + \mathcal{L}_{\mathrm{E}} + \mathcal{L}_{\mathrm{I}} \tag{3}$$

consists of four terms (nomenclature adapted by us):

1. A reconstruction loss $\mathcal{L}_{\mathrm{R}}$ as in (1).
2. The label classification loss $\mathcal{L}_{\mathrm{L}} = \mathrm{CE}(z_{\mathrm{e}}, y)$, where the cross-entropy loss $\mathrm{CE}(a, b) = \sum_{j=1}^{C} a_j \log b_j$ between z_e and y aids the supervised classification of the inputs while enforcing that no other information is encoded in z_e.
3. The random explicit loss $\mathcal{L}_{\mathrm{E}} = \mathrm{CE}(\tilde{y}_{\mathrm{r}}, y_{\mathrm{r}})$: A subsequent re-encoding $(\tilde{y}_{\mathrm{r}}, \tilde{z}_{\mathrm{i}}) = \mathcal{E}(\tilde{x})$ should correctly predict the class y_{r}. $\mathcal{L}_{\mathrm{E}}$ forces the decoder to make use of the information in the explicit part, as otherwise the encoder would not be able to infer this information from $\tilde{x}$.
4. The implicit consistency loss $\mathcal{L}_{\mathrm{I}} = \big\|\tilde{z}_{\mathrm{i}} - \hat{z}_{\mathrm{i}}\big\|_2^2$: Re-encoding $\hat{x}$ with $(\hat{y}_{\mathrm{r}}, \hat{z}_{\mathrm{i}}) = \mathcal{E}_{\mathrm{Y}}(\hat{x})$ should yield the same implicit part as re-encoding $\tilde{x}$. This ensures that the information in z_{i} is used consistently by the decoder.

2.2 Related Work

Interactive media like augmented and virtual reality require to adapt sound effects to the actual interaction at run-time, which makes the use of traditional recorded sound effects ("Foley") more difficult. Physical [6] and signal [7, 8] models can be parametrized to produce appropriate sounds in real-time. However, many models are limited to a specific sound effect type, and the parametrization often requires a manual mapping from an interaction to the model configuration. Furthermore, computational cost may prohibit a very detailed modelling of the effects.

Several works have addressed the use of deep learning for sound effects. In a multimodal approach, fitting sound effects can be selected from a library by analyzing a silent video sequence of interactions between a drum stick and the environment [9]. A similar method can be used to select more general Foley sound effects [10] or even MIDI events that represent music a person may be playing in the analyzed video [11]. Direct synthesis of a wide variety of sounds can be achieved with GANs [12], which also allow to generate varying footsteps specifically [13]. However, the models allow limited control over the produced sound effects.

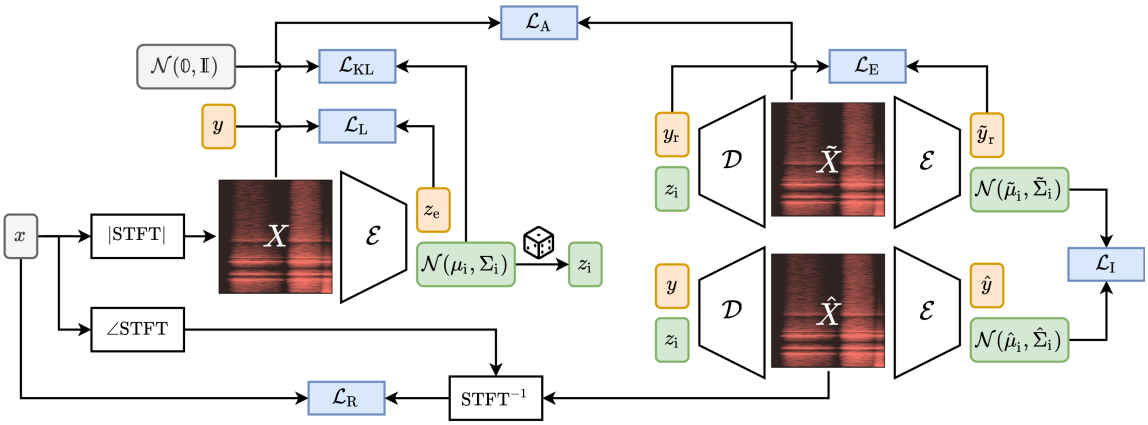

Fig. 2: Visualization of the encoding/decoding method and different losses with their respective inputs at training time. Blue color for losses, grey for inputs, orange for material class labels, green for gesture representations. Further, $\Sigma_\mathrm{i} = \mathrm{diag}(\sigma_\mathrm{i})$, also analogously for $\tilde{\sigma}_\mathrm{i}$ and $\hat{\sigma}_\mathrm{i}$.

3 The VYAE Model

To learn the analysis and synthesis using a disentangled latent representation, we set up the *Variational Y-Autoencoder* (VYAE) model. While the training strategy roughly follows the YAE, we make some important modifications to the original YAE that we highlight in this section.

Our model input x consists of 8192 time-domain audio samples with a sampling rate of 16 kHz. The input of the encoder and output of the decoder is a log-magnitude spectrogram denoted by X and $\hat{X}$, respectively. The spectrogram $X = \log|\mathcal{S}_{N,H}(x) + \epsilon|$ is calculated[2] using the short-time Fourier transform $\mathcal{S}_{N,H}$ with window size $N = 1024$ and hop size $H = 256$. In the following, we omit the subscripts N and H when this window and hop size are used. By discarding the first Fourier coefficient (the DC component of the signal) of the time-frequency representation, we yield 512×32 values for X.

The encoder outputs are an estimate of the material class $z_\mathrm{e} \in [0,1]^C$ for C classes, as well as parameters μ_i, σ_i of a normal distribution representing the estimated gesture $z_\mathrm{i} \sim \mathcal{N}(\mu_\mathrm{i}, \mathrm{diag}(\sigma_\mathrm{i}))$. We use variational regularization for the gesture space to facilitate a continuous and compact representation of different gestures. This allows interpolation between gestures, as well as sampling new and similar gestures from the latent space while preserving a perceptually meaningful output of the decoder. Furthermore, even though this is not generally the case [14], the variational regularization can improve generalization (cf. Fig. 6).

We modify the YAE loss function (3) to accommodate the variational regularization and the inputs/outputs in audio domain. This way, the VYAE loss $\mathcal{L}_\mathrm{VYAE}$ becomes

$$\mathcal{L}_\mathrm{VYAE} = \mathcal{L}_\mathrm{R} + \mathcal{L}_\mathrm{L} + \mathcal{L}_\mathrm{E} + \mathcal{L}_\mathrm{I} + \mathcal{L}_\mathrm{KL} + \mathcal{L}_\mathrm{A}. \quad (4)$$

Fig. 2 gives an overview of the training procedure and the inputs of the loss terms. The individual terms are defined as follows:

1. $\mathcal{L}_\mathrm{R}$: We use a multi-scale spectral (MSS) distance [15], where time-domain audio is transformed to multiple log-magnitude spectrograms with varying hop and window sizes and the L^1 distance between the spectrograms is accumulated. We first have to reconstruct time-domain audio from the decoder output to compare it with the input signal frame x. To do this, we use the phase of the input audio's time-frequency representation and combine it with the spectrogram output of

[2] The small constant ϵ is used to avoid taking a logarithm of 0. We omit it hereinafter.

the decoder to yield

$$\hat{x} = \mathcal{S}^{-1}\Big(\hat{X} \cdot \exp\big(i\angle(\mathcal{S}(x))\big)\Big), \tag{5}$$

where i denotes the imaginary unit and $\angle(\cdot)$ is the angle of the complex Fourier coefficients. Note that only the reconstruction with the correct class label is compared in $\mathcal{L}_\mathrm{R}$. This way, at training time, the phase of x is only used for a signal that contains a reconstruction with the original material, which should ultimately be very similar to x anyway.

The MSS distance is then given by

$$\mathcal{L}_\mathrm{R} = \sum_{(N,H)\in\mathcal{M}} \Big\| \log\big|\mathcal{S}_{N,H}(x)\big| - \log\big|\mathcal{S}_{N,H}(\hat{x})\big| \Big\|_1, \tag{6}$$

where $\mathcal{M}$ is a set of window and hop sizes ($\mathcal{M} = \big\{(2048, 1024), (1024, 256), (512, 128)\big\}$ in our experiments).

2. $\mathcal{L}_\mathrm{L}$ and $\mathcal{L}_\mathrm{E}$: We use cross-entropy between y and z_e and between y_r, and $\tilde{y}_\mathrm{r}$, respectively, as in the original YAE.
3. $\mathcal{L}_\mathrm{I}$: The re-encoding of $\tilde{X}$ and $\hat{X}$ yields parameters $\tilde{\mu}_\mathrm{i}$, $\tilde{\sigma}_\mathrm{i}$ and $\hat{\mu}_\mathrm{i}$, $\hat{\sigma}_\mathrm{i}$ to represent the estimated gesture (cf. Fig. 2). To fulfil the purpose of $\mathcal{L}_\mathrm{I}$ in the original YAE, the distributions $\mathcal{N}(\tilde{\mu}_\mathrm{i}, \mathrm{diag}(\tilde{\sigma}_\mathrm{i}))$ and $\mathcal{N}(\hat{\mu}_\mathrm{i}, \mathrm{diag}(\hat{\sigma}_\mathrm{i}))$ should be as similar as possible. We quantify the similarity using the Bhattacharyya distance [16]. For two multivariate normal distributions $\mathcal{N}(\mu_1, \Sigma_1)$ and $\mathcal{N}(\mu_2, \Sigma_2)$, the loss term is given by

$$\begin{aligned}\mathcal{L}_\mathrm{I} = &\frac{1}{8}(\mu_1 - \mu_2)^T \Sigma^{-1} (\mu_1 - \mu_2) \\ &+ \frac{1}{2}\log\left(\frac{\det\Sigma}{\sqrt{\det\Sigma_1 \det\Sigma_2}}\right)\end{aligned} \tag{7}$$

where $\Sigma = (\Sigma_1 + \Sigma_2)/2$. As the dimensions of the VAE are assumed to be independent, Σ_1 and Σ_2 are diagonal and their inverse and determinant can be calculated efficiently.

4. $\mathcal{L}_\mathrm{KL}$: We use the KL divergence $\mathrm{KL}\big(\mathcal{N}(\mu_\mathrm{i}, \mathrm{diag}(\sigma_\mathrm{i})) \,\|\, \mathcal{N}(\mathbb{0}, \mathbb{I})\big)$ as in (2). It is sufficient to regularize the first encoding, as this ensures a loss for the encoder that favors the continuous and compact representation.

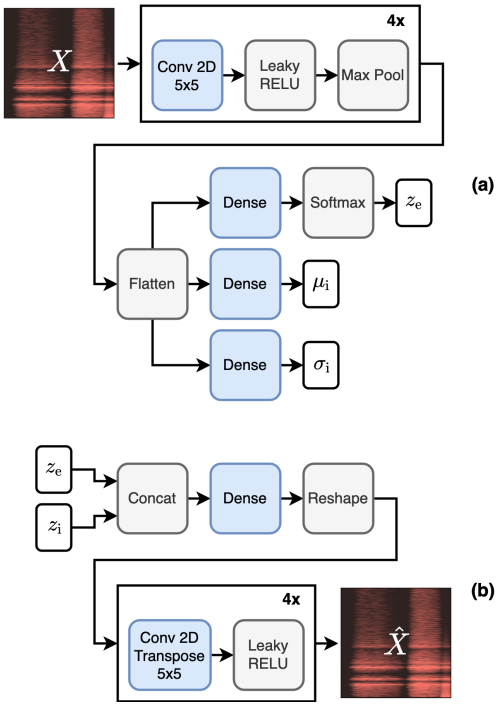

Fig. 3: Encoder **(a)** and decoder **(b)** architecture. Blue color for building blocks with learnable weights.

5. $\mathcal{L}_\mathrm{A}$: An adversarial loss between X and $\tilde{X}$ (the decoder output generated with a random material) improves the decoding of gestures where the material has changed. To achieve this, we use a discriminator $\mathcal{A}_\mathrm{Y} : \mathcal{X} \times [0,1]^C \to \{0, 1\}$ that in addition to X and $\tilde{X}$ receives the corresponding material class label as an input. This way, $\mathcal{A}_\mathrm{Y}$ learns to discriminate material-specific characteristics of the spectrogram.

   The loss for $\mathcal{A}_\mathrm{Y}$ is separate from $\mathcal{L}_\mathrm{VYAE}$ and given by

$$\mathcal{L}_{\mathcal{A}_\mathrm{Y}} = \mathrm{CE}\big(\mathbb{1}, \mathcal{A}_\mathrm{Y}(X, y)\big) + \mathrm{CE}\big(\mathbb{0}, \mathcal{A}_\mathrm{Y}(\tilde{X}, y_\mathrm{r})\big), \tag{8}$$

   where $\mathbb{1}$ and $\mathbb{0}$ are the vectors of all ones and zeros of appropriate dimension. The loss term for $\mathcal{E}$ and $\mathcal{D}$ is

$$\mathcal{L}_\mathrm{A} = \mathrm{CE}\big(\mathbb{1}, \mathcal{A}_\mathrm{Y}(\tilde{X}, y_\mathrm{r})\big). \tag{9}$$

Figure 3 shows the neural network architectures used for encoder and decoder. The encoder consists of a 2D convolutional neural network (CNN) with

four layers with an increasing number of kernels $(16, 32, 48, 64)$ and a 5×5 kernel size. Each layer is followed by a *LeakyRELU* [17] non-linearity and a max pooling layer to downsample the layer output on the time- and frequency axes by selecting the maximum value in each 1×4 block in the first layer and 2×2 block in each subsequent layer. The last convolutional layer outputs 4096 values that are flattened to one dimension and form the input of three fully-connected (dense) layers which output z_e, $\mu_\mathrm{i} \in \mathbb{R}^L$ and $\sigma_\mathrm{i} \in \mathbb{R}^L$ with $L = 32$, respectively. As z_e represents a probability distribution over the material classes, the softmax nonlinearity ensures that the values sum to 1. The discriminator $\mathcal{A}_\mathrm{Y}$ uses the same architecture as the encoder, where the additional class label input is concatenated to the flattened convolutional layer output and the final dense layer outputs a single value between 0 and 1.

The decoder concatenates the inputs z_e and z_i, resulting in $L + C$ values that form the input of a dense layer with 4096 outputs, which are reshaped to match the output dimensions of the encoding convolutional layers. Four subsequent upsampling convolutional layers invert the dimensionality reduction of the max pooling to yield the output log-magnitude spectrogram $\hat{X}$.

4 Experiments

4.1 Datasets

We compile two datasets to evaluate the proposed approach of disentangling material and gesture in recordings of real interaction sounds.

The *Spoon/Bowl* (SB) dataset is a collection of two-minute recordings of interactions using spoons in a bowl (scraping, tapping, etc.). In each recording, a different combination of plastic/metal/wooden spoon and plastic/metal/wooden/glass/porcelain bowl is used to perform random interactions, yielding $C = 15$ classes and 30 minutes of audio material in total. All interactions were recorded in a silent environment with a close-up large-diaphragm condenser microphone. We generate a total of 95000 random excerpts of length 8192 samples (0.512 seconds at 16 kHz sampling rate) from the audio files to form a training dataset. As a test set, we recorded additional audio clips of 20 seconds duration for each class that are not part of the training set. The test set additionally contains interactions with different spoons and bowls recorded in varying acoustic conditions that do not occur in the training set at all. We use them as an "unseen" class to evaluate the generalization of gesture estimations. We denote the classes by "[spoon material] / [bowl material]" and "unseen" in the following.

The *Freesound Footsteps* (FF) dataset consists of 55 recordings of footsteps obtained from the Freesound [18] database with a total of around 17 minutes of audio material. The recordings were selected to contain a variety of footstep sounds, including dry, wet, indoor and outdoor surfaces, with a total of $C = 55$ classes. Background noise, reverb and other degradations are present on some recordings, and all were converted to a single channel signal with 16 kHz sampling rate. As opposed to the SB data, the amount of data for each class varies, with recording lengths between 2 and 118 seconds. We generate 10 random excerpts of the same length as the SB excerpts for each second of audio to form a training set, resulting in a total of 10018 excerpts (between 22 and 1187 per class). We reserve longer clips from two classes ("high heels" and "carpet") that are not part of the training set for evaluation purposes.

4.2 VYAE Model

We train the full model as described in Section 3 separately with the SB and FF datasets for 6.5M and 0.7M steps (70 epochs), respectively, using a batch size of 8 and a learning rate of 10^{-4}. The training can be completed in less than 60 minutes on a single Nvidia Geforce RTX 2080 Ti GPU using Tensorflow. We multiply $\mathcal{L}_\mathrm{R}$ by a constant factor of $\frac{1}{300}$ to facilitate a comparable magnitude of all loss terms.

To evaluate the disentanglement in a real-time setting, we use the test sets of SB and FF and process the audio signals frame-wise with the corresponding trained VYAE model. By using an overlap of 128 samples between frames, we can cross-fade between successive frames to avoid potential discontinuities in the reconstructed audio and are able to process a continuous audio stream with our model. Successful disentanglement should allow to

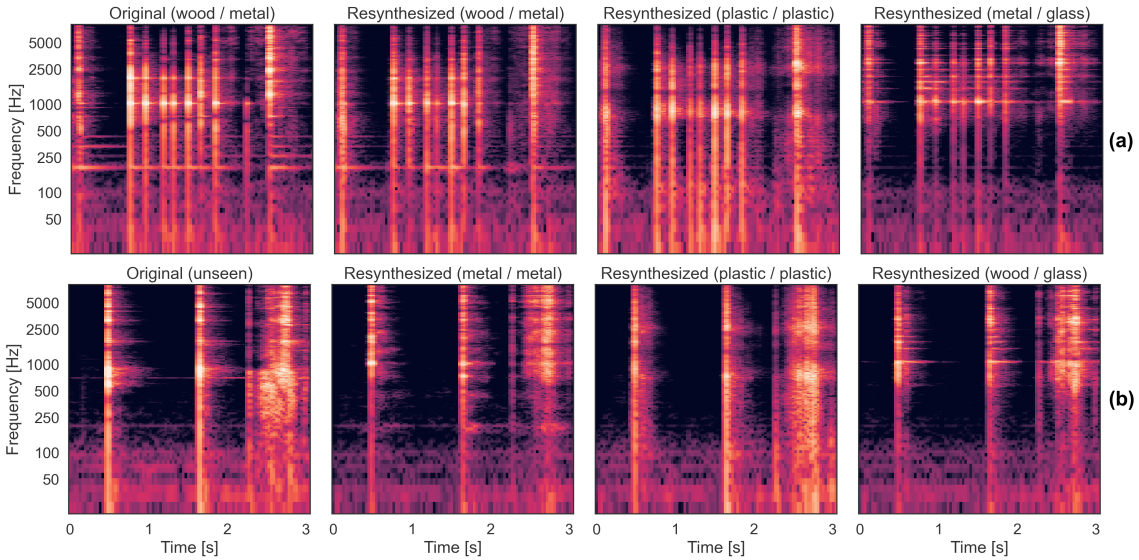

Fig. 4: Results with Model I trained on the Spoon/Bowl dataset. **(a)** Original class "metal / metal". **(b)** Original class "unseen" that is not part of the training dataset.

exchange the material while preserving the gesture in the output signal. We discard the estimated class z_e of each frame and manually specify the material class y in the input of the decoder.

4.3 Variational regularization and adversarial loss

To investigate the effectiveness of variational regularization and adversarial loss, we train three different versions of the VYAE model (see Table 1) with the same parameters as above. Model I is the full model using the $\mathcal{L}_{\mathrm{VYAE}}$ as in (4), including $\mathcal{L}_{\mathrm{A}}$ and $\mathcal{L}_{\mathrm{KL}}$. Models II and III are trained without adversarial loss $\mathcal{L}_{\mathrm{A}}$. The variational regularization $\mathcal{L}_{\mathrm{KL}}$ is additionally removed in Model III, where the encoder outputs the L-dimensional gesture representation directly.

Model	I	II	III
Variational reg. $\mathcal{L}_{\mathrm{KL}}$	Yes	Yes	No
Adversarial loss $\mathcal{L}_{\mathrm{A}}$	Yes	No	No

Table 1: Model configurations to investigate the effects of the loss terms $\mathcal{L}_{\mathrm{KL}}$ and $\mathcal{L}_{\mathrm{A}}$.

To compare the different model outputs to a real recorded example from the target class, we compile a set of single short impact interactions from the SB test set and align them in time to allow a direct qualitative comparison.

5 Results and Discussion

Figure 4 shows excerpts from the longer sequences of the SB test set. We encourage the reader to listen to the examples on the accompanying website[3]. The resynthesis of the examples with different material classes demonstrates that the material characteristics (visually best represented by the strong modes at different frequencies) can be changed independent of the gesture. This includes the "unseen" class in Fig. 4 (b), which shows that the learned gesture representation generalizes to encode the actual underlying factors of variation in the spoon/bowl interaction. However, artifacts of improper disentanglement are sometimes visible and audible. As an example, Fig. 4 (a) exhibits a peak below 250 Hz in the spectrum of the "wood / metal" recording that is preserved to some degree in the resynthesized versions with the "plastic / plastic" material class.

In total, we analyze and resynthesize 1785 frames for the sequential processing of the SB test set

[3] https://audiolabs-erlangen.de/resources/2022-AVAR-InteractionSounds

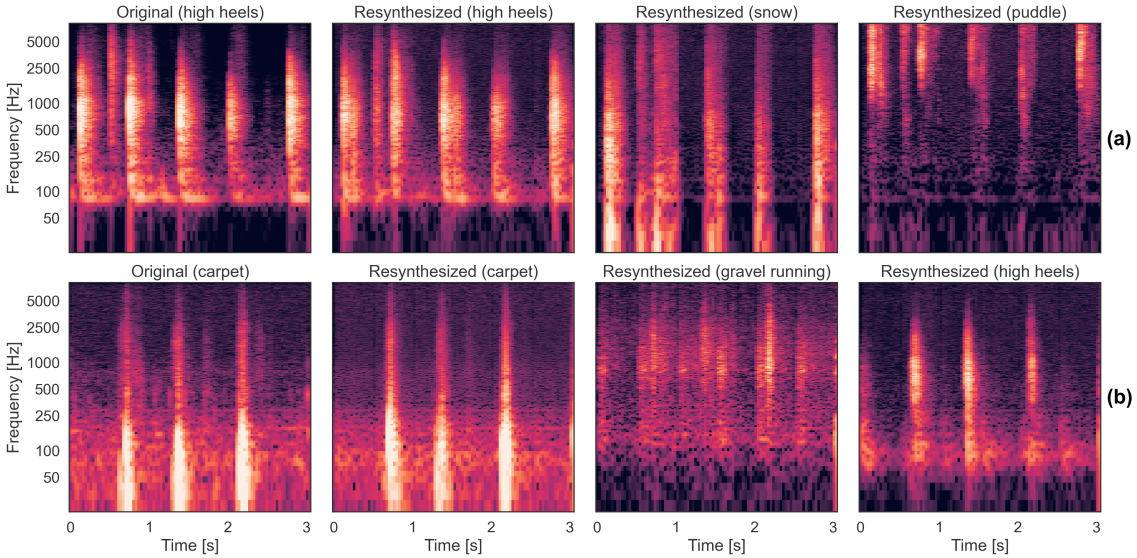

Fig. 5: Results with Model I trained on the Freesound Footsteps dataset. **(a)** Original class "high heels". **(b)** Original class "carpet".

excluding the "unseen" class. The computational complexity of the model easily allows the real-time execution on a recent computer. With an Apple M1 Max CPU, we can analyze and resynthesize a frame of 0.512 seconds at 16 kHz sampling rate in 0.027 seconds on average. Furthermore, for 96% of the frames, the material of the input signal is correctly classified, while most misclassified frames contain very little activity.

Results with the FF dataset are shown in Figure 5. With the FF test set, we can find some combinations where the synthesis with changed material fails. In Fig. 5 (b), the slow steps in the "carpet" class are resynthesized with the "gravel running" class label. As the "gravel running" class recording only contains faster steps, the model produces fast steps from the "carpet" encoded gesture as well. This shows how the learned disentanglement also depends on a balanced training set that contains all expected kinds of gestures for all classes.

Another limitation of our method manifests in the conversion from "carpet" to "high heels". As we use the phase from the input signal for the output, the more transient "high heels" steps are not accurately reproduced, even though the magnitude spectrogram is similar to an original spectrogram from the class (cf. Fig. 5). As the conversion from magnitude spectrogram to time domain at runtime is independent of the training method, a different phase reconstruction method, for example based on deep learning [4], could further improve our results.

Figure 6 shows the effect of variational regularization and adversarial loss. We take a frame with a single impact interaction from the "unseen" class recording, estimate the gesture with the encoder, and resynthesize the sound with the classes "plastic / wood", "plastic / glass", and "plastic / metal". The first column shows a recorded example of the respective class for comparison, while the other three columns show spectrograms of the resynthesized signal. Model III (without $\mathcal{L}_{\mathrm{KL}}$, column 4) clearly fails to generalize to the unseen class input and the resynthesis with different classes exhibits only small variations in the output signal. Model II (without $\mathcal{L}_{\mathrm{A}}$, column 3) produces some artifacts related to the more reverberant recording conditions of the "unseen" class example, whereas Model I performs best in this comparison.

6 Conclusions

In this paper, we showed how a Variational Y-Autoencoder makes it possible to learn disentan-

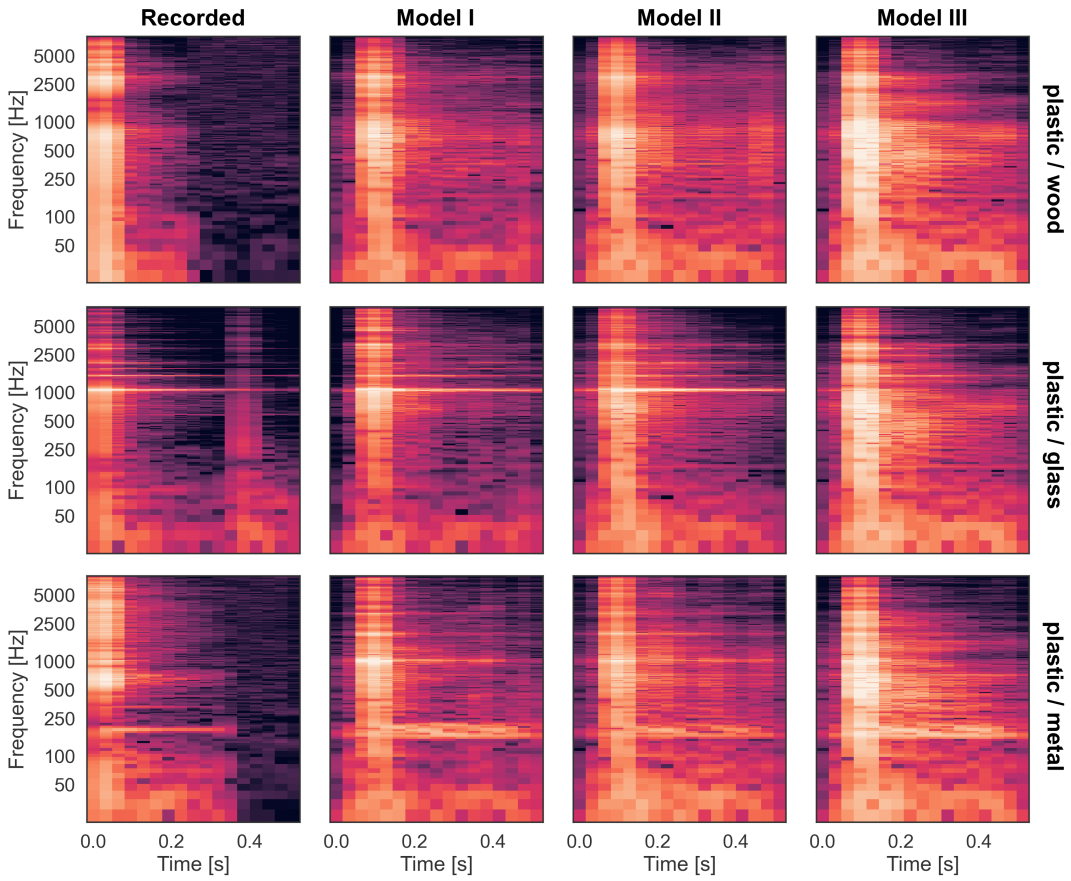

Fig. 6: Results with different models trained on the Spoon/Bowl dataset. The first column shows a recorded example from the material classes "plastic / wood", "plastic / glass", and "plastic / metal". Columns 2-4 show a resynthesis using these classes and a gesture estimated from an "unseen" example.

gling gesture and material in recordings of interaction sound effects. We improved the generalization and resynthesis quality of the model by adding variational regularization and an adversarial loss to the training procedure. The model allows for resynthesis of the sound effect with an exchanged material, even when the encoder input does not belong to a class known to the model. To our knowledge, this is the first application of a timbre transfer method to interaction sound effects. While more complex scenarios, e.g. including nonisotropic materials, may require a multi-modal approach for convincing resynthesis, this pure *audio-to-audio* transformation may be an interesting tool for sound design in mixed reality applications. In future work, we would like to explore disentanglement methods that facilitate a learnable representation of the material as well. This way, authoring of the desired material characteristics for a sound effect could be done by encoding recordings of a few example interactions and storing the estimated material representation for later resynthesis.

Acknowledgements

During this project, Simon Schwär was supported by a fellowship within the IFI programme of the German Academic Exchange Service (DAAD). The International Audio Laboratories Erlangen are a joint institution of the Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU) and Fraunhofer Institute for Integrated Circuits IIS.

References

[1] Patacchiola, M., Fox-Roberts, P., and Rosten, E., "Y-Autoencoders: Disentangling latent representations via sequential encoding," *Pattern Recognition Letters*, 140, pp. 59–65, 2020, doi:10.1016/j.patrec.2020.09.025.

[2] Mirza, M. and Osindero, S., "Conditional Generative Adversarial Nets," *ArXiv preprint*, abs/1411.1784, 2014.

[3] Kingma, D. and Welling, M., "Auto-Encoding Variational Bayes," in *2nd International Conference on Learning Representations (ICLR)*, Banff, AB, Canada, 2014.

[4] Kumar, K., Kumar, R., de Boissiere, T., Gestin, L., Teoh, W. Z., Sotelo, J., de Brebisson, A., Bengio, Y., and Courville, A., "MelGAN: Generative Adversarial Networks for Conditional Waveform Synthesis," in *Proceedings of the 33rd Conference on Neural Information Processing Systems (NeurIPS)*, 2019.

[5] Caillon, A. and Esling, P., "RAVE: A variational autoencoder for fast and high-quality neural audio synthesis," *ArXiv preprint*, abs/2111.05011, 2021.

[6] Bilbao, S., *Numerical Sound Synthesis: Finite Difference Schemes and Simulation in Musical Acoustics*, John Wiley & Sons, Ltd., USA, 2009.

[7] 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.

[8] Cook, P. R., *Real Sound Synthesis for Interactive Applications*, A. K. Peters, Ltd., USA, 2002, ISBN 1568811683.

[9] Owens, A., Isola, P., McDermott, J. H., Torralba, A., Adelson, E., and Freeman, W., "Visually Indicated Sounds," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2405–2413, 2016, doi:10.1109/CVPR.2016.264.

[10] Ghose, S. and Prevost, J. J., "AutoFoley: Artificial Synthesis of Synchronized Sound Tracks for Silent Videos With Deep Learning," *IEEE Transactions on Multimedia*, 23, pp. 1895–1907, 2021, doi:10.1109/TMM.2020.3005033.

[11] Gan, C., Huang, D., Chen, P., Tenenbaum, J. B., and Torralba, A., "Foley Music: Learning to Generate Music from Videos," in *16th European Conference on Computer Vision (ECCV)*, online, 2020.

[12] Donahue, C., McAuley, J., and Puckette, M., "Adversarial Audio Synthesis," in *7th International Conference on Learning Representations (ICLR)*, 2019.

[13] Comunità, M., Phan, H., and Reiss, J. D., "Neural Synthesis of Footsteps Sound Effects with Generative Adversarial Networks," *ArXiv preprint*, abs/2110.09605, 2021.

[14] Bozkurt, A., Esmaeili, B., Tristan, J.-B., Brooks, D., Dy, J., and van de Meent, J.-W., "Rate-Regularization and Generalization in Variational Autoencoders," in *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics*, pp. 3880–3888, 2021.

[15] Engel, J., Hantrakul, L. H., Gu, C., and Roberts, A., "DDSP: Differentiable Digital Signal Processing," in *8th International Conference on Learning Representations (ICLR)*, 2020.

[16] Bhattacharyya, A. K., "On a Measure of Divergence between Two Statistical Populations Defined by Their Probability Distributions," *Bulletin of the Calcutta Mathematical Society*, (35), pp. 99–109, 1943.

[17] Maas, A. L., Hannun, A. Y., and Ng, A. Y., "Rectifier Nonlinearities Improve Neural Network Acoustic Models," in *Proceedings of the 30th International Conference on Machine Learning*, Atlanta, Georgia, USA, 2013.

[18] Font, F., Roma, G., and Serra, X., "Freesound Technical Demo," in *ACM International Conference on Multimedia (MM'13)*, pp. 411–412, ACM, Barcelona, Spain, 2013.