NEURAL SEPARATION OF OBSERVED AND UNOBSERVED DISTRIBUTIONS

Anonymous authors
Paper under double-blind review

ABSTRACT

Separating mixed distributions is a long standing challenge for machine learning and signal processing. Applications include: single-channel multi-speaker separation (cocktail party problem), singing voice separation and separating reflections from images. Most current methods either rely on making strong assumptions on the source distributions (e.g. sparsity, low rank, repetitiveness) or rely on having training samples of each source in the mixture. In this work, we tackle the scenario of extracting an unobserved distribution additively mixed with a signal from an observed (arbitrary) distribution. We introduce a new method: Neural Egg Separation - an iterative method that learns to separate the known distribution from progressively finer estimates of the unknown distribution. In some settings, Neural Egg Separation is initialization sensitive, we therefore introduce GLO Masking which ensures a good initialization. Extensive experiments show that our method outperforms current methods that use the same level of supervision and often achieves similar performance to full supervision.

1 INTRODUCTION

Humans are remarkably good at separating data coming from a mixture of distributions, e.g. hearing a person speaking in a crowded cocktail party. Artificial intelligence, on the the hand, is far less adept at separating mixed signals. This is an important ability as signals in nature are typically mixed, e.g. speakers are often mixed with other speakers or environmental sounds, objects in images are typically seen along other objects as well as the background. Understanding mixed signals is harder than understanding pure sources, making source separation an important research topic.

Mixed signal separation appears in many scenarios corresponding to different degrees of supervision. Most previous work focused on the following settings:

Full supervision: The learner has access to a training set including samples of mixed signals $\{y_i\} \in \mathcal{Y}$ as well as the ground truth sources of the same signals $\{b_i\} \in \mathcal{B}$ and $\{x_i\} \in \mathcal{X}$ (such that $y_i = x_i + b_i$). Having such strong supervision is very potent, allowing the learner to directly learn a mapping from the mixed signal y_i to its sources (x_i, b_i). Obtaining such strong supervision is typically unrealistic, as it requires manual separation of mixed signals. Consider for example a musical performance, humans are often able to separate out the different sounds of the individual instruments, despite never having heard them play in isolation. The fully supervised setting does not allow the clean extraction of signals that cannot be observed in isolation e.g. music of a street performer, car engine noises or reflections in shop windows.

Synthetic full supervision: The learner has access to a training set containing samples from the mixed signal $\{y_i\} \in \mathcal{Y}$ as well as samples from all source distributions $\{b_j\} \in \mathcal{B}$ and $\{x_k\} \in \mathcal{X}$. The learner however does not have access to paired sets of the mixed and unmixed signal ground truth (that is for any given y_i in the training set, b_i and x_i are unknown). This supervision setting is more realistic than the fully supervised case, and occurs when each of the source distributions can be sampled in its pure form (e.g. we can record a violin and piano separately in a studio and can thus obtain unmixed samples of each of their distributions). It is typically solved by learning to separate synthetic mixtures $b_j + x_k$ of randomly sampled b_j and x_k.

No supervision: The learner only has access to training samples of the mixed signal $\mathcal{Y}$ but not to sources $\mathcal{B}$ and $\mathcal{X}$. Although this settings puts the least requirements on the training dataset, it is a hard problem and can be poorly specified in the absence of strong assumptions and priors. It is generally necessary to make strong assumptions on the properties of the component signals (e.g. smoothness, low rank, periodicity) in order to make progress in separation. This unfortunately severely limits the applicability of such methods.

In this work we concentrate on the *semi-supervised* setting: unmixing of signals in the case where the mixture $\mathcal{Y}$ consists of a signal coming from an unobserved distribution $\mathcal{X}$ and another signal from an observed distribution $\mathcal{B}$ (i.e. the learner has access to a training set of clean samples such that $\{b_j\} \in \mathcal{B}$ along with different mixed samples $\{y_i\} \in \mathcal{Y}$). One possible way of obtaining such supervision, is to label every signal sample by a label, indicating if the sample comes only from the observed distribution $\mathcal{B}$ or if it is a mixture of both distributions $\mathcal{B} + \mathcal{X}$. The task is to learn a parametric function able to separate the mixed signal $y_i \in \mathcal{Y}$ into sources $x_i \in \mathcal{X}$ and $b_i \in \mathcal{B}$ s.t. $y_i = b_i + x_i$. Such supervision is much more generally available than full supervision, while the separation problem becomes much simpler than when fully unsupervised.

We introduce a novel method: Neural Egg Separation (NES) - consisting of i) iterative estimation of samples from the unobserved distribution $\mathcal{X}$ ii) synthesis of mixed signals from known samples of $\mathcal{B}$ and estimated samples of $\mathcal{X}$ iii) training of separation regressor to separate the mixed signal. Iterative refinement of the estimated samples of $\mathcal{X}$ significantly increases the accuracy of the learned masking function. As an iterative technique, NES can be initialization sensitive. We therefore introduce another method - GLO Masking (GLOM) - to provide NES with a strong initialization. Our method trains two deep generators end-to-end using GLO to model the observed and unobserved sources ($\mathcal{B}$ and $\mathcal{X}$). NES is very effective when $\mathcal{X}$ and $\mathcal{B}$ are uncorrelated, whereas initialization by GLOM is most important when $\mathcal{X}$ and $\mathcal{B}$ are strongly correlated such as e.g. separation of musical instruments. Initialization by GLOM was found to be much more effective than by adversarial methods.

Experiments are conducted across multiple domains (image, music, voice) validating the effectiveness of our method, and its superiority over current methods that use the same level of supervision. Our semi-supervised method is often competitive with the fully supervised baseline. It makes few assumptions on the nature of the component signals and requires lightweight supervision.

2 Previous Work

Source separation: Separation of mixed signals has been extensively researched. In this work, we focus on single channel separation. Unsupervised (blind) single-channel methods include: ICA (Davies & James, 2007) and RPCA (Huang et al., 2012). These methods attempt to use coarse priors about the signals such as low rank, sparsity or non-gaussianity. HMM can be used as a temporal prior for longer clips (Roweis, 2001), however here we do not assume long clips. Supervised source separation has also been extensively researched, classic techniques often used learned dictionaries for each source e.g. NMF (Wilson et al., 2008). Recently, neural network-based gained popularity, usually learning a regression between the mixed and unmixed signals either directly (Huang et al., 2014) or by regressing the mask (Wang et al., 2014; Yu et al., 2017). Some methods were devised to exploit the temporal nature of long audio signal by using RNNs (Mimilakis et al., 2017), in this work we concentrate on separation of short audio clips and consider such line of works as orthogonal. One related direction is Generative Adversarial Source Separation (Stoller et al., 2017; Subakan & Smaragdis, 2017) that uses adversarial training to match the unmixed source distributions. This is needed to deal with correlated sources for which learning a regressor on synthetic mixtures is less effective. We present an Adversarial Masking (AM) method that tackles the semi-supervised rather than the fully supervised scenario and overcomes mixture collapse issues not present in the fully supervised case. We found that non-adversarial methods perform better for the initialization task.

The most related set of works is semi-supervised audio source separation (Smaragdis et al., 2007; Barker & Virtanen, 2014), which like our work attempt to separate mixtures $\mathcal{Y}$ given only samples from the distribution of one source $\mathcal{B}$. Typically NMF or PLCA (which is a similar algorithm with a probabilistic formulation) are used. We show experimentally that our method significantly outperforms NMF.

Disentanglement: Similarly to source separation, disentanglement also deals with separation in terms of creating a disentangled representation of a source signal, however its aim is to uncover latent factors of variation in the signal such as style and content or shape and color e.g. Denton et al. (2017); Higgins et al. (2016). Differently from disentanglement, our task is separating signals rather than the latent representation.

Generative Models: Generative models learn the distribution of a signal directly. Classical approaches include: SVD for general signals and NMF (Lee & Seung, 2001) for non-negative signals. Recently several deep learning approaches dominated generative modeling including: GAN (Goodfellow et al., 2016), VAE (Kingma & Welling, 2013) and GLO (Bojanowski et al., 2018). Adversarial training (for GANs) is rather tricky and often leads to mode-collapse. GLO is non-adversarial and allows for direct latent optimization for each source making it more suitable than VAE and GAN.

3 Neural Egg Separation (NES)

In this section we present our method for separating a mixture of sources of known and unknown distributions. We denote the mixture samples y_i, the samples with the observed distribution b_i and the samples from the unobserved distribution x_i. Our objective is to learn a parametric function $T()$, such that $b_i = T(y_i)$.

Full Supervision: In the fully supervised setting (where pairs of y_i and b_i are available) this task reduces to a standard supervised regression problem, in which a parametric function $T()$ (typically a deep neural network) is used to directly optimize:

$$\arg\min_T L = \sum_i \ell(T(y_i), b_i) \tag{1}$$

Where typically ℓ is the Euclidean or the L_1 loss. In this work we use $\ell() = L_1()$.

Mixed-unmixed pairs are usually unavailable, but in some cases it is possible to obtain a training set which includes unrelated samples x_j and b_k e.g. (Wang et al., 2014; Yu et al., 2017). Methods typically randomly sample x_j and b_k sources and synthetically create mixtures $y_{jk} = x_j + b_k$. The synthetic pairs (b_k, y_{jk}) can then be used to optimize Eq. 1. Note that in cases where $\mathcal{X}$ and $\mathcal{B}$ are correlated (e.g. vocals and instrumental accompaniment which are temporally dependent), random synthetic mixtures of x and b might not be representative of y and fail to generalize on real mixtures.

Semi-Supervision: In many scenarios, clean samples of both mixture components are not available. Consider for example a street musical performance. Although crowd noises without street performers can be easily observed, street music without crowd noises are much harder to come by. In this case therefore samples from the distribution of crowd noise $\mathcal{B}$ are available, whereas the samples from the distribution of the music $\mathcal{X}$ are unobserved. Samples from the distribution of the mixed signal $\mathcal{Y}$ i.e. the crowd noise mixed with the musical performance are also available.

The example above illustrates a class of problems for which the distribution of the mixture and a single source are available, but the distribution of another source is unknown. In such cases, it is not possible to optimize Eq. 1 directly due to the unavailability of pairs of b and y.

Neural Egg Separation: Fully-supervised optimization (as in Eq. 1) is very effective when pairs of b_i and y_i are available. We present a novel algorithm, which iteratively solves the semi-supervised task as a sequence of supervised problems without any clean training examples of $\mathcal{X}$. We name the method Neural Egg Separation (NES), as it is akin to the technique commonly used for separating egg whites and yolks.

The core idea of our method is that although no clean samples from $\mathcal{X}$ are given, it is still possible to learn to separate mixtures of observed samples b_j from distribution $\mathcal{B}$ combined with some estimates of the unobserved distribution samples $\tilde{x_i}$. Synthetic mixtures are created by randomly sampling an approximate sample $\tilde{x_i}$ from the unobserved distribution and combining with training sample b_j:

$$\tilde{y}_{ij} = b_j + \tilde{x}_i \tag{2}$$

thereby creating pairs $(\tilde{y}_{ij}, b_j)$ for supervised training. Note that the distribution of synthetic mixtures $\tilde{y}_{ij}$ might be different from the real mixture sample distribution y_j, but the assumption (which is empirically validated) is that it will eventually converge to the correct distribution.

During each iteration of NES, a neural separation function $T()$ is trained on the created pairs by optimizing the following term:

$$T = arg\min_{T} L_1(T(b_j + \tilde{x_i}), b_j) \tag{3}$$

At the end of each iteration, the separation function $T()$ can be used to approximately separate the training mixture samples y_i into their sources:

$$\tilde{b}_i = T(y_i), \tilde{x}_i = y_i - \tilde{b}_i \tag{4}$$

The refined $\mathcal{X}$ domain estimates $\tilde{x}_i$ are used for creating synthetic pairs for finetuning $T()$ in the next iteration (as in Eq. 3).

The above method relies on having an estimate of the unobserved distribution samples as input to the first iteration. One simple scheme is to initialize the estimates of the unobserved distribution samples in the first iteration as $\tilde{x}_i = c \cdot y_i$, where c is a constant fraction (typically 0.5). Although this initialization is very naive, we show that it achieves very competitive performance in cases where the sources are independent. More advanced initializations will be discussed below.

At test time, separation is simply carried out by a single application of the trained separation function $T()$ (exactly as in Eq. 4).

```
**Data:** Mixture samples $\{y_i\}$, Observed source samples $\{b_j\}$
**Result:** Separation function $T()$
Initialize synthetic unobservable samples with $\tilde{x}_i \leftarrow c \cdot y_i$ or using AM or GLOM;
Initialize $T()$ with random weights;
while $iter < N$ do
    Synthesize mixtures $\tilde{y}_{ij} = b_j + \tilde{x}_i$ for all $b_j$ in $\mathcal{B}$;
    Optimize separation function for $P$ epochs: $T = arg\min_T \sum_i L_1(T(\tilde{y}_{ij}), b_i)$;
    Update estimates of unobserved distribution samples: $\tilde{x}_i = y_i - T(y_i)$
end
```

Algorithm 1: NES Algorithm

Our full algorithm is described in Alg. 1. For optimization, we use SGD using ADAM update with a learning rate of 0.01. In total we perform $N = 10$ iterations, each consisting of optimization of T and estimation of $\tilde{x}_i$, $P = 25$ epochs are used for each optimization of Eq. 3.

GLO Masking: NES is very powerful in practice despite its apparent simplicity. There are some cases for which it can be improved upon. As with other synthetic mixture methods, it does not take into account correlation between $\mathcal{X}$ and $\mathcal{B}$ e.g. vocals and instrumental tracks are highly related, whereas randomly sampling pairs of vocals and instrumental tracks is likely to synthesize mixtures quite different from $\mathcal{Y}$. Another issue is finding a good initialization—this tends to affect performance more strongly when $\mathcal{X}$ and $\mathcal{B}$ are dependent.

We present our method GLO Masking (GLOM), which separates the mixture by a distributional constraint enforced via GLO generative modeling of the source signals. GLO (Bojanowski et al., 2018) learns a generator $G()$, which takes a latent code z_b and attempts to reconstruct an image or a spectrogram: $b = G(z_b)$. In training, GLO learns end-to-end both the parameters of the generator $G()$ as well as a latent code z_b for every training sample b. It trains per-sample latent codes by *direct gradient descent over the values of* z_b (similar to word embeddings), rather than by a feedforward encoder used by autoencoders (e.g. $z_b = E(b)$). This makes it particularly suitable for our scenario. Let us define the set of latent codes: $Z = [z_1..z_N]$. The optimization is therefore:

$$\arg\min_{Z,G} \sum_{b \in \mathcal{B}} \|b, G(z_b)\| \tag{5}$$

We propose GLO Masking, which jointly trains generators: $G_B()$ for $\mathcal{B}$ and $G_X()$ for $\mathcal{X}$ such that their sum results in mixture samples $y = G_B(z_y^B) + G_X(z_y^X)$. We use the supervision of the observed source $\mathcal{B}$ to train $G_B()$, while the mixture $\mathcal{Y}$ contributes residuals that supervise the training of $G_X()$. We also jointly train the latent codes for all training images: $z_b \in Z$ for all $b \in \mathcal{B}$, and $z^B_y \in Z^B$, $z^X_y \in Z^X$ for all $y \in \mathcal{Y}$. The optimization problem is:

$$\arg\min_{Z, Z_B, Z_X, G_B, G_X} \sum_{b \in \mathcal{B}} \|b, G_B(z_b)\| + \lambda \cdot \sum_{y \in \mathcal{Y}} \|y, G_B(z^B_y) + G_X(z^X_y)\| \tag{6}$$

As GLO is able to overfit arbitrary distributions, it was found that constraining each latent code vector z to lie within the unit ball $z \cdot z \leq 1$ is required for generalization. Eq. 6 can either be optimized end-to-end, or the left-hand term can be optimized first to yield $Z, G_B()$, then the right-hand term is optimized to yield $Z_B, Z_X, G_X()$. Both optimization procedures yield similar performance (but separate training does not require setting λ). Once $G_B()$ and $G_X()$ are trained, for a new mixture sample we infer its latent codes:

$$\arg\min_{z_x, z_b} \|y, G_B(z^B_y) + G_X(z^X_y)\| \tag{7}$$

Our estimate for the sources is then:

$$\tilde{b} = G_B(z^B_y) \quad \tilde{x} = G_X(z^X_y) \tag{8}$$

Masking Function: In separation problems, we can exploit the special properties of the task e.g. that the mixed signal y_i is the sum of two positive signals x_i and b_i. Instead of synthesizing the new sample, we can instead simply learn a separation mask $m()$, specifying the fraction of the signal which comes from $\mathcal{B}$. The attractive feature of the mask is always being in the range $[0, 1]$ (in the case of positive additive mixtures of signals). Even a constant mask will preserve all signal gradients (at the cost of introducing spurious gradients too). Mathematically this can be written as:

$$T(y) = y \odot m(y) \tag{9}$$

For NES (and baseline AM described below), we implement the mapping function $T(y_i)$ using the product of the masking function $y_i \cdot m(y_i)$. In practice we find that learning a masking function yields much better results than synthesizing the signal directly (in line with other works e.g. Wang et al. (2014); Gabbay et al. (2017)).

GLOM models each source separately and is therefore unable to learn the mask directly. Instead we refine its estimate by computing an effective mask from the element-wise ratio of estimated sources:

$$m = \frac{G_B(z_b)}{G_B(z_b) + G_X(z_x)} \tag{10}$$

Initializing Neural Egg Separation by GLOM: Due to the iterative nature of NES, it can be improved by a good initialization. We therefore devise the following method: i) Train GLOM on the training set and infer the mask for each mixture. This is operated on images or mel-scale spectrograms at 64×64 resolutions ii) For audio: upsample the mask to the resolution of the high-resolution linear spectrogram and compute an estimate of the $\mathcal{X}$ source linear spectrogram on the training set iii) Run NES on the observed $\mathcal{B}$ spectrograms and estimated $\mathcal{X}$ spectrograms. We find experimentally that this initialization scheme improves NES to the point of being competitive with fully-supervised training in most settings.

4 Experiments

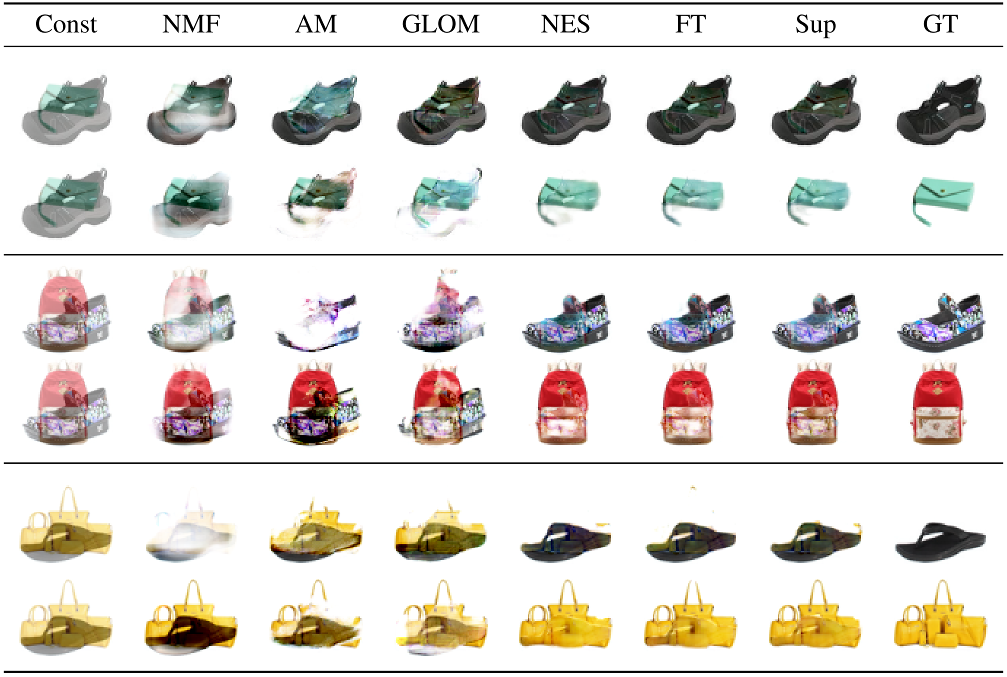
To evaluate the performance of our method, we conducted experiments on distributions taken from multiple real-world domains: images, speech and music, in cases where the two signals are correlated and uncorrelated.

We evaluated our method against 3 baseline methods:

Constant Mask (Const): This baseline uses the original mixture as the estimate.

Semi-supervised Non-negative Matrix Factorization (SS-NMF): This baseline method, proposed by Smaragdis et al. (2007), first trains a set of l bases on the observed distribution samples B by Sparse

Figure 1: A Qualitative Separation Comparison on Mixed Bag and Shoe Images



NMF (Hoyer, 2004; Kim & Park, 2007). It factorizes $B = H_b * W_b$, with activations H_b and bases W_b, all matrices are non-negative. The optimization is solved using the Non-negative Least Squares solver by Kim & Park (2011). It then proceeds to train another factorization on the mixture Y training samples with $2l$ bases, where the first l bases (W_b) are fixed to those computed in the previous stage: $Y = H_{yb} * W_b + H_{yx} * W_x$. The separated sources are then: $\tilde{x} = h_{yx} * W_x$ and $\tilde{b} = h_{yb} * W_b$. More details can be found in the appendix B.

Adversarial Masking (AM): As an additional contribution, we introduce a new semi-supervised method based on adversarial training, to improve over the shallow NMF baseline. AM trains a masking function $m()$ so that after masking, the training mixtures are indistinguishable from the distribution of source $\mathcal{B}$ under an adversarial discriminator $D()$. The loss functions (using LS-GAN (Mao et al., 2017)) are given by:

$$\arg\min_D L_D = \sum_{y \in \mathcal{Y}} D(y \odot m(y))^2 + \sum_{b \in \mathcal{B}} (D(b) - 1)^2 \qquad \arg\min_m L_m = \sum_{y \in \mathcal{Y}} (D(y \odot m(y)) - 1)^2 \tag{11}$$

Differently from CycleGAN (Zhu et al., 2017) and DiscoGAN (Kim et al., 2017), AM is not bidirectional and cannot use cycle constraints. We have found that adding magnitude prior $L_1(m(y), 1)$ improves performance and helps prevent collapse. To partially alleviate mode collapse, we use Spectral Norm (Miyato et al., 2018) on the discriminator.

We evaluated our proposed methods:

GLO Masking (GLOM): GLO Masking on mel-spectrograms or images at 64×64 resolution.

Neural Egg Separation (NES): The NES method detailed in Sec. 3. Initializing $\mathcal{X}$ estimates using a constant (0.5) mask over $\mathcal{Y}$ training samples.

Finetuning (FT): Initializing NES with the $\mathcal{X}$ estimates obtained by GLO Masking.

To upper bound the performance of our method, we also compute a *fully supervised* baseline, for which paired data of $b_i \in \mathcal{B}$, $x_i \in \mathcal{X}$ and $y_i \in \mathcal{Y}$ are available. We train a masking function with the same architecture as used by all other regression methods to directly regress synthetic mixtures to unmixed sources. This method uses more supervision than our method and is an upper bound.

Table 1: Image Separation Accuracy (PSNR dB/SSIM)

$\mathcal{X}$	$\mathcal{B}$	*Const*	*NMF*	*AM*	*GLOM*	*NES*	*FT*	*Supervised*
0-4	5-9	10.6/0.65	16.5/0.71	17.8/0.83	15.1/0.76	23.4/**0.95**	**23.9/0.95**	24.1/0.96
5-9	0-4	10.8/0.65	15.5/0.66	18.2/0.84	15.3/0.79	23.4/**0.95**	**23.8/0.95**	24.4/0.96
Bags	Shoes	6.9/0.48	13.9/0.48	15.5/0.67	15.1/0.66	22.3/0.85	**22.7/0.86**	22.9/0.86
Shoes	Bags	10.8/0.65	11.8/0.51	16.2/0.65	14.8/0.65	22.4/0.85	**22.8/0.86**	22.8/0.86

More implementation details can be found in appendix A.

4.1 Separating Mixed Images

In this section we evaluate the effectiveness of our method on image mixtures. We conduct experiments both on the simpler MNIST dataset and more complex Shoes and Handbags datasets.

4.1.1 MNIST

To evaluate the quality of our method on image separation, we design the following experimental protocol. We split the MNIST dataset (LeCun & Cortes, 2010) into two classes, the first consisting of the digits 0-4 and the second consisting of the digits 5-9. We conduct experiments where one source has an observed distribution $\mathcal{B}$ while the other source has an unobserved distribution $\mathcal{X}$. We use $12k$ $\mathcal{B}$ training images as the $\mathcal{B}$ training set, while for each of the other $12k$ $\mathcal{B}$ training images, we randomly sample a $\mathcal{X}$ image and additively combine the images to create the $\mathcal{Y}$ training set. We evaluate the performance of our method on 5000 $\mathcal{Y}$ images similarly created from the test set of $\mathcal{X}$ and $\mathcal{B}$. The experiment was repeated for both directions i.e. 0-4 being $\mathcal{B}$ while 5-9 in $\mathcal{X}$, as well as 0-4 being $\mathcal{X}$ while 5-9 in $\mathcal{B}$.

In Tab. 1, we report our results on this task. For each experiment, the top row presents the results (PSNR and SSIM) on the $\mathcal{X}$ test set. Due to the simplicity of the dataset, NMF achieved reasonable performance on this dataset. GLOM achieves better SSIM but worse PSNR than NMF while AM performed 1-2dB better. NES achieves much stronger performance than all other methods, achieving about 1dB worse than the fully supervised performance. Initializing NES with the masks obtained by GLOM, results in similar performance to the fully-supervised upper bound. FT from AM (numbers for finetuning from AM were omitted from the tables for clarity, as they were inferior to finetuning from GLOM in all experiments) achieved similar performance ($24.0/0.95$ and $23.8/0.95$) to FT from GLOM.

4.1.2 Bags and Shoes

In order to evaluate our method on more realistic images, we evaluate on separating mixtures consisting of pairs of images sampled from the Handbags (Zhu et al., 2016) and Shoes (Yu & Grauman, 2014) datasets, which are commonly used for evaluation of conditional image generation methods. To create each $\mathcal{Y}$ mixture image, we randomly sample a shoe image from the Shoes dataset and a handbag image from the Handbags dataset and sum them. For the observed distribution, we sample another 5000 different images from a single dataset. We evaluate our method both for cases when the $\mathcal{X}$ class is Shoes and when it is Handbags.

From the results in Tab. 1, we can observe that NMF failed to preserve fine details, penalizing its performance metrics. GLOM (which used a VGG perceptual loss) performed much better, due to greater expressiveness. AM performance was similar to GLOM on this task, as the perceptual loss and stability of training of non-adversarial models helped GLOM greatly. NES performed much better than all other methods, even when initialized from a constant mask. Finetuning from GLOM, helped NES achieve stronger performance, nearly identical to the fully-supervised upper bound. It performed better than finetuning from AM (not shown in table) which achieved $22.5/0.85$ and $22.7/0.86$. Similar conclusions can be drawn from the qualitative comparison in the figure above.

Table 2: Speech Separation Accuracy (PSNR dB)

$\mathcal{X}$	$\mathcal{B}$	*Const*	*NMF*	*AM*	*GLOM*	*NES*	*FT*	*Supervised*
Speech	Noise	0.0	2.4	5.7	3.3	**7.5**	**7.5**	8.3

4.2 Separating Speech and Environmental Noise

Separating environmental noise from speech is a long standing problem in signal processing. Although supervision for both human speech and natural noises can generally be obtained, we use this task as a benchmark to evaluate our method's performance on audio signals where $\mathcal{X}$ and $\mathcal{B}$ are not dependent. This benchmark is a proxy for tasks for which a clean training set of $\mathcal{X}$ sounds cannot be obtained e.g. for animal sounds in the wild, background sounds training without animal noises can easily be obtained, but clean sounds made by the animal with no background sounds are unlikely to be available.

We obtain clean speech segments from the Oxford-BBC Lip Reading in the Wild (LRW) Dataset (Chung & Zisserman, 2016), and resample the audio to 16 kHz. Audio segments from ESC-50 (Piczak, 2015), a dataset of environmental audio recordings organized into 50 semantic classes, are used as additive noise. Noisy speech clips are created synthetically by first splitting clean speech into clips with duration of 0.5 seconds, and adding a random noise clip, such that the resulting SNR is zero. We then compute a mel-scale spectrogram with 64 bins, using STFT with window size of 25 ms, hop length of 10 ms, and FFT size of 512, resulting in an input audio feature of 64×64 scalars. Finally, power-law compression is performed with $p = 0.3$, i.e. $A^{0.3}$, where A is the input audio feature.

From the results in Tab. 2, we can observe that GLOM, performed better than Semi-Supervised NMF by about 1dB better. AM training, performed about 2dB better than GLOM. Due to the independence between the sources in this task, NES performed very well, even when trained from a constant mask initialization. Performance was less than 1dB lower than the fully supervised result (while not requiring any clean speech samples). In this setting due to the strong performance of NES, initializing NES with the speech estimates obtained by GLOM (or AM), did not yield improved performance.

4.3 Music Separation

Separating vocal music into singing voice and instrumental music as well as instrumental music and drums has been a standard task for the signal processing community. Here our objective is to understand the behavior of our method in settings where $\mathcal{X}$ and $\mathcal{B}$ are dependent (which makes synthesis by addition of random $\mathcal{X}$ and $\mathcal{B}$ training samples a less accurate approximation).

For this task we use the MUSDB18 Dataset (Rafii et al., 2017), which, for each music track, comprises separate signal streams of the mixture, drums, bass, the rest of the accompaniment, and the vocals. We convert the audio tracks to mono, resample to 20480 Hz, and then follow the procedure detailed in Sec. 4.2 to obtain input audio features.

From the results in Tab. 3, we can observe that NMF was the worst performer in this setting (as its simple bases do not generalize well between songs). GLOM was able to do much better than NMF and was even competitive with NES on Vocal-Instrumental separation. Due to the dependence between the two sources and low SNR, initialization proved important for NES. Constant initialization NES performed similarly to AM and GLOM. Finetuning NES from GLOM masks performed much better than all other methods and was competitive with the supervised baseline. GLOM was much better than AM initialization (not shown in table) that achieved 0.9 and 2.9.

5 Discussion

GLO vs. Adversarial Masking: GLO Masking as a stand alone technique usually performed worse than Adversarial Masking. On the other hand, finetuning from GLO masks was far better than finetuning from adversarial masks. We speculate that mode collapse, inherent in adversarial training,

Table 3: Music Separation Accuracy (Median SDR dB)

$\mathcal{X}$	$\mathcal{B}$	*Const*	*NMF*	*AM*	*GLOM*	*NES*	*FT*	*Supervised*
Vocals	Instrumental	-3.5	0.0	0.3	0.6	0.3	**2.1**	2.7
Drums	Instrumental	-3.3	-0.5	1.5	0.8	1.3	**3.5**	3.6

makes the adversarial masks a lower bound on the $\mathcal{X}$ source distribution. GLOM can result in models that are too loose (i.e. that also encode samples outside of $\mathcal{X}$). But as an initialization for NES finetuning, it is better to have a model that is too loose than a model which is too tight.

Supervision Protocol: Supervision is important for source separation. Completely blind source separation is not well specified and simply using general signal statistics is generally unlikely to yield competitive results. Obtaining full supervision by providing a labeled mask for training mixtures is unrealistic but even synthetic supervision in the form of a large training set of clean samples from each source distribution might be unavailable as some sounds are never observed on their own (e.g. sounds of car wheels). Our setting significantly reduces the required supervision to specifying if a certain sound sample contains or does not contain the unobserved source. Such supervision can be quite easily and inexpensively provided. For further sample efficiency increases, we hypothesize that it would be possible to label only a limited set of examples as containing the target sound and not, and to use this seed dataset to finetune a deep sound classifier to extract more examples from an unlabeled dataset. We leave this investigation to future work.

Signal-Specific Losses: To showcase the generality of our method, we chose not to encode task specific constraints. In practical applications of our method however we believe that using signal-specific constraints can increase performance. Examples of such constraints include: repetitiveness of music (Rafii & Pardo, 2011), sparsity of singing voice, smoothness of natural images.

Non-Adversarial Alternatives: The good performance of GLOM vs. AM on the vocals separation task, suggests that non-adversarial generative methods may be superior to adversarial methods for separation. This has also been observed in other mapping tasks e.g. the improved performance of NAM (Hoshen & Wolf, 2018) over DCGAN (Radford et al., 2015).

Convergence of NES: A perfect signal separation function is a stable global minimum of NES as i) the synthetic mixtures are equal to real mixtures ii) real mixtures are perfectly separated. In all NES experiments (with constant, AM or GLOM initialization), NES converged after no more than 10 iterations, typically to different local minima. It is empirically evident that NES is not guaranteed to converge to a global minimum (although it converges to good local minima). We defer formal convergence analysis of NES to future work.

6 Conclusions

In this paper we proposed a novel method—Neural Egg Separation—for separating mixtures of observed and unobserved distributions. We showed that careful initialization using GLO Masking improves results in challenging cases. Our method achieves much better performance than other methods and was usually competitive with full-supervision.

References

Tom Barker and Tuomas Virtanen. Semi-supervised non-negative tensor factorisation of modulation spectrograms for monaural speech separation. In *2014 International Joint Conference on Neural Networks (IJCNN)*, pp. 3556–3561. IEEE, 2014.

Piotr Bojanowski, Armand Joulin, David Lopez-Paz, and Arthur Szlam. Optimizing the latent space of generative networks. In *ICML'18*, 2018.

J. S. Chung and A. Zisserman. Lip reading in the wild. In *Asian Conference on Computer Vision*, 2016.

Mike E Davies and Christopher J James. Source separation using single channel ica. *Signal Processing*, 87(8):1819–1832, 2007.

Emily L Denton et al. Unsupervised learning of disentangled representations from video. In *Advances in Neural Information Processing Systems*, pp. 4414–4423, 2017.

Aviv Gabbay, Ariel Ephrat, Tavi Halperin, and Shmuel Peleg. Seeing through noise: Visually driven speaker separation and enhancement. *arXiv preprint arXiv:1708.06767*, 2017.

Ian Goodfellow, Yoshua Bengio, Aaron Courville, and Yoshua Bengio. *Deep learning*, volume 1. MIT Press, 2016.

Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. 2016.

Yedid Hoshen and Lior Wolf. Nam: Non-adversarial unsupervised domain mapping. In *ECCV'18*, 2018.

Patrik O Hoyer. Non-negative matrix factorization with sparseness constraints. *Journal of machine learning research*, 5(Nov):1457–1469, 2004.

Po-Sen Huang, Scott Deeann Chen, Paris Smaragdis, and Mark Hasegawa-Johnson. Singing-voice separation from monaural recordings using robust principal component analysis. In *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on*, pp. 57–60. IEEE, 2012.

Po-Sen Huang, Minje Kim, Mark Hasegawa-Johnson, and Paris Smaragdis. Deep learning for monaural speech separation. In *ICASSP*, 2014.

Hyunsoo Kim and Haesun Park. Sparse non-negative matrix factorizations via alternating non-negativity-constrained least squares for microarray data analysis. *Bioinformatics*, 23(12):1495–1502, 2007.

Jingu Kim and Haesun Park. Fast nonnegative matrix factorization: An active-set-like method and comparisons. *SIAM Journal on Scientific Computing*, 33(6):3261–3281, 2011.

Taeksoo Kim, Moonsu Cha, Hyunsoo Kim, Jungkwon Lee, and Jiwon Kim. Learning to discover cross-domain relations with generative adversarial networks. *arXiv preprint arXiv:1703.05192*, 2017.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Yann LeCun and Corinna Cortes. MNIST handwritten digit database. 2010.

Daniel D Lee and H Sebastian Seung. Algorithms for non-negative matrix factorization. In *Advances in neural information processing systems*, pp. 556–562, 2001.

Xudong Mao, Qing Li, Haoran Xie, Raymond YK Lau, Zhen Wang, and Stephen Paul Smolley. Least squares generative adversarial networks. In *Computer Vision (ICCV), 2017 IEEE International Conference on*, pp. 2813–2821. IEEE, 2017.

Stylianos Ioannis Mimilakis, Konstantinos Drossos, João F Santos, Gerald Schuller, Tuomas Virtanen, and Yoshua Bengio. Monaural singing voice separation with skip-filtering connections and recurrent inference of time-frequency mask. *arXiv preprint arXiv:1711.01437*, 2017.

Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. In *International Conference on Learning Representations*, 2018.

Karol J. Piczak. ESC: Dataset for Environmental Sound Classification. In *Proceedings of the 23rd Annual ACM Conference on Multimedia*, pp. 1015–1018. ACM Press, 2015.

Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

Zafar Rafii and Bryan Pardo. A simple music/voice separation method based on the extraction of the repeating musical structure. In *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on*, pp. 221–224. IEEE, 2011.

Zafar Rafii, Antoine Liutkus, Fabian-Robert Stöter, Stylianos Ioannis Mimilakis, and Rachel Bittner. The MUSDB18 corpus for music separation, December 2017. URL `https://doi.org/10.5281/zenodo.1117372`.

Sam T Roweis. One microphone source separation. In *Advances in neural information processing systems*, pp. 793–799, 2001.

Paris Smaragdis, Bhiksha Raj, and Madhusudana Shashanka. Supervised and semi-supervised separation of sounds from single-channel mixtures. In *International Conference on Independent Component Analysis and Signal Separation*, pp. 414–421. Springer, 2007.

Daniel Stoller, Sebastian Ewert, and Simon Dixon. Adversarial semi-supervised audio source separation applied to singing voice extraction. *arXiv preprint arXiv:1711.00048*, 2017.

Cem Subakan and Paris Smaragdis. Generative adversarial source separation. *arXiv preprint arXiv:1710.10779*, 2017.

Yuxuan Wang, Arun Narayanan, and DeLiang Wang. On training targets for supervised speech separation. *TASLP*, 2014.

Kevin W Wilson, Bhiksha Raj, Paris Smaragdis, and Ajay Divakaran. Speech denoising using nonnegative matrix factorization with priors. In *Acoustics, Speech and Signal Processing, 2008. ICASSP 2008. IEEE International Conference on*, pp. 4029–4032. IEEE, 2008.

Aron Yu and Kristen Grauman. Fine-grained visual comparisons with local learning. In *CVPR*, 2014.

Dong Yu, Morten Kolbæk, Zheng-Hua Tan, and Jesper Jensen. Permutation invariant training of deep models for speaker-independent multi-talker speech separation. In *ICASSP*, 2017.

Jun-Yan Zhu, Philipp Krähenbühl, Eli Shechtman, and Alexei A Efros. Generative visual manipulation on the natural image manifold. In *European Conference on Computer Vision*, pp. 597–613. Springer, 2016.

Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

A Architectures

In this section we give details of the architectures used in our experiments:

A.1 Image Architectures:

GLOM Generator: GLOM trains generators for $\mathcal{X}$ and $\mathcal{B}$, each generates an image given a latent vector. The architecture followed the architecture first employed by DCGAN (Radford et al., 2015). We used 64 filters across MNIST and Shoes-Bags experiments. MNIST had one fewer layer, owing to its 32×32 resolution. Generators were followed by sigmoid layers to ensure outputs within $[0, 1]$.

GAN Discriminator: The discriminator used by Adversarial Masking is a DCGAN discriminator with filter dimensionality of 64. SpectralNorm is implemented exactly as described in (Miyato et al., 2018).

Masking Network: Adversarial Masking, NES and the fully supervised baseline all use the same masking function architecture. The masking function takes a mixed image and outputs a mask, that when multiplied by the image results in the $\mathcal{B}$ source: $b = y \cdot m(y)$. The architecture is an autoencoder similar to the one used in DiscoGAN (Kim et al., 2017). MNIST has two fewer layers owing to its lower resolution. We used 64 filters on the top and bottom layers, and doubling / halving the filter number before / after the autoencoder mid-way layer.

A.2 Audio Architectures:

GLOM and AM use the same generator and discriminator architectures respectively for audio as they do for images. They operate on mel-scale spectrogram at 64×64 resolution.

Masking Network: The generator for AM operates on 64×64 mel-scale audio spectrograms. It consists of 3 convolutional and 3 deconvolutional layers with stride 2 and no pooling. Outputs of convolutional layers are normalized with BatchNorm and rectified with ReLU activation, except for the last layer where sigmoid is used.

In addition to the LSGAN loss, an additional magnitude loss is used, with relative weight of $\lambda = 1$.

NES and the supervised method operate on full linear spectrogram of dimensions 257×64, without compression. They use the same DiscoGAN architecture, which contains two additional convolutional and deconvolutional layers.

B NMF Semi-Supervised Separation:

In this section we describe our implementation of the NMF semi-supervised source separation baseline (Smaragdis et al., 2007). NMF trains a decomposition: $B = WZ$ where W are the weights and $Z = [z_1, ..., z_N]$ are the per sample latent codes. Both W and Z are non-negative. Regularization is important for the performance of the method. We follow (Hoyer, 2004; Kim & Park, 2007) and use L_1 regularization to ensure sparsity of the weights. The optimization problem therefore becomes:

$$L = \|B, WZ\|_2^2 + \alpha|W1|_1^2 = \|[B;0],[WZ;\sqrt{\alpha}\cdot W1]\|_2^2 \quad s.t. \quad W, Z \geq 0 \tag{12}$$

This equation is solved iteratively using non-negative least squares (NNLS) with the solver by Kim & Park (2011). The Z iteration solves the following NNLS problem:

$$\arg\min_{Z} \|B, WZ\|_2^2 \quad s.t. \quad Z \geq 0 \tag{13}$$

The W iteration optimizes the following NNLS problem:

$$L = \|[B^T, 0], [Z^T W^T, \sqrt{\alpha}\cdot W1^T]\|_2^2 \quad s.t. \quad W^T \geq 0 \tag{14}$$

W and Z iterations are optimized until convergence.

Following Smaragdis et al. (2007), we first train sparse NMF for the training $\mathcal{B}$ samples: $B = W^B Z^B$. Using the weights from this stage, we proceed to train another NMF decomposition on the residuals of the mixture:

$$L = \|Y, W^B Z^B + W^X Z^X\| \quad s.t. \quad W^B, W^X, Z^B, Z^X \geq 0 \tag{15}$$

The W^X iteration consists of NNLS optimization:

$$\arg\min_{W^X} \|(Y - W^B Z^B)^T, (Z^X)^T (W^X)^T\| \quad s.t. \quad W^X \geq 0 \tag{16}$$

The Z iteration consists of NNLS optimization of both Z^B and Z^X on the mixture samples:

$$\arg\min_{Z^B, Z^X} \|Y, [W^B; W^X][Z^B; Z^X]\| \quad s.t. \quad Z^B, Z^X \geq 0 \tag{17}$$

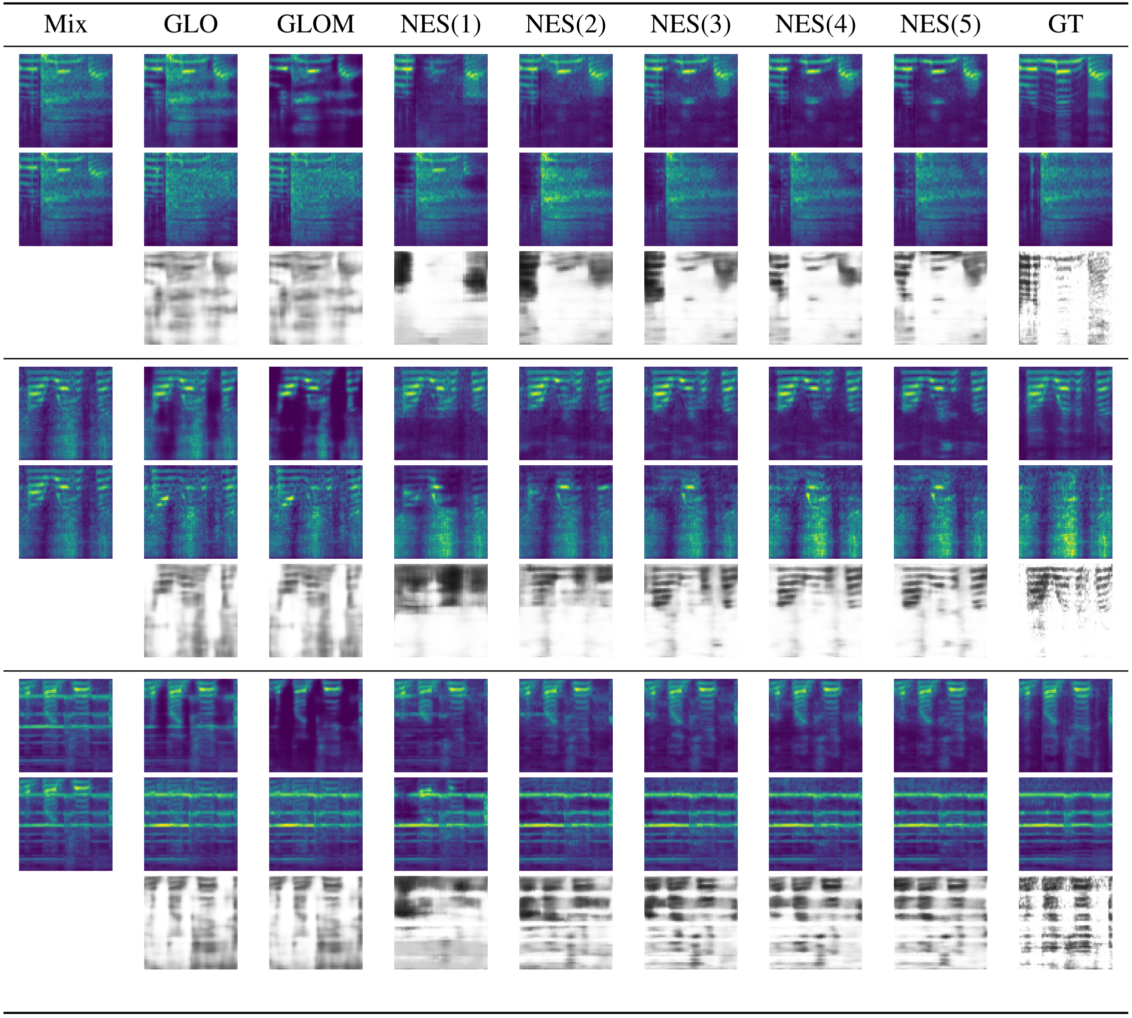
In the above we neglected the sparsity constraint for pedagogical reasons. It is implemented exactly as in Eq. 14.

At inference time, the optimization is equivalent to Eq. 17. After inference of Z^B and Z^X, our estimated B and Z sources are given by:

$$\tilde{B} = W^B Z^B \qquad \tilde{X} = W^X Z^X \tag{18}$$

In our experiments we used Z dimension of 50 and sparsity $\alpha = 1.0$. The hyper-parameters were determined by performance on a small validation set.

Figure 2: A qualitative comparison of speech and noise mixtures separated by GLO and GLOM, as well as NES after k iterations. NES(k) denotes NES after k iterations. Note that GLO and GLOM share the same mask since GLOM is generated by the mask computed from GLO.



C Further qualitative analysis of GLOM and NES

We present a qualitative analysis of the results of GLOM and NES. To understand the quality of generations of GLO and the effect of the masking function, we present in Fig.2 the results of the GLO generations given different mixtures from the Speech dataset. We also show the results after the masking operation described in Eq. 10. It can be observed that GLO captures the general features of the sources, but is not able to exactly capture fine detail. The masking operation in GLOM helps it recover more fine-grained details, and results in much cleaner separations.

We also show in Fig.2 the evolution of NES as a function of iteration for the same examples. NES(k) denotes the result of NES after k iterations. It can be seen the NES converges quite quickly, and results improve further with increasing iterations. In Fig.3, we can observe the performance of NES on the Speech dataset in terms of SDR as a function of iteration. The results are in line with the qualitative examples presented before, NES converges quickly but makes further gains with increasing iterations.

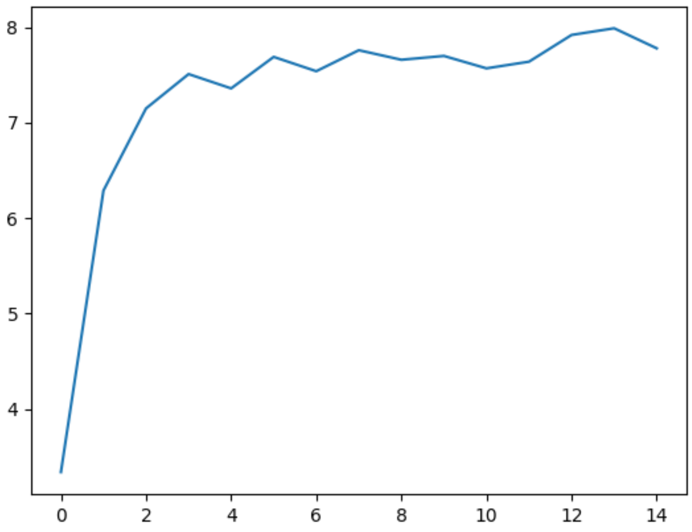


Figure 3: Speech test set separation quality as a function of NES iteration (SDR dB)