# Deep Audio Prior: Learning Sound Source Separation from a Single Audio Mixture

Yapeng Tian[1], Chenliang Xu[1], and Dingzeyu Li[2]
[1]University of Rochester [2]Adobe Research

## 1. Introduction

Typically, a deep neural network distills robust priors from a large amount of labeled or unlabeled data. In audio research, deep networks such as VGGish also benefit from large datasets like AudioSet [3].

We are interested in learning to separate sources from a single audio mixture, without relying on any additional training data. We took inspiration from auditory perception research. When one listens to a voice in the presence of other voices, *auditory coherence* provides a perceptual continuity and temporal correlation. To separate simultaneous speech messages, humans rely on the *discontinuity* in different voices, as been studied in perceptual research [1].

We propose Deep Audio Prior (DAP) and leverage the power of deep networks to encode *auditory coherence* and *discontinuity*. DAP's capability to train on a single audio mixture has several advantages. First, bypassing the inherent dataset bias, DAP generalizes well to a wide variety of unseen data. Second, our training process is fully unsupervised and therefore possible to pre-process large volumes of data in the wild. Last but not least, we show several novel applications that are only possible because of the unique features of DAP, including universal source separation, interactive editing, audio texture synthesis, and audio co-separation. Both quantitative and qualitative results demonstrate the effectiveness and potential of our DAP.

Domain gap between *audio* and *visual images* precludes direct adoption of the deep image priors [8, 2]. Many assumptions that are true for images no longer hold for audio. By nature, audio signals exhibit strong temporal coherence, e.g., one's voice changes smoothly (See $A \rightarrow B$ in Fig. 1). Since natural images tend to exhibit more spatial patterns, most existing deep image priors have focused on how to encapsulate the spatial redundancy without temporal coherence. Another challenge specific to audio is the activation discontinuity. Unlike in videos where an object moves continuously in the scene, a sound source can make sound or turn silence at any given time (see $A \rightarrow C$ in Fig. 1).

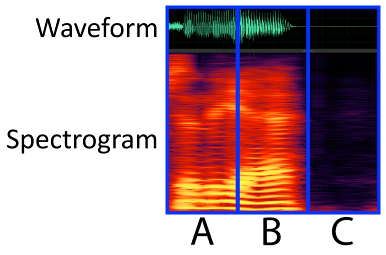


Figure 1: Coherence and discontinuity in audio: the domain gap between audio and visual images requires novel deep priors that respect these inherent auditory properties.

The proposed DAP framework shows the powerful priors embedded in the deep network structure. Specifically, we demonstrate the following features of DAP.

**Auditory Coherence.** DAP's temporally coherent source generator can reproduce a wide spectrum of natural sound sources including human speech, musical instruments, etc.

**Discontinuity.** DAP's mask activation scheme can enable and disable sources without temporal dependence, addressing the source discontinuity.

**Challenging Applications.** Without any prior training or external dataset, DAP is effective at many audio tasks, including universal audio source separation, interactive mask-based editing, audio texture synthesis, and co-separation.

**Dataset/Code.** We introduce Universal-150, a benchmark dataset on blind source separation for universal sound sources. DAP outperforms other blind source separation (BSS) methods in both numerical metrics and qualitative comparisons. Code and dataset are released in https://github.com/adobe/Deep-Audio-Prior.

## 2. Deep Audio Prior Separation Framework

DAP is an unsupervised blind source separation framework. More specifically, DAP does not train on any extra data other than the input audio mixture. Similar to image foreground and background segmentation, audio blind two-source separation can be expressed as:

$$S_{\text{mixture}} = Source_1 \odot Mask_1 + Source_2 \odot Mask_2 \quad (1)$$

The goal of DAP is to predict all 4 unknown variables using deep networks from the only input audio mixture:

$$S_{\text{mixture}} = S_1(z_1) \odot M_1(g_1) + S_2(z_2) \odot M_2(g_1) \quad (2)$$

where $S_1$ and $S_2$ are two audio generator networks, $M_1$ and $M_2$ are two mask networks, $z_1, z_2$ and $g_1, g_2$ are latent variables sampled from random distributions. Our method works in Time-Frequency (T-F) spectrogram space. All the input and output variables are of dimension $F \times T$. Figure 2 shows an overview of our framework.

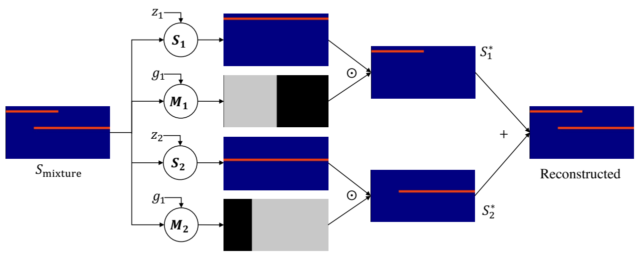


Figure 2: DAP framework illustrated with a synthetic sound mixture spectrogram. DAP learns to separate sounds from a single input sound mixture: $S_{\text{mixture}}$ without requiring any external training data. With random noises as inputs, we use two sound prediction networks: $S_1$ and $S_2$ and two mask modulation networks: $M_1$ and $M_2$ to perform source separation. The four networks share the same U-Net structure.

The intuition for our single-audio source separation is inspired by single-image decomposition: it is much easier for two generator networks ($S_1$ and $S_2$) to learn two distinct sound sources respectively, rather than forcing one of the generators to learn the mixture. [2] analyzed this in terms of complex mixture versus simple individual components.

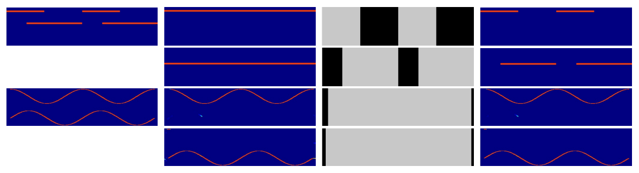
**Reconstruction Loss** Let $S_1^* = S_1(z_1) \odot M_1(g_1)$ and $S_2^* = S_2(z_2) \odot M_2(g_2)$, then we have $S_{\text{mixture}}^* = S_1^* + S_2^*$. Clearly, when combining the separated sounds $S_1^*$ and $S_2^*$, we should obtain the original sound mixture $S_{\text{mixture}}$. The data fidelity term is expressed as a reconstruction loss, $l_{\text{rec}} = ||S_{\text{mixture}} - S_1^* - S_2^*||_2$, which will push the combination of the two separated sounds to be close to the original sound mixture.

## 2.1. Temporally Coherent Generator

**Coherence** Audio signals from the same source across time would be similar. To explicitly model the temporal property, we use multiple audio frames with temporal consistent noise as inputs for each sound source. We split the $S_{\text{mixture}}$ into $N$ frames along the time axis and obtain $\{S_{\text{mixture}}^i\}_{i=1}^N$. Accordingly, we use $n$ noise input pairs $\{z_1^i, z_2^i\}_{i=1}^N$ to predict the corresponding sounds $\{S_1(z_1^i), S_2(z_2^i)\}_{i=1}^N$. For the rest of this paper, we will omit the underscript when there is no confusion, i.e., $z^i$ instead of $z_1^i$. To explicitly enforce the temporal coherence, we impose strong correlations on input noises:

$$z^1 = n^* \quad \text{and} \quad z^i = z^{i-1} + \Delta u^i \tag{3}$$

where $\Delta u^i$ is a random noise sampled from an uniform distribution with a variance significantly lower than that of Gaussian noise $n^*$ initialization of $z^1$. Since we use shared networks $S_1$ and $S_2$ across different frames for predicting individual sounds and the noise inputs are temporally consistent, which will enforce the network to predict temporal consistent sounds. A similar idea was also adopted in [2] to



(a) Mixture (b) $S_1$ and $S_2$ (c) $M_1$ and $M_2$ (d) $S_1^*$ and $S_2^*$

Figure 3: Synthetic tests: two single-tone sources (top two rows) and two cosine-tone sources (bottom two rows). Given a single input mixture, DAP achieves perfect separation on both inputs.

preserve video coherence but they use the noise to predict masks. We also employ a temporal continuity loss function to further enforce it by pushing the absolute of gradients along time to be small: $l_{\text{tc}} = \sum_i \sum_p \sum_q |S_i(p,q) - S_i(p,q-1)|$.

**Pitch Exclusion** We assume that two different sounds $S_1^*$ and $S_2^*$ are dissimilar. To enforce the constraint, we utilize an exclusion loss function $l_{\text{ex}}$ from [10] in which it is formulated as the product of normalized gradient fields of the two sound predictions.

## 2.2. Discontinuity-Aware Masks for Audio

Sound sources will not always make sounds all the time, which would break the coherence in the time domain. To address this issue, we predict audio masks to activate sounding spectrum regions and deactivate silent regions.

If a source is sounding at a time, the spectrum bin at the time should be activated. Based on the observation, mask values within the same temporal bin should be consistent and binary. Namely, if a dog barking sound is present, it should appear across all frequency ranges under the same timestamp. Therefore, we force the mask within the same temporal bin to be consistent: $M_i(0,q) = M_i(1,q) = ... = M_i(F-1,q) = m_i^q$, where $i$ refers to $i$-th sound source, $q \in \{0, 1..., T\}$, $F$ and $T$ are height (frequency axis) and width (time axis) of the spectrogram $M$. In our implementation, we use a max pooling operator to aggregate the mask content along frequency-coordinate and generate mask value $m_i^q$ using a Sigmoid function. $m_i^q$ represents the mask for $i$-th source at time $q$. We define $m_i^q$ to be a continuous variable for ease of optimization and will enforce an extra binary loss term.

**Non-Zero Masks** Furthermore, at any time, if there is a sound in $S_{\text{mixture}}$, at least one of the masks should be activated. To this end, we introduce a nonzero mask loss term: $l_{\text{nonzero}} = \sum_q w_q \cdot (\epsilon + \min(\sigma, \sum_i m_i^q))^{-1}$, where $w_q = \sum_p \log(1 + S_{\text{mixture}}(p,q))$, $\epsilon$ = $10^{-6}$ for numerical stability and $\sigma = 1$ is a margin value. The margin value is to ensure when the sum of mask activation is already larger than the $\sigma$, the loss will not continue to push the mask values. $w^q$ is to suppress this loss term if there is no sound

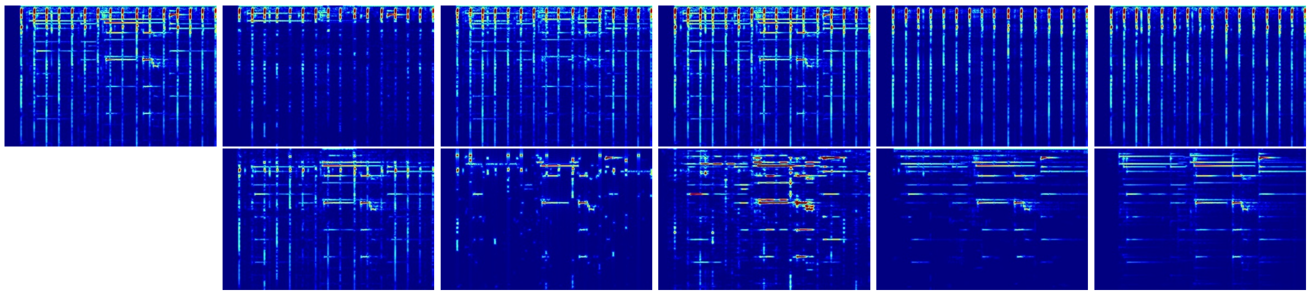


Figure 4: Comparison on a sample audio from Universal-150 benchmark.

in $S_{\text{mixture}}$ at time $q$. Moreover, in our observations, we found that masks are either fully activated or fully deactivated. Very rarely would a sound source be at $50\%$ activation level. Therefore, we introduce a differential loss term to encourage the mask networks to generate binary masks: $l_{\text{binary}} = \sum_i (\epsilon + \sum_{p,q} |M_i(p,q) - 0.5|)^{-1}$, where again $\epsilon = 10^{-6}$ is used to avoid numerical issues. The $l_{\text{binary}}$ will force the mask values to be as far away as possible to $0.5$, which means they are close to either $0$ or $1$.

## 2.3. Walkthrough on Synthetic Separation

With all the pieces together, our total loss is defined as: $l_{\text{total}} = l_{\text{rec}} + l_{\text{tc}} + l_{\text{ex}} + l_{\text{nonzero}} + l_{\text{binary}}$. We optimize the networks in an end-to-end manner with all the loss functions. We empirically keep the weights for all loss terms the same, with the only exception being $l_{\text{binary}}$ has a $0.01$ factor. To validate the implementation of our algorithm, we generate two types of synthetic spectrograms.

**(i) Single-frequency synthetic sounds** We generate a synthetic mixture where each of the two audio sources is producing a flat tone at a single frequency. For this test, since we know the spectrogram at different segments will always produce the same output (each sound source is just a flat bar), we set the input noise $z_1^1 = z_1^2 = \cdots = z_1^n$ for sound source 1 and the same for source 2. The top two rows from Figure 3 show that our DAP achieves perfect prediction on both the source generators and masks.

**(ii) Curved input sounds** Our next test is to validate if our DAP framework can handle temporally coherent spectrogram changes. Two shifted cosine curves are combined together as input to our separation pipeline (see bottom two rows in Figure 3). We set the input noise according to equation (3). Again, our DAP framework achieves the desired separation of sources and masks.

## 3. DAP Applications

The same DAP idea that works on the synthetic example naturally extends to real-world audio spectrograms. Next, we present several challenging applications with DAP.

**Universal Blind Source Separation** Given a sound mixture $S_{\text{mixture}}$, universal blind source separation aims to separate individual sounds from the sound mixture without using any external data. Universal blind separation is challenging because the input audio can be in an arbitrary domain, not just commonly studied speech or music domains. With traditional separation models, the first step is to learn a strong prior from a large audio dataset. However, in our case, we only ran the training step on this 6-second input audio and learned the inherent source generators that match the ground truth reference almost perfectly.

Table 1: Comparison between our DAP/NMF/RPCA/KAM on numerical metrics: SDR/SIR/ASD. For SDR/SIR, higher is better and for ASD, smaller is better.

| | NMF | RPCA | KAM | DAP |
|---|---|---|---|---|
| SDR | -6.37 | -5.40 | -3.38 | **-1.58** |
| SIR | -1.93 | -0.23 | -0.75 | **3.83** |
| ASD | 2.30 | 2.01 | 2.32 | **1.96** |

We build an *universe audio source separation dataset* that contains 150 audio mixtures and each sample is a two-sound mixture. These mixture samples come from pairs of 30 6-second long unique sounds from YouTube and ESC-50 sound classification dataset [6] covering a large range of sound categories appeared in our daily life. We compare our method with several blind source separation (BSS) methods: non-negative matrix factorization (NMF) [7], robust principal component analysis (RPCA) [4], and kernel additive modelling (KAM) [9]. Figure 4 shows that our method outperforms the compared methods qualitatively. DAP is the only method that can successfully separate the two complex sound sources, closely matching ground truth reference. Moreover, for quantitative evaluation, we run separation on 150 sounds and compare these methods in three metrics: Signal-to-Distortion Ratio (SDR), Signal-to-Interference Ratio (SIR), and Audio Spectrum Distance (ASD) [5]. DAP achieves the best performance in three numerical metrics, as shown in Table 1.

**Interactive Mask-based Editing** Since the output of our method contains both a mask and the sound generator, we can add additional constraints on either the mask or the sound spectrogram. Fig. 5 shows the simple 1D box given

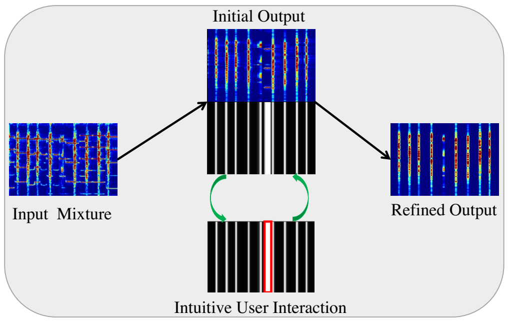


Figure 5: Example of interactive DAP. Given a sound mixture, our DAP model can predict separated sounds and the corresponding mask activation maps. Users can simply draw boxes to interact with our DAP model to tell where to deactivate or activate the predicted masks for refining predicted sounds.

by the user can quickly improve the results. These 1D box constraints can be easily drawn by users that are not familiar with frequency spectrograms. Basically they select the regions they think should have sound or should be silence for source $k$. These selected regions would become input activation masks: $M_k^{\text{act}}$ or deactivate mask: $M_k^{\text{deact}}$, respectively. The mask values in the annotated regions are one. To impose these annotated constraints, we introduce mask activation and deactivation losses to refine results from our separation networks: $l_{\text{act}} = \sum_k M_k^{\text{act}} \| M_k - M_k^{\text{act}} \|_1$ and $l_{\text{deact}} = \sum_k M_k^{\text{deact}} \| M_k - (1 - M_k^{\text{deact}}) \|_1$. With the activation and deactivation losses, our DAP refines the source generator *and* masks. This refinement process typically takes a few seconds. We can see that with adding a mask deactivation loss for the "dog" sound into optimization, our network will remove the "violin" patterns from it.

**Co-separation / Audio Watermark Removal** Audio watermarks are commonly used in the music industry for copyright-protected audios. Supervised deep networks will require a lot of training data to learn both diverse clean audio patterns and watermark patterns for separating watermark sounds from clean sounds. Our DAP model can also easily generalize to handle co-separation for removing audio watermarks. Given $K$ sounds: $S_k = S_k^* + S_{\text{watermark}}$ $(k = 1, 2, ..., K)$ containing the mixture of the same unknown audio watermark $S_{\text{watermark}}$ and the clean audios: $S_1^*, S_2^*, ..., S_K^*$. The goal is to recover clean audios. Using the proposed separation framework, we learn $K + 1$ generator networks; $K$ for the music signals and 1 for the watermark with shared network weights and input noise. As shown in Figure 6, from the $K = 3$ input mixtures, we can extract the clean music as well as the embedded watermark.

**Acknowledgement:** Work done during YT's internship at Adobe Research. The work was partly supported by NSF 1741472, 1813709, and 1909912. We gratefully acknowledge the gift donations of Adobe. This article solely reflects the opinions and conclusions of its authors and neither NSF, nor Adobe. We would like to thank Nicholas Bryan for discussing blind source separation methods and Wilmot Li for brainstorming application scenarios.

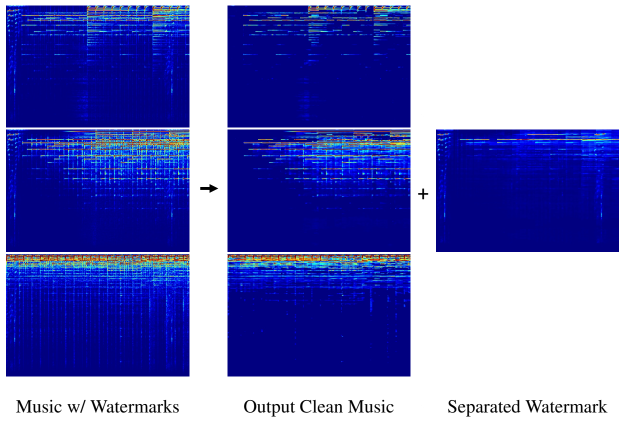


Figure 6: DAP can also be applied to automatically remove audio watermark. Given these 3 mixtures, DAP can separate each individual sound source out. Note that DAP is only trained on these 3 input mixtures.

## References

[1] J. P. L. Brokx and S. G. Nooteboom. Intonation and the perceptual separation of simultaneous voices. *Journal of Phonetics*, 10(1):23–36, 1982. 1

[2] Y. Gandelsman, A. Shocher, and M. Irani. "double-dip": Unsupervised image decomposition via coupled deep-image-priors. In *CVPR*, 2019. 1, 2

[3] J. F. Gemmeke, D. P. Ellis, D. Freedman, A. Jansen, W. Lawrence, R. C. Moore, M. Plakal, and M. Ritter. Audio set: An ontology and human-labeled dataset for audio events. In *ICASSP*, pages 776–780. IEEE, 2017. 1

[4] P.-S. Huang, S. D. Chen, P. Smaragdis, and M. Hasegawa-Johnson. Singing-voice separation from monaural recordings using robust principal component analysis. In *ICASSP*, 2012. 3

[5] P. Morgado, N. Vasconcelos, T. Langlois, and O. Wang. Self-supervised generation of spatial audio for 360 video. In *Neural Information Processing Systems (NIPS)*, 2018. 3

[6] K. J. Piczak. Esc: Dataset for environmental sound classification. In *ACM MM*, 2015. 3

[7] M. Spiertz and V. Gnann. Source-filter based clustering for monaural blind source separation. In *Proceedings of the 12th International Conference on Digital Audio Effects*, 2009. 3

[8] D. Ulyanov, A. Vedaldi, and V. Lempitsky. Deep image prior. In *CVPR*, 2018. 1

[9] D. F. Yela, D. Stowell, and M. Sandler. Does k matter? k-nn hubness analysis for kernel additive modelling vocal separation. In *International Conference on Latent Variable Analysis and Signal Separation*, pages 280–289. Springer, 2018. 3

[10] X. Zhang, R. Ng, and Q. Chen. Single image reflection separation with perceptual losses. In *CVPR*, 2018. 2