**Noise schedule.** Choosing the noise schedule $\beta$ has been found to be crucial for the performance of diffusion models [13] [2] and in fact, efforts have been done to learn said variable instead of defining it manually [29]. In our scenario, we require a schedule $\beta$ that accounts for the proposed diffusion-inspired approach, in which $x_0$ is expected to be predominant in the completely perturbed signal $x_T$. Moreover, our perturbation $m$ is the mixture corresponding to $x_0$, thus $x_0$ is contained in the perturbation, which may lead to an abnormally over-loud source in the completely diffused $x_T$. Ultimately, the inference input is a mixture, therefore we propose to use a schedule that produces a latent variable $x_T$ as close as possible to an actual mixture.

Our noise schedule is defined by $\beta_0 = 1^{-4}$, $\beta_T = 0.2$, and $T = 20$, being 20 a reliable option for audio as found in [16]. This noise schedule produces $q(x_T|x_0) \approx m$, and we denote it $\beta_{20}$. Note that the closer $\beta_T$ to 1, we should expect a more aggressive transformation, leading to less interference at the expense of losing quality of the estimated source. Recent diffusion-based works in the audio domain successfully model their task using 4–8 steps [21]. To study the effect of the number of diffusion steps and explore the feasibility of modelling the task with less computational expense, we experiment with a new schedule $\beta_8$, which is defined by $\beta_0 = 1^{-4}$, $\beta_T = 0.5$, and $T = 8$.

**Accompaniment estimation.** To perform accompaniment estimation we initialize $x_0$ to be the musical accompaniment $a$, while the singing voice $v$ is the target $\epsilon$ for training in Eq. 4, and we adjust $\beta$. Since usually the accompaniment is a mixture of multiple sources, and the singing voice – now the perturbation – is normally predominant, we may increase the number of diffusion steps to 100 in the schedule and experiment with a more granular reverse process. The other parameters remain unchanged.

### 2.2 Network details

We propose to use the unconditional vanilla DiffWave to learn the reverse process [16]. Although recent works in music separation propose various improvements with regards to the architectures used, in this work we focus on exploring a novel diffusion-inspired process for source separation. At the same time, we aim at improving a smaller model that has been previously applied for source separation using our diffusion-inspired training and sampling method, while leaving the improvements on the network, or using a different architecture, as future work. Nonetheless, we consider the versatility and light weight of the entire method an advantage. Note that within the scope of this paper, we perform monaural source separation.

**Architecture.** The DiffWave architecture is based on a modified WaveNet [30] extended with bidirectional dilated convolution modules (Bi-DilConv), aiming at removing the autoregressive generation constraint so that the model is non-causal and the reverse process is done in $T$ steps. The said Bi-DilConv modules have been previously applied for the problem of music source separation [12, 28]. The used non-causal WaveNet consists of a stack of $L$ residual layers, which are equally grouped into $N$ blocks. Therefore, each block includes $\nicefrac{L}{N}$ layers with skip-connections as the original WaveNet. A Bi-DilConv module with kernel-size 3 is included in each layer. The size of the dilation, initialized at 1, is doubled at each layer of a block: $[1, 2, 4, 8, ..., 2\nicefrac{L}{N} - 1]$. Before going through the stacked blocks, the input is projected using a 1D convolutional layer of $C$ channels of features. Similarly to the original WaveNet, the output is obtained by summing the skip connections of all the residual blocks. For our experiments we configure the WaveNet architecture with $L = 30$, $N = 10$, and $C = 64$, which in [16] is found to effectively work while preserving the light weight of the architecture.

**Diffusion step embedding.** Since the training process is based on optimizing Eq. 4 given a pair $(x_t, x_0)$, we need to input the diffusion step $t$ to the model. We use a 128-dimension encoding vector for each $t$ [16], defined as [31]:

$$\begin{aligned} t_{embed} = \Big[ & sin\left(10^{\frac{0*F}{S-1}} t\right),\ ...,\ sin\left(10^{\frac{(S-1)*f}{S-1}} t\right), \\ & cos\left(10^{\frac{0*F}{S-1}} t\right),\ ...,\ cos\left(10^{\frac{(S-1)*F}{S-1}} t\right) \Big] \end{aligned} \tag{5}$$

where $F$ is the embedding factor and $S$ is half the embedding size. Note that $F$ and $S$ are pre-defined and fixed hyperparameters, which in our experimentation are set to $F = 4$ and $S = 64$. Next, the embedded diffusion step is passed through three dense layers, the first two having size $S * 2$, while the latter maps the latent embedding into the $C$ channels the input is projected to, therefore we can add the embedding to the input of each residual layer.

**Conditioning.** Using a vocoder paradigm, diffusion-based approaches for audio modelling usually use conditioning to guide the signal generation, providing clues to obtain a particular desired output. Audio-related diffusion works in the literature propose to guide the signal generation using, for instance, mel-spectrogram [16, 32] or linguistic features [33]. We do not condition our network for three reasons: **(1)** Conditioning our vocal extraction model, for instance on the mel-spectrogram of the target vocal signal, would probably improve the output quality, however the task of source separation assumes that data of this kind is not available at inference, **(2)** Our latent variable $x_T$ is not an isotropic Gaussian but a known mixture recording containing the target signal, therefore it serves as a conditioner to guide the transformation towards the isolated singing voice, **(3)** We reduce the model size.

### 2.3 Post-processing

To account for a possible over-increase of the signal amplitude during the reverse process, the output of said process is clamped to the amplitude levels of the input mixture.

During the iterative signal transformation at inference we may generate audio content and artifacts not contained in the original vocal signal. Although not perceptually significant, these artifacts are heavily penalized by objective evaluation metrics [34]. Moreover, the iterative nature of the algorithm may accumulate several of the said artifacts

[2] Despite being aware that our perturbation is a deterministic signal instead of stochastic noise, we still use the term noise schedule in this work for easier understanding in relation with cited works.

in the final prediction. As an optional step, we use the multichannel Wiener filter to improve our separation, a well-known process used in numerous source separation works [35]. We use the Python version in `norbert` [36].

Since our model operates in an end-to-end manner, we use the following procedure to apply the Wiener filtering. Let $\hat{v}$ be an estimated vocal source and $\hat{a}$ the corresponding estimated musical accompaniment. The inputs of the Wiener process are the Short-Time Fourier Transform (STFT) of the input mixture $m$, and the magnitude STFT of both $\hat{v}$ and $\hat{a}$ estimates. The outputs of the Wiener process are the filtered complex spectrograms for $\hat{v}$ and $\hat{a}$. We take the magnitude of said spectrograms and combine it with the corresponding phases $\phi(\hat{v})$ and $\phi(\hat{a})$ that our proposed model estimates. In that sense, we do not use the Wiener filtering to estimate the phase as typically done for spectrogram-based approaches that cannot estimate such complex target, but refine our estimation using the Expectation-Maximization algorithm [37] to make sure that the predicted signal is contained in the input mixture.

# 3. RELATION WITH PREVIOUS WORK

The literature on waveform-based singing voice extraction is mainly based on encoder/decoder architectures, with the non-causal adaptation of WaveNet [12, 30] as the only exception. Wave-U-Net [3] is an autoencoder inspired by its spectrogram-based counterpart U-Net [7], while ConvTas-Net [2, 4] estimates prediction masks in the mid-point of the network. The leading model is Demucs, now available in two versions v1 [1] and v2 [2], which is also a convolutional autoencoder with a bidirectional LSTM in the bottleneck. There is a growing tendency on the size of the above-mentioned models, while all use similar, standard training procedures. In contrast, we focus on the training strategy and propose a diffusion-inspired approach for a light-weight model. Building on top of DiffWave, we train a non-causal WaveNet very similar to the one used for music source separation [30], differing only on the number of residual layers, the output projection (since WaveNet in [30] estimates multiple targets while in DiffWave the target is only the added perturbation by the diffusion process), and the additional diffusion step embedding.

Our work has common aspects with [39], in which a novel training strategy is built on top of Demucs (v2) by modelling and learning the dependencies between target sources, and adding an iterative refinement at the output using a Gibbs sampling process. Contrastingly, we use a diffusion-inspired algorithm to train a smaller baseline model for source separation, not to directly estimate the target sources but to gradually transform a mixture signal into its corresponding singing voice.

The literature of diffusion-related approaches for music source separation is scarce. In [25], an improved reverse process based on Langevin dynamics is proposed and applied to several autoregressive models, including WaveNet, which shows competitive performance on separating the vocals from a piano accompaniment. However no experiments on MUSDB18 [40] are reported.

# 4. EXPERIMENTS

## 4.1 Experimental setup

We include the following models in our experiments: **(1)** Singing voice extraction model with different noise schedules: $\beta_{20}$ and $\beta_8$ and $\beta_1$, **(2)** Singing voice extraction model with $\beta_{20}$ and Wiener filtering, using the accompaniment obtained by subtracting the estimated vocals $\hat{v}$ from the input mixture $m$, **(3)** Accompaniment extraction model with $\beta_{100}$, **(4)** Combination of singing voice extraction model with $\beta_{20}$ and accompaniment extraction model with $\beta_{100}$ using Wiener filtering. For the experiments we use ADAM optimizer with learning rate of $2^{-4}$ and batch size of 8. The models are first trained for 200k steps and next, we keep training while evaluating the performance using BSS Eval [41] repeatedly when $\approx 500$ training steps are completed, storing the model that performs the best on the validation set, until 500k steps.

We use the MUSDB18 dataset [40] for training. The accompaniment is computed as the linear sum of the *bass*, *drums*, and *other* sources as represented in the dataset. To train the models we first split the tracks in MUSDB18 in chunks of 4 seconds to optimize the training process and obtain more variate data batches along the training steps. We do not disregard the unvoiced samples, aiming at improving the estimation on vocal silences [28].

For a comparison with the state-of-the-art, our models are evaluated on the test set of MUSDB18, using the standardized metrics for source separation (Signal-to-Distortion Ratio or SDR, Signal-to-Interference Ratio or SIR, and Signal-to-Artifact Ratio or SAR) [42]. We use the BSS Eval implementation and the evaluation setup from the SiSEC challenge [41], using window and hop sizes of 1 second, and subsequently computing the median over all the 1-second estimations of each song. We finally report the median over the entire MUSDB18 testing tracks. We compare our approach with the waveform-based state-of-the-art: WaveNet, Wave-U-Net, ConvTas-Net and Demucs v1 and v2. We report the metrics that the best versions of these methods obtain on the testing set of MUSDB18 [41]. For WaveNet we consider the best performing configuration in [28], and for ConvTas-Net the music source separation version proposed in [2].

Being aware of the possible biases that might occur if training and evaluating on data from the same distribution, even if the splits are properly differentiated, we consider two additional testing sets: (1) A non-MUSDB-overlapping version of MedleyDB [43], in which we remove the 46 overlapping tracks between MedleyDB and MUSDB18 [44] disregarding also the tracks from shared artists between the two even if the track is not present in both, and (2) A manually-cleaned subset of the multi-track audio of the Saraga Carnatic dataset [45] (ground-truth accompaniment is not available). The track list of both datasets is made available in the accompanying repository.

The objective metrics in [41] are not always correlated with the scores from perceptual evaluations of music source separation [46]. However, perceptual tests are

| | | | **Singing Voice** | | | **Accompaniment** | | |
|---|---|---|---|---|---|---|---|---|
| **Model** | **Params** | **Diff. steps** | **SDR** | **SIR** | **SAR** | **SDR** | **SIR** | **SAR** |
| WaveNet [12] (w/ add. loss [38]) | $\approx$ 3.3M | - | 4.49 | 13.52 | 6.17 | 11.39 | 16.37 | 13.49 |
| Wave-U-Net [3] | $\approx$ 10.2M | - | 4.97 | 13.98 | 4.41 | 11.11 | 15.30 | 11.44 |
| ConvTas-Net [2] | $\approx$ 8.75M | - | 6.43 | - | - | - | - | - |
| Demucs (v1) [1] | - | - | 5.44 | - | - | - | - | - |
| Demucs (v2) [2] | $\approx$ 450M | - | 6.84 | - | - | - | - | - |
| Ours (vocal) | $\approx$ 750K | 1 | 4.81 | 9.21 | 8.09 | - | - | - |
| Ours (vocal) | $\approx$ 750K | 8 | 5.63 | 10.55 | 8.86 | - | - | - |
| Ours (vocal) | $\approx$ 750K | 20 | 5.59 | 10.78 | 8.89 | - | - | - |
| Ours (vocal) + Wiener | $\approx$ 750K | 20 | 5.66 | 11.60 | 8.49 | - | - | - |
| Ours (accomp) | $\approx$ 750K | 100 | - | - | - | 11.12 | 13.11 | 16.44 |
| Ours (vocal & accomp) + Wiener | $\approx$ 750K + 750K | 20 + 100 | 6.07 | 12.77 | 8.61 | 11.72 | 14.44 | 16.81 |

**Table 1**. Performance comparison between our model and the waveform-based state-of-the-art. Metrics in dB.

time-consuming and expensive to conduct. We run a perceptual evaluation of the vocal separation for four models: Wavenet (again the best model in [28]), Wave-U-Net (the best model in [3] for monaural separation), Demucs (the v2 model for stereo mixture and 4 sources), and our best model (combining both vocal and accompaniment extraction models using Wiener). We reiterate the experiment in [28] with the same 5 songs (10 second excerpts) and including now our model and Demucs v2. [3]

In this perceptual experiment we follow a double-blind multi-stimulus experimental design with a hidden reference. Similarly to [28], participants are asked to assess the global quality of vocal separation taking into account the suppression of other sources and the lack of distortion, rating the stimuli on scale from 1 to 5, with 1 being *very intrusive interferences from other sources and degraded audio*, and 5 being *unnoticeable interferences from other sources and not degraded audio*. In contrast to [28], the order of the songs is randomized, so that the final rating does not depend on a predefined ordering. In addition, we include the ground-truth vocal stem as a hidden reference along the other stimuli corresponding to the vocal separation of the four models being tested. This hidden reference is used as a control stimuli to filter-out participants that have not performed the training stage, have not understood the task, or do not have sufficient expertise. The participants are asked to calibrate the volume using a tone burst. Then, they perform a training stage where detailed instructions and three audio examples from the same song are presented: the reference mixture, the ground-truth vocal stem and a poor quality separation using a model not included in our test. We use the webMUSHRA framework [47] to implement the experimental design in an online test.

## 4.2 Objective evaluation

In Table 1 we compare the performance of our approach with the waveform-domain state-of-the-art models. No metrics on accompaniment separation and SIR/SAR for vocals are reported for Demucs and ConvTas-Net since these target to *vocals, bass, drums and other*. We observe that our vocal extraction model outperforms WaveNet (note that DiffWave is based on WaveNet), Wave-U-Net, and Demucs v1 in terms of SDR, the latter by a slight difference. Our model provides notable improvement on SAR, which is translated into an output with less artifacts. When combining our vocal and accompaniment extraction models through Wiener filter we obtain closer performance to ConvTas-Net, while Demucs v2 is still leading on SDR $\approx 0.8dB$ above. While iteratively transforming an input mixture, for instance, to the corresponding singing voice, incorrectly estimated accompaniment that is not recognised in the subsequent reverse steps may accumulate in the final prediction. Although the said interferences might not be audible at naked ear, these are penalized by the metrics. The Wiener filtering provides notable improvement on that issue, as especially noted in the SIR and also in the perceptual evaluation in Section 4.3, albeit the source quality is slightly compromised. Finally, note that we use fewer parameters, enhancing the portability and reproducibility of our approach.

The $\beta_1$ singing extraction model, which directly estimates the perturbation by transforming the input mixture in a single run, scores similar than the baseline WaveNet, however we observe a notable decrease of SIR, while SAR improves. This may be given the transformation nature of our approach, which removes the perturbation from the target source instead of directly estimating the source. The $\beta_8$ model scores similar than $\beta_{20}$, however the SIR decreases while SAR is maintained. While adding more steps provides improved interference removal, no notable negative effect is observed in the quality of the estimated source. Nonetheless, if the computational expense is prioritized, the $\beta_8$ model may be used for an optimized inference since the measured performance drop in our experiments is not dramatic. In fact, both models predict faster than real-time on a TITAN Xp GPU. For accompaniment separation, using $\beta_{100}$ provides better predictions. However, as observed for vocals, we may be also capable of modelling this task using less steps, with no significant performance drop.

[3] jordipons.me/apps/end-to-end-music-source-separation

| | | MUSDB18 | | | MedleyDB | | | Saraga Carnatic |
|---|---|---|---|---|---|---|---|---|
| **Model for singing voice** | **Model weight** | **SDR** | **SIR** | **SAR** | **SDR** | **SIR** | **SAR** | **SDR** |
| Ours (vocal, $\beta_{20}$, no Wiener) | $\approx$ 26MB | 5.59 | 10.78 | 8.89 | 4.86 | 8.87 | 9.06 | 4.11 |
| Wave-U-Net [3] | $\approx$ 117MB | 4.97 | 13.98 | 4.41 | 1.61 | 7.47 | 4.50 | 2.13 |
| Demucs (v2) [2] | $\approx$ 1GB | 6.84 | - | - | 6.01 | - | - | 6.12 |

**Table 2**. Performance comparison of our baseline model and state-of-the-art on additional test datasets. Metrics in dB.

In Table 2 we evaluate our baseline $\beta_{20}$ singing voice extraction model (with no post-processing) on the two additional testing datasets presented in Section 4.1. We perform the same evaluation procedure for Wave-U-Net and Demucs v2, comparing how the three models generalize to the testing datasets, using the performance on MUSDB18 – which is also the training dataset for the three – as reference. Our model and Demucs v2 show good generalization to MedleyDB, both getting a similar and small performance drop. Contrastingly, the Wave-U-Net performance is negatively affected in terms of both SDR and SIR. A similar scenario is observed in the Carnatic Music experiment. While Wave-U-Net generalization is again compromised, our model and Demucs v2 are decently able to maintain the performance, the latter being less affected by the change of domain. Similarly to what observed in the MUSDB18 experiment, Demucs v2 predictions include less artifacts, especially in the high-frequency range, being reflected in the metrics as such. Audio examples of this experiment are available in the accompanying repository.

We analyse the behaviour of the $\beta_{20}$ singing voice model along the steps in the reverse process. We observe that the SIR (interf.) notably increases along the steps, at a compromise of a much less steeply SAR (artifacts) decrease. Namely, as we iteratively transform the signal from mixture to singing voice, we remove the accompaniment while trying to maintain the quality of the singing voice, relying on the model trained with our diffusion-inspired strategy to estimate the perturbation at each step while alleviating the additional interferences incorrectly generated during the reverse process. We note that given the parametrization of the reverse process, stronger transformation is performed in the first steps (1 to 5 for $\beta_{20}$), while the rest of the steps refine the final estimation. For that reason, fair or good performance – relatively to the overall track difficulty – on the first step normally leads to enhanced final output, while bad initial performance may even be further degraded through the reverse process.

### 4.3 Perceptual evaluation

In total 40 people participated in our experiment, 4 of them being excluded because they scored the ground-truth stimuli lower than a separation stimuli. We compute Mean Opinion Score (MOS) by averaging the ratings for all songs and all participants. The results in terms of MOS are presented in Figure 2. Note that Ground-truth is not reaching 5, meaning that the test includes difficult cases with distorted vocals or large unvoiced segments. We observe that the 95% Confidence Interval for our model is very similar to Demucs and notably higher than both Wave-U-Net and especially WaveNet, a very similar architecture to the instance we have trained using our diffusion-inspired strategy. Such test suggests that the predictions made by our approach may include artifacts or interferences that affect negatively the standardized metrics but are not perceivable at naked ear. This test may be extended in future to separately study the perceivable distortion and interference.

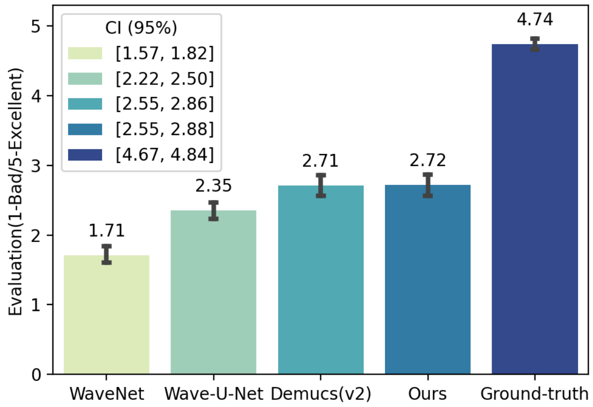


**Figure 2**. Perceptual evaluation report for the waveform-based state-of-the-art models and ours

## 5. CONCLUSIONS

In this paper we leverage from the denoising diffusion algorithm to propose a training and sampling strategy for singing voice extraction. The model trained using our approach learns to gradually transform a mixture into its corresponding vocal source or accompaniment, achieving comparable performance to the waveform-based state-of-the-art on MUSDB18. In addition, we evaluate how our approach generalizes to other testing sets, showing decent generalization to these out-of-domain data. We also run a perceptual test in which our approach scores similar than Demucs v2 and outperforms the others. Our approach operates on an architecture similar to WaveNet and obtains better objective and perceptual evaluation. This work has a broad future outlook. For the next steps, we look at separating other sources and supporting stereo. We also look at extending the approach, for instance, by using a different or improved network to learn the reverse process, focusing on U-Nets which are the leading architecture music source separation. We may also consider conditioning, the key element of diffusion-based approaches in the literature. Finally, our diffusion-inspired reverse parametrization may be further improved to better refine the predictions.

## 6. ACKNOWLEDGEMENTS

This work was carried out under the projects Musical AI - PID2019-111403GB-I00/AEI/10.13039/501100011033 and NextCore - RTC2019-007248-7 funded by the Spanish Ministerio de Ciencia, Innovación y Universidades (MCIU) and the Agencia Estatal de Investigación (AEI). We would like to thank the 40 participants that took the perceptual test for this work.

## 7. REFERENCES

[1] A. Défossez, N. Usunier, L. Bottou, and F. Bach, "Demucs: Deep Extractor for Music Sources with extra unlabeled data remixed," 2019. [Online]. Available: http://arxiv.org/abs/1909.01174

[2] A. Défossez, N. Usunier, L. Bottou, and F. Bach, "Music Source Separation in the Waveform Domain," 2019. [Online]. Available: http://arxiv.org/abs/1911.13254

[3] D. Stoller, S. Ewert, and S. Dixon, "Wave-U-Net: A multi-scale neural network for end-to-end audio source separation," *In Proc. of the 19th Int. Society for Music Information Retrieval Conf. (ISMIR),* Paris, France, pp. 334–340, 2018.

[4] Y. Luo and N. Mesgarani, "Conv-TasNet: Surpassing Ideal Time-Frequency Magnitude Masking for Speech Separation," *IEEE/ACM Transactions on Audio Speech and Language Processing*, vol. 27, no. 8, pp. 1256–1266, 2019.

[5] N. Takahashi and Y. Mitsufuji, "D3Net: Densely connected multidilated DenseNet for music source separation," 2020. [Online]. Available: http://arxiv.org/abs/2010.01733

[6] T. Li, J. Chen, H. Hou, and M. Li, "Sams-Net: A Sliced Attention-based Neural Network for Music Source Separation," *In Proc. on the 12th Int. Symposium on Chinese Spoken Language Processing (ISCSLP)*, 2021.

[7] A. Jansson, E. Humphrey, N. Montecchio, R. Bittner, A. Kumar, and T. Weyde, "Singing voice separation with deep U-Net convolutional networks," *In Proc. of the 18th Int. Society for Music Information Retrieval Conf. (ISMIR),* Suzhou, China, pp. 745–751, 2017.

[8] A. Défossez, "Hybrid spectrogram and waveform source separation," 2021. [Online]. Available: http://arxiv.org/abs/2111.03600

[9] M. Kim, W. Choi, J. Chung, D. Lee, and S. Jung, "KUIELab-MDX-Net: A Two-Stream Neural Network for Music Demixing," 2021. [Online]. Available: http://arxiv.org/abs/2111.12203

[10] C.-Y. Yu and K.-W. Cheuk, "Danna-Sep: Unite to separate them all," 2021. [Online]. Available: http://arxiv.org/abs/2112.03752

[11] Y. Mitsufuji, G. Fabbro, S. Uhlich, F.-R. Stöter, A. Défossez, M. Kim, W. Choi, C.-Y. Yu, and K.-W. Cheuk, "Music Demixing Challenge 2021," *Frontiers in Signal Processing*, no. January, 2022.

[12] D. Rethage, J. Pons, and X. Serra, "A wavenet for speech denoising," *In Proc. of the IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP),* Paris, France, vol. 2018-April, pp. 5069–5073, 2018.

[13] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," *In Proc. of the 33th Advances in Neural Information Processing Systems (NeurIPS),* Online, pp. 6840–6851, 2020.

[14] C. Jarzynski, "Equilibrium free-energy differences from nonequilibrium measurements: A master-equation approach," *Physical Review*, 1997.

[15] J. Zhang, S. Jayasuriya, and V. Berisha, "Restoring degraded speech via a modified diffusion model," *In Proc. of the Annual Conf. of the Int. Speech Communication Association (INTERSPEECH),* Online, vol. 4, pp. 2753–2757, 2021.

[16] Z. Kong, W. Ping, J. Huang, K. Zhao, and B. Catanzaro, "DiffWave: A Versatile Diffusion Model for Audio Synthesis," *In Proc. of the 9th Int. Conf. on Learning Representations (ICLR),* Vienna, Austria, 2021.

[17] J. Sohl-Dickstein, E. A. Weiss, N. Maheswaranathan, and S. Ganguli, "Deep unsupervised learning using nonequilibrium thermodynamics," *In Proc. of the 32nd Int. Conf. on Machine Learning (ICML),* Lille, France, vol. 3, pp. 2246–2255, 2015.

[18] N. Chen, Y. Zhang, H. Zen, R. J. Weiss, M. Norouzi, and W. Chan, "WaveGrad: Estimating Gradients for Waveform Generation," *In Proc. of the 9th Int. Conf. on Learning Representations (ICLR),* Vienna, Austria, 2021.

[19] Y.-J. Lu, Z.-Q. Wang, S. Watanabe, A. Richard, C. Yu, and Y. Tsao, "Conditional Diffusion Probabilistic Model for Speech Enhancement," *In Proc. of the IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP),* Singapore, 2022.

[20] Y. J. Lu, Y. Tsao, and S. Watanabe, "A Study on Speech Enhancement Based on Diffusion Probabilistic Model," *In Proc. of the Asia-Pacific Signal and Information Processing Association Annual Summit and Conf. (APSIPA ASC),* Online, pp. 659–666, 2021.

[21] J. Serrà, S. Pascual, J. Pons, R. O. Araz, and D. Scaini, "Universal speech enhancement with score-based diffusion," 6 2022. [Online]. Available: http://arxiv.org/abs/2206.03065

[22] J. Lee and S. Han, "NU-wave: A diffusion probabilistic model for neural audio upsampling," *In Proc. of the Annual Conf. of the Int. Speech Communication Association (INTERSPEECH),* Brno, Czech Republic, vol. 4, no. 3, pp. 2698–2702, 2021.

[23] J. Liu, C. Li, Y. Ren, F. Chen, and Z. Zhao, "DiffSinger: Singing Voice Synthesis via Shallow Diffusion Mechanism," *In Proc. of the 36th Conf. on Artificial Intelligence (AAAI),* Online, 2022.

[24] G. Mittal, J. Engel, C. Hawthorne, and I. Simon, "Symbolic Music Generation with Diffusion Models," *In Proc. of the 22th Int. Society for Music Information Retrieval Conf. (ISMIR),* Online, pp. 468–475, 2021.

[25] V. Jayaram and J. Thickstun, "Parallel and Flexible Sampling from Autoregressive Models via Langevin Dynamics," *In Proc. of the Int. Conf. on Learning Representations (ICLR),* Online, 2021.

[26] A. Bansal, E. Borgnia, H.-M. Chu, J. S. Li, H. Kazemi, F. Huang, M. Goldblum, J. Geiping, and T. Goldstein, "Cold diffusion: Inverting arbitrary image transforms without noise," 8 2022. [Online]. Available: http://arxiv.org/abs/2208.09392

[27] J. L. Durrieu, G. Richard, B. David, and C. Févotte, "Source/filter model for unsupervised main melody extraction from polyphonic audio signals," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, no. 3, 2010.

[28] F. Lluís, J. Pons, and X. Serra, "End-to-end music source separation: Is it possible in the waveform domain?" *In Proc. of the Annual Conf. of the Int. Speech Communication Association (INTERSPEECH),* Graz, Austria, vol. 2019-September, pp. 4619–4623, 2019.

[29] D. P. Kingma, T. Salimans, B. Poole, and J. Ho, "Variational Diffusion Models," *In Proc. of the 35th Conf. on Neural Information Processing Systems (NeurIPS 2021),* Online, pp. 21 696–21 707, 2021.

[30] A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu, "WaveNet: A Generative Model for Raw Audio," 2016. [Online]. Available: http://arxiv.org/abs/1609.03499

[31] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in Neural Information Processing Systems (NeurIPS),* Long Beach, USA, vol. 2017-December, pp. 5999–6009, 2017.

[32] W. Ping, K. Peng, A. Gibiansky, S. O. Arik, A. Kannan, S. Narang, J. Raiman, and J. Miller, "Deep Voice 3: Scaling text-to-speech with convolutional sequence learning," *In Proc. of the Int. Conf. on Learning Representations (ICLR),* Vancouver, Canada, 2018.

[33] S. O. Arik, G. Diamos, A. Gibiansky, J. Miller, K. Peng, W. Ping, J. Raiman, and Y. Zhou, "Deep Voice 2: Multi-speaker neural text-to-speech," *In Proc. of the Conf. on Neural Information Processing Systems (NeurIPS),* Long Beach, USA, 2017.

[34] P. Chandna, M. Blaauw, J. Bonada, and E. Gomez, "A Vocoder Based Method for Singing Voice Extraction," *In Proc. of the IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP),* Brighton and Hove, UK, vol. 2019-May, pp. 990–994, 2019.

[35] A. Nugraha, A. Liutkus, and E. Vincent, "Multichannel audio source separation with deep neural networks," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 24, no. 9, pp. 1652–1664, 2016.

[36] A. Liutkus and F. R. Stöter, "sigsep/norbert," 2019. [Online]. Available: https://doi.org/10.5281/zenodo.3269749

[37] N. Duong, E. Vincent, and R. Gribonval, "Underdetermined reverberant audio source separation using a full-rank spatial covariance model," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, no. 7, pp. 1830–1840, 2010.

[38] P. Huang, M. Kim, M. Hasegawa-Johnson, and P. Smaragdis, "Singing-voice separation from monaural recordings using deep recurrent neural networks," *In Proc. of the 15th Int. Society for Music Information Retrieval Conf. (ISMIR),* Taipei, Taiwan, pp. 477–482, 2014.

[39] E. Manilow, C. Hawthorne, C.-Z. Huang, B. Pardo, and J. Engel, "Improving Source Separation by Explicitly Modeling Dependencies Between Sources," *In Proc. of the IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP),* Singapore, 2022.

[40] Z. Rafii, A. Liutkus, F.-R. Stöter, S. I. Mimilakis, and R. Bittner, "MUSDB18 - a corpus for music separation," Dec. 2017. [Online]. Available: https://doi.org/10.5281/zenodo.1117372

[41] F. R. Stöter, A. Liutkus, and N. Ito, "The 2018 Signal Separation Evaluation Campaign," *Lecture Notes in Computer Science*, vol. 10891, pp. 293–305, 2018.

[42] E. Vincent, R. Gribonval, and C. Févotte, "Performance measurement in blind audio source separation," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 4, no. 14, pp. 1462–1469, 2006.

[43] R. Bittner, J. Salamon, M. Tierney, M. Mauch, C. Cannam, and J. Bello, "MedleyDB: A multitrack dataset for annotation-intensive MIR research," *In Proc. of the 15th Int. Society for Music Information Retrieval Conf. (ISMIR),* Taipei, Taiwan, pp. 155–160, 2014.

[44] "SigSep MUSDB Website," https://sigsep.github.io/datasets/musdb.html, accessed: 01-05-2022.

[45] A. Srinivasamurthy, S. Gulati, R. C. Repetto, and X. Serra, "Saraga: Open Datasets for Research on Indian Art Music," *Empirical Musicology Review*, 2020.

[46] E. Cano, D. Fitzgerald, and K. Brandenburg, “Evaluation of quality of sound source separation algorithms: Human perception vs quantitative metrics,” *In Proc. of the 24th European Signal Processing Conf. (EUSIPCO),* Budapest, Hungary, pp. 1758–1762, 2016.

[47] M. Schoeffler, S. Bartoschek, F.-R. Stöter, M. Roess, S. Westphal, B. Edler, and J. Herre, “webmushra—a comprehensive framework for web-based listening tests,” *Journal of Open Research Software*, vol. 6, no. 1, 2018.

# A MODEL YOU CAN HEAR: AUDIO IDENTIFICATION WITH PLAYABLE PROTOTYPES

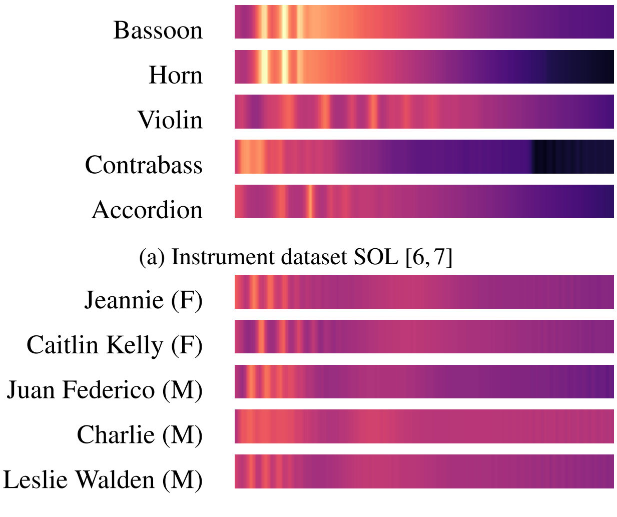
**Romain Loiseau**[1,2] **Baptiste Bouvier**[3] **Yann Teytaut**[3]
**Elliot Vincent**[1,4] **Mathieu Aubry**[1] **Loic Landrieu**[2]
[1] LIGM, Ecole des Ponts, Univ Gustave Eiffel, CNRS, France
[2] LASTIG, Univ Gustave Eiffel, IGN, ENSG
[3] STMS Lab, UMR 9912 (IRCAM, CNRS, Sorbonne University), Paris, 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 Transformation 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.



(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.

## 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-

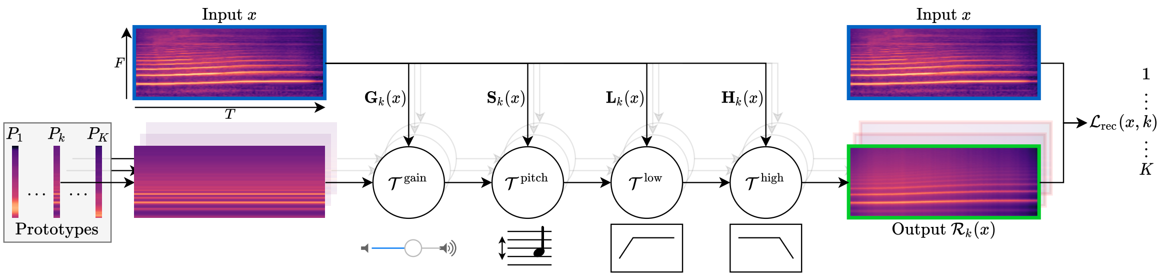

Input $x$
$F$
$T$
$P_1$
$P_k$
$P_K$
Prototypes
$\mathbf{G}_k(x)$
$\mathbf{S}_k(x)$
$\mathbf{L}_k(x)$
$\mathbf{H}_k(x)$
$\mathcal{T}^{\text{gain}}$
$\mathcal{T}^{\text{pitch}}$
$\mathcal{T}^{\text{low}}$
$\mathcal{T}^{\text{high}}$
Input $x$
Output $\mathcal{R}_k(x)$
$\mathcal{L}_{\text{rec}}(x, k)$
1
$K$

**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 $s$.