A MODEL YOU CAN HEAR:
AUDIO IDENTIFICATION WITH PLAYABLE PROTOTYPES

Romain Loiseau[1,2]   **Baptiste Bouvier**[3]   **Yann Teytaut**[3]
Elliot Vincent[1,4]   **Mathieu Aubry**[1]   **Loic Landrieu**[2]

[1] LIGM, Ecole des Ponts, Univ Gustave Eiffel, CNRS, France

[2] LASTIG, Univ Gustave Eiffel, IGN, ENSG

[3] STMS Lab, UMR 9912 (IRCAM, CNRS, Sorbonne University), Paris, France

[4] INRIA and DIENS (ENS-PSL, CNRS, INRIA)

`romain.loiseau@enpc.fr`

ABSTRACT

Machine learning techniques have proved useful for classifying and analyzing audio content. However, recent methods typically rely on abstract and high-dimensional representations that are difficult to interpret. Inspired by transformation-invariant approaches developed for image and 3D data, we propose an audio identification model based on learnable spectral prototypes. Equipped with dedicated transformation networks, these prototypes can be used to cluster and classify input audio samples from large collections of sounds. Our model can be trained with or without supervision and reaches state-of-the-art results for speaker and instrument identification, while remaining easily interpretable. The code is available at: `https://github.com/romainloiseau/a-model-you-can-hear`

1. INTRODUCTION

The emergence of deep learning approaches dedicated to audio analysis has led to significant performance improvements [1, 2]. Although these methods take sound excerpts as input, they typically rely on high-dimensional latent spaces, making the interpretation of their decisions difficult and limiting the insights gained on the considered data. Furthermore, such methods typically require large corpora of annotated sounds for supervision. We take a different approach, inspired by the recent Deep Transformation-Invariant (DTI) clustering [3] framework which learns prototypes in input space to analyze images or 3D point clouds [4]. Each prototype is associated with a set of transformations (*e.g.* rotations or morphological transformations) learned in the manner of Spatial Transfor-
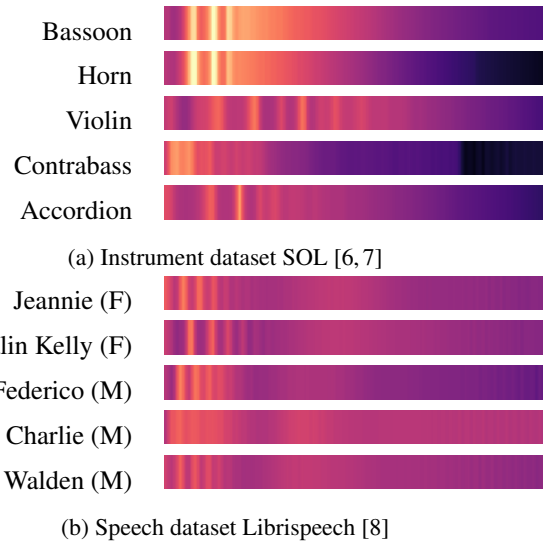

(a) Instrument dataset SOL [6, 7]

(b) Speech dataset Librispeech [8]

Figure 1: **Playable Prototypes.** Our model rely on a small set of spectral prototypes that can be used for clustering and classification. We show examples of such prototypes learned from two datasets.

mation Networks [5]. The DTI clustering model optimizes a reconstruction loss and associates each input with the prototype that provides the best reconstruction. We propose adapting this approach to the audio domain. The prototypes become log-scaled, Mel-frequency spectrograms, and their associated learnable transformations correspond to plausible sound alterations. We also introduce an additional loss based on cross-entropy in the supervised setting, which we demonstrate can boost the classification accuracy while preserving some interpretability.

2. RELATED WORK

Audio Classification. Musical instrument and speaker identification are two of the standard tasks used to benchmark audio classification models. Although early musical instrument identification methods focused on distinguish-
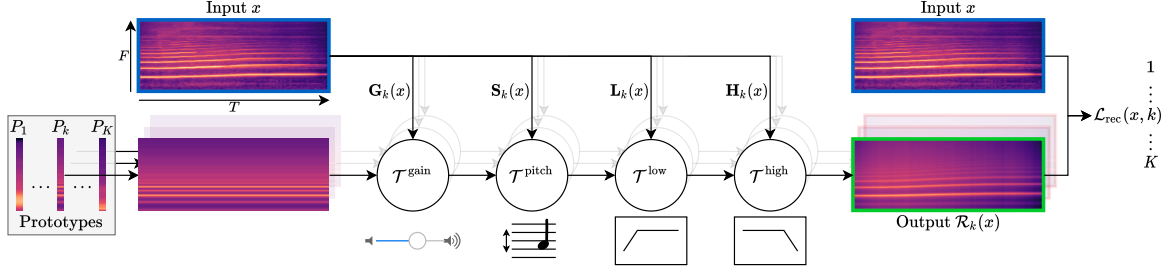
Figure 2: **Method overview.** Given an **input sound**, we predict for each prototype a *gain*, a *pitch* shift, as well as *low* and *high frequency filters* at each timestamp to generate the **output**. Prototypes and transformations are learned jointly using a reconstruction loss in either a supervised or unsupervised setting.

ing individual tones achieve appreciable precision, support vector machines operating on spectro-temporal sound representations can reach almost perfect accuracy. Similarly to supervised classification of instruments that play individual notes that could be considered solved [9, 10], speaker identification is mostly handled by convolutional and/or recurrent neural networks [11], which achieve almost perfect performances [12]. However, these models rely on complex and abstract latent representations that are not easily interpretable and cannot be visualized as spectrograms or heard in the audio domain.

Prototype-Based Methods. Auto-Encoders [13, 14] learn a compact latent representation of an input sample and are trained using a reconstruction loss. Using regularizations and constraints, the latent space can be adjusted for various tasks, such as classification [15]. However, the learned features remain largely non-interpretable and abstract. Inspired by the literature on images [16], the authors of [17] propose to generate prototypes in the latent space and analyze their similarity to the encoded input samples using a frequency-dependent measure. Although the latent prototypes can be decoded into input space, their role in class prediction remains obfuscated, as all similarities are mixed with a learned linear layer.

Transformation-Invariant Modeling. Deep transformation-invariant clustering [3] learns explicit prototypes in input space. Each prototype is equipped with dedicated transformation networks, allowing a small set of prototypes to faithfully represent a collection of samples. The resulting models can be used for downstream tasks such as classification [3], few-shot segmentation [4], and even multi-object instance discovery [18]. Jaderberg *et al.* [5] also proposes learning differentiable transformations in the input space, and feed the transformed data to a classification network. We propose to extend these ideas to the audio domain by learning prototypical log-Mel spectrograms along with adapted transformations.

3. METHOD

We consider a set of N audio clips $x_1, \cdots, x_N \in \mathbb{R}^{F \times T}$, each characterized by their log-Mel spectrogram with T time steps and a spectral resolution of F bins. The samples

are annotated with class labels $y_1, \cdots, y_N \in \mathcal{Y}$ with $\mathcal{Y}$ a given class set $\{1, \cdots, K\}$. Our goal is to learn an interpretable model that can perform classification and clustering for the audio sample collection $x_1, \cdots, x_N$. Our model is based on prototypical spectrograms equipped with spectral transformation networks, which we present in Sec. 3.1. We propose an unsupervised training scheme in Sec. 3.2, and a supervised setting in Sec. 3.3. Finally, we provide details on parameterization and training of the model in Sec. 3.4.

3.1 Sound Reconstruction Model

Our model consists of K prototypes $P_k \in \mathbb{R}^F$ directly interpretable in the spectral domain, each equipped with a set of dedicated transformation networks (See Figure 2). Both prototypes and networks are jointly trained to reconstruct input samples. We can then use the reconstruction error as a measure of the compatibility between a sample and a given prototype.

Spectral Transformations Model. Even a single instrument or speaker cannot be meaningfully characterized by a single sound. In fact, they are typically capable of producing a rich and varied set of sounds. We propose equipping each prototype with a set of transformation networks that learn to change specific characteristics of the prototype, such as its amplitude and pitch, so that it can faithfully approximate a wide variety of samples. This also allows the prototypes to learn more subtle audio attributes, such as timbre or intonations.

Transformation networks associated to each prototype P_k take as input an audio sample $x \in \mathbb{R}^{F \times T}$ as input and predicts a spectral transformation for each time step t in $[1, T]$. For each prototype k, we propose a set of transformations specifically designed for the audio domain, illustrated in Figure 3:

• *Amplification.* Given an audio sample x, we predict a gain $\mathbf{G}_k(x) \in \mathbb{R}^T$ for all time step t. The transformation $\mathcal{T}_g^{\text{gain}}(m)$ adds the gain g to all frequencies of a log-Mel spectrogram $m \in \mathbb{R}^F$.

• *Pitch Shifting.* We map an audio sample x to a pitch shift $\mathbf{S}_k(x) \in \mathbb{R}^T$ for all time step t. The transformation $\mathcal{T}_s^{\text{pitch}}(m)$ shifts the log-Mel spectrogram $m \in \mathbb{R}^F$ by s.
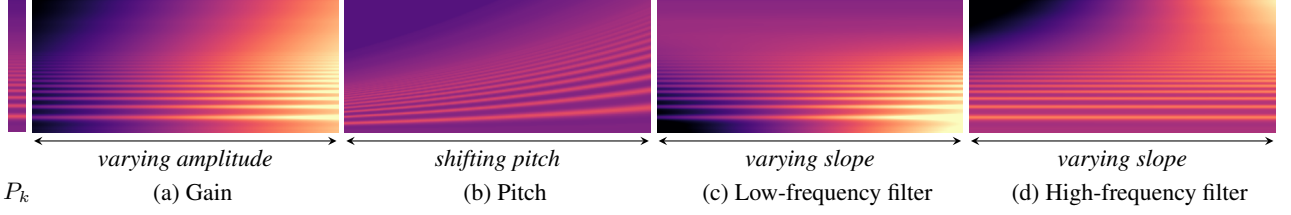
P_k *varying amplitude*   *shifting pitch*   *varying slope*   *varying slope*

(a) Gain   (b) Pitch   (c) Low-frequency filter   (d) High-frequency filter

Figure 3: **Spectral Transformations.** Our proposed transformations learn to alter the gain (a), pitch (b) and frequency support (c,d) of a given prototype spectrogram P_k.

We use linear interpolation for handling non-integer shifts.

• *High- and Low-Frequency Filter.* Given an audio sample x, we define for each time t an affine low-frequency filter $\mathbf{L}_k(x)[t] \in \mathbf{R}^F$ by predicting its slope and cutoff frequency. Similarly, we predict a high-frequency filter $\mathbf{H}_k(x)[t] \in \mathbb{R}^F$. The transformations $\mathcal{T}_L^{\text{low}}$ and $\mathcal{T}_H^{\text{high}}$ simply add the corresponding filters L and H to a log-Mel spectrogram.

These transformations can be written as follows for a log-Mel spectrogram $m \in \mathbb{R}^F$ and a mel-frequency $f \in [1, F]$:

$$\mathcal{T}_g^{\text{gain}}(m)[f] = m[f] + g \tag{1}$$

$$\mathcal{T}_s^{\text{pitch}}(m)[f] = m[\Phi(f, s)] \tag{2}$$

$$\mathcal{T}_L^{\text{low}}(m)[f] = m[f] + L[f] \tag{3}$$

$$\mathcal{T}_H^{\text{high}}(m)[f] = m[f] + H[f], \tag{4}$$

with $\Phi(f, s) = A \log_{10}\left(1 + s\left(10^{f/A} - 1\right)\right)$ the mel-aware pitch shifting, with $A = 2595$ [19]. Our transformations have links with moment matching methods [20, 21]. In fact, modulating the amplitude, pitch, and spectral support of spectrograms is similar to matching the moment of zeroth order (amplitude gain), first order (pitch shift), and second order (frequency filters).

Reconstruction Model. For each prototype k and each time step t, we successively apply the gain, pitch shift, low-pass, and high-pass transformations in this specific order. The resulting reconstruction $\mathcal{R}_k(x)$ of an input sound x by a prototype P_k is defined as follows:

$$\mathcal{R}_k(x)[t] = \mathcal{T}_{\mathbf{H}_k(x)[t]}^{\text{high}} \circ \mathcal{T}_{\mathbf{L}_k(x)[t]}^{\text{low}}$$
$$\circ \mathcal{T}_{\mathbf{S}_k(x)[t]}^{\text{pitch}} \circ \mathcal{T}_{\mathbf{G}_k(x)[t]}^{\text{gain}}(P_k), \tag{5}$$

where $\mathcal{R}_k(x)[t]$ denotes the t-th timestamp of the reconstruction, and $\circ$ denotes the composition of transformations. Note that the chosen transformations are constrained and limited on purpose, as we want each prototype to represent a restricted and consistent set of audio samples.

We measure the quality of the reconstruction $\mathcal{R}_k(x)$ of an input sample x by the k-th prototype as the spectro-temporal average of their ℓ_2 distance:

$$\mathcal{L}_{\text{rec}}(x, k) = \frac{1}{T} \sum_{t=1}^T \|x[t] - \mathcal{R}_k(x)[t]\|^2. \tag{6}$$

Note that since this function is differentiable and continuous, it can be used as a loss function to train prototypes and transformation networks as defined in the next sections.

3.2 Unsupervised Training

We propose an unsupervised learning scheme in which we do not have access to labels $y_1, \cdots, y_N$ during training. Our goal is to cluster the samples $x_1, \cdots, x_N$ into meaningful groups of sounds that can be well approximated by the same prototype spectrogram and its associated transformations. We can train our model by minimizing the following loss:

$$\mathcal{L}_{\text{clustering}} = \sum_{n=1}^N \min_{k=1}^K \mathcal{L}_{\text{rec}}(x_n, k). \tag{7}$$

This loss is the sum of the reconstruction errors with optimal cluster assignment and is a classical clustering objective used by K-means-based methods.

3.3 Supervised Training

In the supervised setting, each prototype and associated transformation networks are dedicated to a class and trained to reconstruct the samples from this class only. Once trained, the model predicts for an input x the class of the prototype with the best reconstruction.

To promote better class discrimination, we propose a dual reconstruction-prediction objective. We associate each input x with a probabilistic class prediction defined as the softmax of the negative reconstruction error between prototypes. For an input x of label y, we define the prediction loss $\mathcal{L}_{\text{ce}}$:

$$\mathcal{L}_{\text{ce}}(x, y) = -\log\left(\frac{\exp(-\beta \mathcal{L}_{\text{rec}}(x, y))}{\sum_{k=1}^K \exp(-\beta \mathcal{L}_{\text{rec}}(x, k))}\right), \tag{8}$$

where β is a learnable parameter. This loss encourages each prototype to specialize in reconstructing the samples of its associated class. To train our network, we use a weighted sum of both losses:

$$\mathcal{L}_{\text{classif}} = \sum_{n=1}^N \mathcal{L}_{\text{rec}}(x_n, y_n) + \lambda_{\text{ce}} \mathcal{L}_{\text{ce}}(x_n, y_n), \tag{9}$$

with λ_{ce} an hyperparameter set to 0.01.

3.4 Parameterization and Training Details

Architecture. We implement the functions $\mathbf{G}_k$, $\mathbf{S}_k$, $\mathbf{L}_k$ and $\mathbf{H}_k$ as U-Net style neural networks [23] operating on

| | SOL [6, 7] | | | | | | LibriSpeech [8] | | | | | |
| | w/o supervision | | | w supervision | | | w/o supervision | | | w supervision | | |
	OA	AA	$\mathcal{L}_{rec}$	OA	AA	$\mathcal{L}_{rec}$	OA	AA	$\mathcal{L}_{rec}$	OA	AA	$\mathcal{L}_{rec}$
Raw prototypes	29.3	13.2	12.4	28.4	39.5	12.9	13.3	13.2	8.6	60.5	59.5	8.6
∟ *gain* transformation	32.2	15.8	3.4	37.4	40.1	3.8	44.4	43.3	3.4	78.7	79.1	3.4
∟ *pitch-shifting*	**37.6**	**19.1**	2.6	51.6	46.1	2.9	46.8	46.9	2.6	77.2	77.8	2.6
∟ *frequency-filters*	33.4	14.6	**1.5**	55.1	47.7	**2.1**	48.8	49.5	**1.6**	88.8	89.4	**1.6**
∟ cross-entropy loss	—	—	—	**98.9**	**94.5**	2.5	—	—	—	**99.9**	**99.9**	1.9

Table 1: **Ablation Study.** We show the impact of the increasing complexity of our modeling on datasets' validation sets.

	OA	AA
SOL [6, 7]		
Autoencoder + K-means [22]	28.7	12.3
† APNet [17] + K-means [22]	**37.3**	**18.2**
Ours w/o supervision	34.5	15.4
LibriSpeech [8]		
Autoencoder + K-means [22]	11.0	11.1
† APNet [17] + K-means [22]	36.3	36.4
Ours w/o supervision	**48.6**	**49.5**

Table 2: **Clustering Results.** Clustering performances on the test sets. †: Note that APNet requires labels at training time.

	OA	AA	$\mathcal{L}_{rec}$
SOL [6, 7]			
Direct Classification	97.8	94.8	—
APNet [17]	95.3	91.3	**0.1**
Ours w supervision	**99.3**	**95.8**	2.6
LibriSpeech [8]			
Direct Classification	99.4	99.5	—
APNet [17]	97.8	97.8	**0.2**
Ours w supervision	**99.9**	**99.9**	2.6

Table 3: **Classification Results.** Accuracy and reconstruction error computed on the test sets.

4. EXPERIMENTS

We evaluate our approach in both supervised and unsupervised settings and on two datasets.

Datasets. We consider the following datasets:
• **SOL [6,7].** This dataset contains $24\,450$ samples of individual notes played with various playing techniques by 33 different instruments and sampled at 44.1kHz. We evaluate instrument classification.

• **Librispeech [8].** This 1000-hour corpus contains English speech sampled at 16kHz. We selected the 128 predominant speakers from the `train-clean-360` set. We evaluate the classification of speaker.

For both datasets, we randomly selected 70% of the clips for training, 10% for validation, and 20% for testing. We always use as many prototypes as the number of classes in the dataset.

Metrics. To assess the quality of classification models, we report the following metrics:
• *Overall Accuracy (OA).* Percentage of input samples correctly classified by our model, *i.e.*, best reconstructed by the prototype assigned to their true class.

• *Average Accuracy (AA).* Average of the classwise accuracy, computed across classes without weights.

• *Reconstruction error ($\mathcal{L}_{rec}$).* To assess the quality of the reconstruction, we also report the reconstruction error $\mathcal{L}_{rec}$.

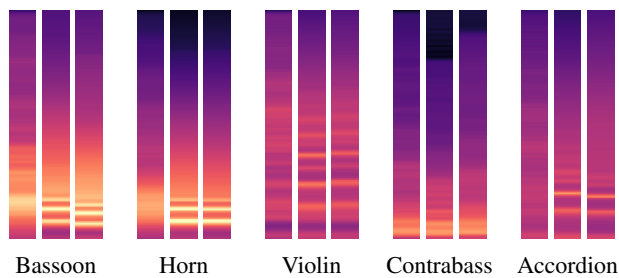Methods trained in the unsupervised setting cluster the

the log-Mel spectrograms $x \in \mathbb{R}^{T \times F}$. To save parameters, all networks share the same encoder, defined as a sequence of 2-dimensional spectro-temporal convolutions and pooling operations. This encoder produces a sequence of feature maps $z_0, \cdots, z_R$ with decreasing temporal and spectral resolutions $T/2^r \times F/2^r$. The spectral dimension of these maps is then collapsed using learned pooling operations: $\tilde{z}_r$ is of resolution $T/2^r$. Each decoder is then defined as a sequence of 1-dimensional temporal convolutions and unpooling operations. The spectral maps $\tilde{z}_0, \cdots, \tilde{z}_R$ are used through skip-connections in the decoders. The prototypes $P_1, \cdots, P_K$ and the inverse temperature β are directly trainable parameters of the model.

Curriculum Learning. We ensure the stability of our model by gradually increasing its complexity along the training procedure. First, we learn the raw prototypes without any transformation. Once the training loss does not decrease for 10 straight epochs, we add the gain transformation. We then add the pitch shift and spectral filters successively following the same procedure.

Implementation details. We use the ADAM [24] optimizer with a learning rate of 10^{-4}, and a weight decay of 10^{-6} for the transformation networks and 0 for β and the prototypes. Our model has 545k parameters with $K = 128$ prototypes. For comparison, the reconstruction model proposed by Zinemanas *et al.* [17] has 1.8M parameters.

| Bassoon | Horn | Violin | Contrabass | Accordion |

Figure 4: **Learned Meaningful Prototypes.** Comparison between the average spectrogram from one class (left), and the prototypes learned in the supervised setting without (middle) and with (right) cross-entropy loss. Our model can learn spectral representations that are crisper and more discriminative than a simple average.

audio samples instead of classifying them. We associate each cluster with the majority class of its assigned training samples. We can then predict for each test sample the class of its best-fitting cluster, allowing us to compute all the classification metrics.

Baselines. To put the performance of our method in context, we trained different baselines:
• *Classification.* We trained a temporal convolutional network to classify log-Mel spectrograms. We supervise this network with the cross-entropy on the labels. This network, which does not provide a reconstruction error, is called *Direct Classification*. We also evaluated the AP-Net [17] approach on our datasets. However, this method is limited in its interpretability, as it relies on latent prototypes and uses a fully connected layer to classify samples based on their similarity to the prototypes.
• *Clustering.* We trained a two-dimensional convolutional autoencoder without skip-connections to reconstruct log-Mel spectrograms. We supervised this network with the ℓ_2 norm between the input and the reconstruction. We then use K-means [22] on the innermost feature maps to cluster the samples. We also use K-means directly on the predicted similarities between feature maps and APNet prototypes [17]. Note this last approach requires to first train APNet in a supervised setting.

4.1 Quantitative Results

Reconstruction. As can be seen in Table 1, each successive addition of a transformation decreases the reconstruction error $\mathcal{L}_{\mathrm{rec}}$ and improves the accuracy of classification. This shows that the curriculum training scheme allows our model to learn insightful and complementary spectral transformations. Supervised with the cross-entropy loss, our model reaches high accuracy, at the cost of a slightly higher reconstruction error. In fact, the additional objective not only encourages the reconstruction models $\mathcal{R}_k$ to provide a good reconstruction for the samples of their assigned class, but also a worse reconstruction for other classes.

Finally, we see that while adding frequency-filters yields better reconstruction, they hurt classification per-

formance on SOL [6, 7] in the unsupervised setting. We hypothesize that the filters makes the transformation too expressive, reducing the specificities of the prototypes.

Clustering. As shown in Table 2, our method performs well compared to the baselines. For SOL, our method is affected by the class imbalance between samples, but still produces competitive results. In particular, note that our approach does not need any labels during training, in contrast to the APNet-based approach [17]. For LibriSpeech, our method produces convincing clustering results for speaker identification.

Classification. As shown in Table 3, our method reaches almost perfect precision for both datasets. While APNet [17] reaches better reconstruction scores, this method learns powerful transformations in an abstract latent space. This leads to a model that is more expressive but less interpretable. In contrast, we restrict the scope of the spectral transformations so that a prototype relates in a meaningful way to the samples that it reconstructs well. Yet, our model is still capable of producing faithful reconstructions, which would be suitable for audio generation tasks.

4.2 Interpretability

As shown in Figure 5, our model learns to reconstruct complex input samples (*e.g.* different notes, techniques, or voices) by automatically adjusting the amplitude, pitch, and spectral support of spectral prototypes. As seen on Figure 4 and Figure 1, these prototypes capture rich spectral characteristics such as timbre, and can be interpreted as an "acoustical fingerprint". Furthermore, as the prototypes are given in the spectral domain, they can be easily played and analyzed. Audio examples featuring the learned prototypes are provided in the supplementary materials.

In Figure 6, we represent the reconstruction of sounds played by various instrument with different prototypes. While the reconstructions are clearly more faithful when using the instruments' dedicated prototype, this opens the perspective for further MIR application such as interpretable timbre transfer.

5. CONCLUSION

We presented a novel approach to audio understanding by representing large collections of audio clips with few spectral prototypes equipped with learned transformation networks. Our model produces concise, expressive, and interpretable representations of raw audio excerpts that can be used to obtain state-of-the-art results for audio classification and clustering tasks.
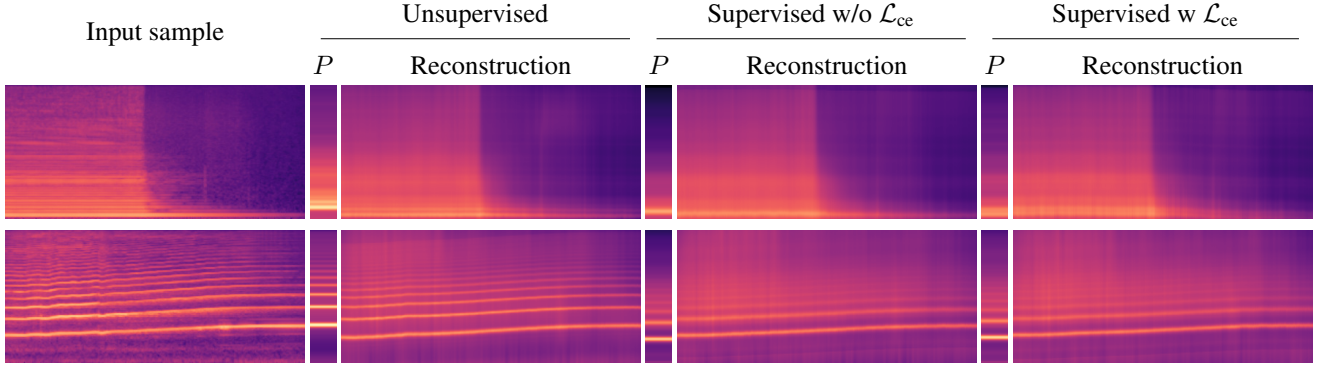
(a) **SOL [6, 7].** Reconstructions of an audio clip of contrabass (top) and clarinet (bottom).
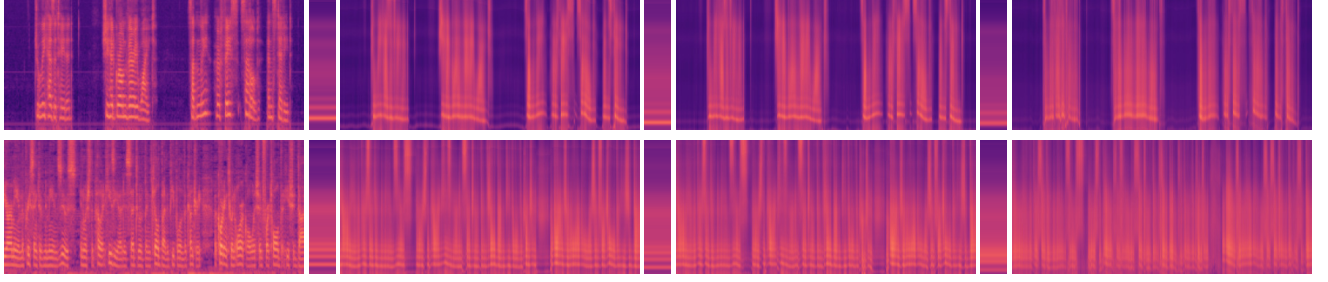
(b) **LibriSpeech [8].** Reconstructions of an audio clip of C. Nendza (F) (top) and C. Berriux (M) (bottom) speaking.

Figure 5: **Reconstruction of Input Sounds.** For each input sample, we show the reconstruction provided by different settings of our model as well as the corresponding prototype P used for reconstruction. Note how the timbre of the speaker or instrument is well represented while the inputs fluctuates in pitch and intensity. This leads to an insightful characterization of each instrument or speaker.
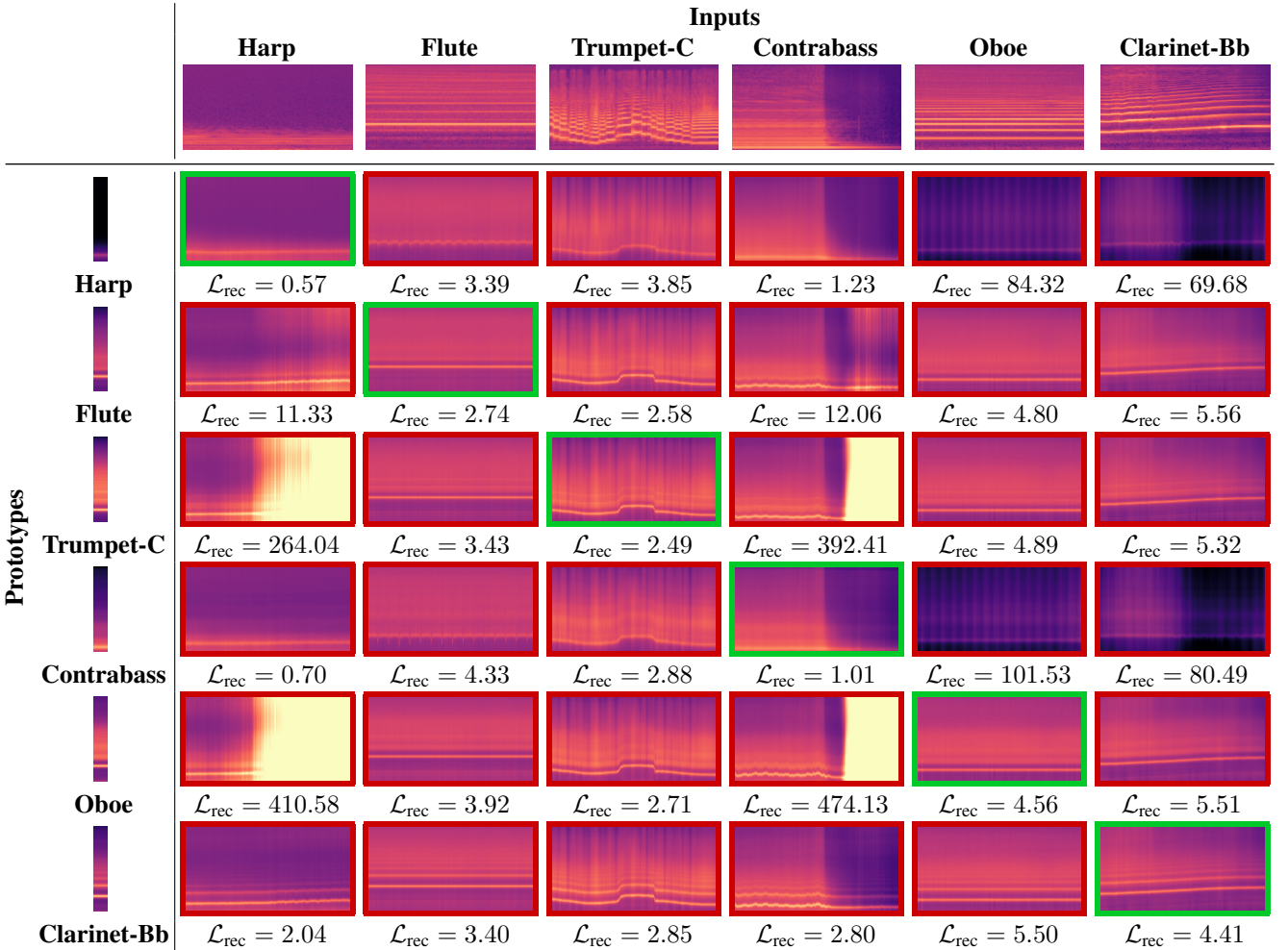

Figure 6: **Reconstructions from Different Prototypes.** We show the reconstructions of input samples (columns) from SOL [6, 7] by learned prototypes (lines) in the supervised setting without cross-entropy loss. The prototype of the corresponding instruments provides the lowest reconstruction error (green frame).

6. REFERENCES

[1] J. Abeßer, "A review of deep learning based methods for acoustic scene classification," *Applied Sciences*, 2020.

[2] N. Ndou, R. Ajoodha, and A. Jadhav, "Music genre classification: A review of deep-learning and traditional machine-learning approaches," in *IEMTRON-ICS*, 2021.

[3] T. Monnier, T. Groueix, and M. Aubry, "Deep Transformation-Invariant Clustering," in *NeurIPS*, 2020.

[4] R. Loiseau, T. Monnier, M. Aubry, and L. Landrieu, "Representing Shape Collections with Alignment-Aware Linear Models," in *3DV*, 2021.

[5] M. Jaderberg, K. Simonyan, and A. Zisserman, "Spatial transformer networks," *NeurIPS*, 2015.

[6] G. Ballet, R. Borghesi, P. Hoffmann, and F. Lévy, "Studio online 3.0: An internet "killer application" for remote access to IRCAM sounds and processing tools," in *Journées d'Informatique Musicale*, 1999.

[7] C. E. Cella, D. Ghisi, V. Lostanlen, F. Lévy, J. Fineberg, and Y. Maresz, "OrchideaSOL: a dataset of extended instrumental techniques for computer-aided orchestration," *ICMC*, 2020.

[8] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: an asr corpus based on public domain audio books," in *ICASSP*, 2015.

[9] D. Bhalke, C. Rao, and D. S. Bormane, "Automatic musical instrument classification using fractional fourier transform based-MFCC features and counter propagation neural network," *Journal of Intelligent Information Systems*, 2016.

[10] V. Lostanlen, J. Andén, and M. Lagrange, "Extended playing techniques: the next milestone in musical instrument recognition," in *International Conference on Digital Libraries for Musicology*, 2018.

[11] Z. Bai and X.-L. Zhang, "Speaker recognition based on deep learning: An overview," *Neural Networks*, 2021.

[12] F. Ye and J. Yang, "A deep neural network model for speaker identification," *Applied Sciences*, 2021.

[13] N. Tits, F. Wang, K. E. Haddad, V. Pagel, and T. Dutoit, "Visualization and interpretation of latent spaces for controlling expressive speech synthesis through audio analysis," *Conference of the International Speech Communication Association*, 2019.

[14] F. Roche, T. Hueber, S. Limier, and L. Girin, "Autoencoders for music sound modeling: a comparison of linear, shallow, deep, recurrent and variational models," *Sound and Music Computing*, 2018.

[15] J. Naranjo-Alcazar, S. Perez-Castanos, P. Zuccarello, F. Antonacci, and M. Cobos, "Open set audio classification using autoencoders trained on few data," *Sensors*, 2020.

[16] O. Li, H. Liu, C. Chen, and C. Rudin, "Deep learning for case-based reasoning through prototypes: A neural network that explains its predictions," in *AAAI*, 2018.

[17] P. Zinemanas, M. Rocamora, M. Miron, F. Font, and X. Serra, "An interpretable deep learning model for automatic sound classification," *Electronics*, 2021.

[18] T. Monnier, E. Vincent, J. Ponce, and M. Aubry, "Unsupervised Layered Image Decomposition into Object Prototypes," in *ICCV*, 2021.

[19] S. S. Stevens, J. Volkmann, and E. B. Newman, "A scale for the measurement of the psychological magnitude pitch," *The Journal of the Acoustical Society of America*, 1937.

[20] H. Tamaru, Y. Saito, S. Takamichi, T. Koriyama, and H. Saruwatari, "Generative moment matching network-based random modulation post-filter for dnn-based singing voice synthesis and neural double-tracking," in *ICASSP*, 2019.

[21] M. Andreux and S. Mallat, "Music generation and transformation with moment matching-scattering inverse networks," in *ISMIR*, 2018.

[22] L. Bottou and Y. Bengio, "Convergence properties of the k-means algorithms," in *NeurIPS*, 1995.

[23] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," *MICCAI*, 2015.

[24] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *ICLR*, 2015.