Representation Learning for Single-Channel Source Separation and Bandwidth Extension

Matthias Zöhrer, Robert Peharz and Franz Pernkopf, *Senior Member, IEEE*

***Abstract*—In this paper, we use deep representation learning for model-based single-channel source separation (SCSS) and artificial bandwidth extension (ABE). Both tasks are ill-posed and source-specific prior knowledge is required. In addition to well-known generative models such as restricted Boltzmann machines and higher order contractive autoencoders two recently introduced deep models, namely generative stochastic networks (GSNs) and sum-product networks (SPNs), are used for learning spectrogram representations. For SCSS we evaluate the deep architectures on data of the 2^{nd} CHiME speech separation challenge and provide results for a speaker dependent, a speaker independent, a matched noise condition and an unmatched noise condition task. GSNs obtain the best PESQ and overall perceptual score on average in all four tasks. Similarly, frame-wise GSNs are able to reconstruct the missing frequency bands in ABE best, measured in frequency-domain segmental SNR. They outperform SPNs embedded in hidden Markov models and the other representation models significantly.**

***Index Terms*—Representation learning, deep neural networks, generative stochastic networks, sum-product networks, single-channel source separation, bandwidth extension**

I. Introduction

In recent years, deep learning has been increasingly applied in signal processing and speech technology outperforming many well-established approaches [1], [2], [3], [4], [5], [6], [7], [8]. In [4], key aspects of deep models have been identified explaining some of the performance gain, namely, the re-use of features in consecutive layers and the degree of abstraction of features at higher layers. Deep models can be grouped into generative, hybrid and discriminative learning models, however the learning objective for *good* representations is not always evident. Generative models learn a representation of the underlying data. They are mainly used for reconstruction, filtering or enhancing information [9], [10]. Hybrid models use representations obtained by generative models and further fine-tuning, i.e. discriminative training of the model using output labels. Hybrid models commonly outperform discriminatively trained models e.g. deep belief networks (DBNs) outperform multi-layer perceptrons (MLPs) [11] on data with limited sample size.

Most accomplishments originate from generative (or hybrid) variants of restricted Boltzmann machines (RBMs) [12], [13], auto-encoders (AEs) [14], [15], [16] and sparse-coding [17], [18]. RBMs also form the basis of powerful neural networks, i.e. deep belief networks [19]. Recently, two particularly interesting representation models have been introduced, namely generative stochastic networks (GSNs) [9], [20] and sum-product networks (SPNs) [21], [22], [23], [24], [25], [26].

GSNs, a generalization of generalized auto-encoders (GAEs), [27] are a multi-layer network architecture using backprop-able stochastic neurons, learned with *walkback* training [9]. GSNs are able to outperform various competitive baseline systems in image reconstruction tasks, i.e. recovering missing or covered parts of an image [9], [28]. We extended GSNs to a hybrid generative-discriminative learning model paving the way for multi-objective learning [20]. They outperformed many state-of-the-art models including DBNs and AEs in classification.

In the context of probabilistic graphical models, SPNs are a promising model for potentially deep representation learning. SPNs can be interpreted as Bayesian networks [29], [30] with a deep hierarchical structure of latent variables with a high degree of context-specific independence. Inference in SPNs is *linear* in the networks size, i.e. in the number of nodes and edges in the network. They also show convincing performance on the task of image completion [21], [22], [31]. Both models, SPNs and GSNs, are therefore interesting candidates for learning speech representations required for single-channel source separation (SCSS) and artificial bandwidth extension (ABE).

In this paper, we systematically analyze representation learning for two speech processing applications: SCSS and ABE. We use popular generative models from representation learning including Gauss Bernoulli restricted Boltzmann machines (GBRBMs) [13], conditional Gauss Bernoulli restricted Boltzmann machines (CGBRBMs) [32], higher order contractive autoencoders (HCAEs) [16], SPNs, and GSNs. Furthermore, a rectifier MLP is applied to both tasks to facilitate a comparison of the results in this work to [33], [34], [35], [4], [36].

In the following, we shortly introduce both tasks including relevant literature and the experimental setup.

A. Single-channel source separation

In SCSS a mixture of two signals is separated into its underlying source signals. This problem is ill-posed and

Matthias Zöhrer and Franz Pernkopf are with the Intelligent Systems Group at the Signal Processing and Speech Communication Laboratory, Graz University of Technology, Graz, Austria. Robert Peharz is with iDN – Institute of Physiology, Medical University of Graz, working in the Brain, Ears and Eyes – Pattern Recognition Initiative, BioTechMed-Graz.

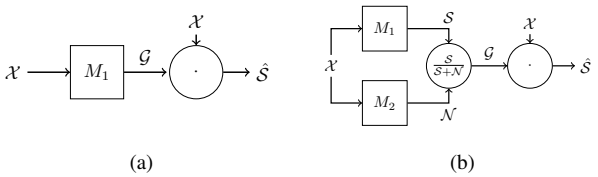
This work was supported by the Austrian Science Fund (FWF) under the project number P25244-N15, P27803-N15, and the K-Project ASD. The K-Project ASD is funded in the context of COMET Competence Centers for Excellent Technologies by BMVIT, BMWFJ, Styrian Business Promotion Agency (SFG), the Province of Styria - Government of Styria and the Technology Agency of the City of Vienna (ZIT). The program COMET is conducted by Austrian Research Promotion Agency (FFG). Furthermore, we acknowledge NVIDIA for providing GPU computing resources.

difficult to solve. One of the first model-based approach is the factorial-max vector quantization (VQ) [37], where the underlying assumption is that speech is sparse, i.e. each time frequency bin belongs to one of the two assumed sources leading to the notion of binary mask. Another method for identifying components with temporal structure in a time-frequency representation is non-negative matrix factorization (NMF) [38], [39], [40], [41]. All these approaches usually require sufficient speaker/source specific data for learning which restricts their applicability to scenarios with matching training/test conditions. In case of model mismatch, adaption techniques can be applied. Some of the most successful approaches for model adaption in the context of speech recognition are the maximum likelihood linear regression (MLLR) framework [42], [43], maximum a posteriori (MAP) estimation [44], and rapid adaption in eigenvoice space [45]. While these approaches assume that adaption data consists of clean speech, methods for adaption of *undistorted* source models from contaminated speech have been also developed, e.g. in [46]. Rose et al. [47] extended this in terms of a more general interaction and background noise model based on Gaussian mixture models (GMMs). Recently, we developed an adaption technique to overcome any model mismatch between training and testing conditions introduced by variations in the room acoustics or changed speaker position [48]. We successfully adapted speaker independent (SI) source models trained on clean utterances to a different acoustic channel and achieve almost the same performance level as speaker dependent (SD) models trained on reverberated utterances from this acoustic environment.

From the literature we identify two main approaches to SCSS:

1) Direct learning of either the binary- or the soft-mask $\mathcal{G}$ given a signal mixture $\mathcal{X}$ as input. Figure (1a) shows the direct learning approach. In [49], structured prediction was used to directly learn/estimate the ideal binary mask (IBM) from a mixture spectrogram. This has been extended in [34] using several neural network architectures and in [33] to several other masks, such as, TBM – the target binary mask, IRM – the ideal ratio mask, and FFT-Mask, i.e. short-time Fourier transform (STFT) mask.
2) Indirect learning of the binary- or soft-mask by predicting the individual source representations separately with two individual models [37], [48], [50], [51]. Typically both models are trained on speaker/source specific data and during separation the combination of both models fitting the observed mixture best is determined to extract the mask for separation. Approaches based on NMF or VQ typically belong to this class.

Here, we follow a different approach and learn a mapping from speech mixture to single-source spectrograms, i.e. the models *filter* the spectral representation of the speech mixture. During separation, the individual source spectrograms are inferred by the models and the mask $\mathcal{G}$ for re-synthesis of the time signals can be easily computed. Figure (1b) shows this approach. Two individual models M_1 and M_2 are trained to predict the log-magnitude spectrogram of speech $\mathcal{S}$ or log-magnitude spectrogram of the interfering source (e.g. noise) $\mathcal{N}$ given the log-magnitude spectrogram of the mixed utterance $\mathcal{X}$. The models M_1 and M_2 are representation models described in Section II. Both predictions $\mathcal{S}$ and $\mathcal{N}$ are used to compute the softmask i.e. $\mathcal{G}(t,f) = \frac{\mathcal{S}(t,f)}{\mathcal{S}(t,f)+\mathcal{N}(t,f)}$, where f and t are the time and frequency bins and $\mathcal{N}(t,f)$ and $\mathcal{S}(t,f)$ are the interfering source and speech spectrogram bins recovered by the corresponding models, respectively. Mask $\mathcal{G}$ is then applied to the mixed signal, recovering an estimate of the speech spectrogram $\hat{\mathcal{S}}$. Figure (1a) shows the direct learning approach studied in detail in [33], [34]. In this case a model M_1 is trained to *directly* predict the mask $\mathcal{G}$ given $\mathcal{X}$. This has been slightly modified in [35] for speech enhancement. There, the enhanced signal is directly predicted from the log-magnitude spectral data.



(a) (b)

Fig. 1: Approaches for SCSS: (a) Direct learning of the mask $\mathcal{G}$ by one model (M_1); (b) *Filter* approach with two models, one for each source ($M_1 + M_2$), to determine the mask $\mathcal{G}$.

In experiments, signal mixtures of the 2nd CHiME speech separation challenge [52] are separated in four different setups: SD, SI, a matched noise condition (MN) and an unmatched noise condition (UN) task. The GSN, using a *filter* approach, outperforms on average MLPs, RBMs and HCAEs in terms of the PESQ [53] score and the overall perceptual score (OPS) from the PEASS toolbox [54].

B. Artificial Bandwidth Extension

ABE aims to recover high frequency signal components given the telephone band of a speech signal. Motivated by the success of GSNs and SPNs on the task of image completion [9], [21], we use both models to complete the high frequency parts of log-spectrograms, lost due to the telephone bandpass filter. In a recent work deep neural networks (DNNs) have been proposed for ABE [36]. First, unsupervised pre-training of RBMs is performed using the log-spectrum of the narrow-band signal as input. These models are then discriminatively fine-tuned for providing the high-frequency part of the wide-band signal. We extend this work to generative deep networks, such as SPNs and GSNs. In particular, we modify the HMM-based framework for ABE [55], [58] and incorporate the representation SPN learning approach for modeling the observations in a hybrid ansatz. In addition, we use generative models for inferring the missing frequencies in a frame-wise fashion reducing the model complexity. Our experiments show that HMM-SPNs achieve the best log-spectral distortion (LSD) – slightly better than GSNs, while GSNs outperform HMM-SPNs in terms of

segmental SNR (sSNR) in the frequency-domain, resulting in an average relative improvement of 3.90dB and 4.08dB on both speaker dependent (SD) and speaker independent (SI) tasks, respectively.

In [55], an HMM based system is introduced, where time signals are processed frame-wise. The spectral envelope of the high-frequency band in each frame is modeled using cepstral coefficients obtained from linear prediction (LP). On a training set, these cepstral coefficients are clustered using the LBG algorithm [56] and the temporally aligned cluster indices are used as hidden state sequence of an HMM, whose prior and transition probabilities can be estimated using the observed frequency estimates. For each hidden state, an observation GMM is trained on features taken from the low-band (see [55] for details about these features). In the test phase, the high frequency components and therefore the hidden states of the HMM are missing and the marginal probability of the hidden state is inferred using the forward-backward algorithm [57].[1] Using the hidden state posterior, an MMSE estimate of the high-band cepstral coefficients is obtained [55], which together with the periodogram of the low-band yield estimates of the wide-band cepstral coefficients. To extend the excitation signal to the high-band, the low-band excitation is modulated either with a fixed frequency carrier, or with a pitch-dependent carrier. According to [55] and related ABE literature, the results are quite insensitive to the method of extending the excitation.

The paper is organized as follows: In Section II we shortly discuss the used representation models. Section III describes the experimental setup (III-A) and presents the results (III-B) for SCSS. In Section IV, the setup (IV-A) and results (IV-C) for ABE are discussed. Section V concludes the paper.

II. Models for Representation Learning

We use four representatives of deep models for representation learning. The first two are RBM and AE variants shortly summarized in the next section. The recently introduced SPNs and GSNs are discussed in more detail in the remaining two sections.

A. RBMs and AEs

RBMs [59], [60], [61], [13], [19] are a particular form of log-linear Markov random fields, where the energy function is linear in its free parameters. Learning in RBMs corresponds to modifying this energy function to obtain desirable properties. This can be accomplished via contrastive divergence training, i.e. a kind of block Gibbs sampling applied to the RBM Markov chain for k-steps. There are binary RBMs [62] or real-valued GBRBMs [13] for learning representations of the underlying data. They can be also used to capture temporal relations, i.e. Conditional GBRBMs [32].

AEs [63], [15], [64], [16], [65], [27] map the input to a hidden representation and transfer the latent representation back into a reconstruction using an inverse transformation. AEs are mainly used as feature extractors [64], filters or data generators [27]. They are able to learn a representation of the underlying data and can also be stacked forming deep models. An interesting encoder variant, also used in this work, is the HCAE [16], regularizing the norm of the Jacobian (*analytically*) and Hessian (*stochastically*) to obtain a better data representation.

[1]For real-time capable systems, the backward-messages have to be obtained from a limited number of look-ahead frames.

B. GSNs

GSNs [9], [20], extend the class of AEs to multiple hidden layers, which are jointly optimized during training.

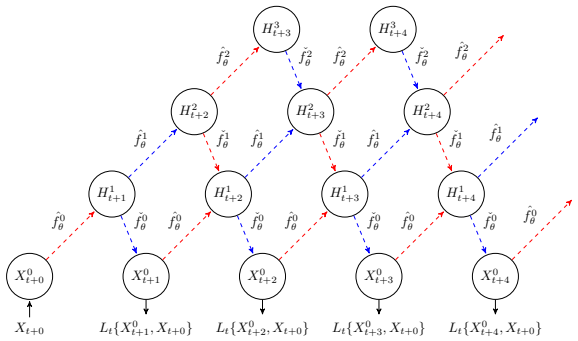


Fig. 2: *Unfolded multi-layer GSN with backprop-able stochastic units [20].*

Figure 2 visualizes an unfolded multi-layer GSN for $k = 4$ *walkback* steps, described in detail in [20]. GSNs indirectly capture the data distribution as the stationary distribution of the Markov chain, defined by a corruption/denoising process, under mild conditions. Due to *walkback* training and their hierarchical structure which is jointly optimized they form a powerful model class, especially when used for input reconstruction [9].

The model uses backprop-able stochastic neurons, modeled by functions of random variables $f^i_\theta \supseteq \{\hat{f}^i_\theta, \check{f}^i_\theta\}$. These functions express a Markov chain with additional dependencies between the hidden states, i.e. $H_{t+1} \sim P_{\theta_1}(H|H_{t+0}, X_{t+0})$, $X_{t+1} \sim P_{\theta_2}(X|H_{t+1})$. In particular, the function $\hat{f}^i_\theta$ models $H^i_{t+1} = \hat{f}_{\theta^i}(X_{t+0}, Z_{t+0}, H_{t+0})$, specified for some independent noise source Z_{t+0}, with the condition that the input X_{t+0} cannot be recovered exactly from H_{t+1}. The function $\hat{f}^i_\theta = \eta^i_{out} + g(\eta^i_{in} + \hat{a}^i)$ is a backprop-able stochastic non-linearity for layer i, where $Z^i_t \supseteq \{\eta^i_{in}, \eta^i_{out}\}$ are noise processes and g is a non-linear activation function. The term $\hat{a}^i = W^i I^i_t + b^i$ defines the activations for layer i with a weight matrix W^i and bias b^i, representing the parametric distribution P_{θ_1}. In general, $\hat{f}^i_\theta(I^i_t)$ specifies an upward path in a GSN, where the input I^i_t is either the realization x^i_t of observed sample X^i_t or the hidden realization h^i_t of H^i_t. In the case of $X^i_{t+1} = \check{f}^i_\theta(Z_{t+0}, H_{t+1})$, $\check{f}^i_\theta(H^i_t) = \eta^i_{out} + g(\eta^i_{in} + \check{a}^i)$ defines a downward path in the network i.e. $\check{a}^i = (W^i)^T H^i_t + b^i$, using the transpose of the weight matrix W^i. This formulation allows to directly back-propagate the reconstruction log-likelihood $P_{\theta_2}(X|H)$ for all parameters $\theta \supseteq \{W^0, ..., W^d, b^0, ..., b^d\}$ using multiple functions of random variables $f_\theta \in \{\hat{f}^0_\theta, ..., \hat{f}^d_\theta, \check{f}^0_\theta, ...\check{f}^d_\theta\}$,

where d is the number of hidden layers.

C. SPNs

SPNs are potentially deep acyclic directed graph consisting of sum and product nodes. Each internal node recursively calculates its value from the values of its child nodes: sum nodes calculate a *non-negatively weighted* sum of the values of their child nodes, where the non-negative weights are the parameters associated with the outgoing edges of the sum node. Product nodes calculate the product of their child nodes' values. While SPNs generally can have multiple roots [31], we assume SPNs with a single root. The value of the root node is the output of the SPN, while the input of the SPN is provided by its leave nodes. In [21], the leaves of an SPN were defined to be *indicator nodes* of discrete random variables, such that the SPN represents the *network polynomial* of a Bayesian network [66]. In [67], [24], [31] the concept of SPN leaves was generalized such that they represent *tractable* distributions over single random variables (RV) $X_m, m \in \{1, \dots M\}$, or (small) sets of RV $\mathbf{X}_I := \{X_m : m \in I\}$, where I denotes index set $I \subseteq \{1, \dots, M\}$ of M RVs. When $\mathbf{x}_I$ is an instantiation of $\mathbf{X}_I$ and N is a leave of an SPN, then the value of N for some input $\mathbf{x}$ is $N(\mathbf{x}) := p_N(\mathbf{x}_{\mathbf{sc}(N)})$, where the *scope* $\mathbf{sc}(N) \subseteq \{1, \dots, m\}$ are the indices of variables associated with N, and p_N is a tractable distribution over $\mathbf{X}_{\mathbf{sc}(N)}$. p_N can either be a probability mass function (PMF) or a probability density function. Generally, there are several leave nodes with the same scope, representing a collection of distributions over the same variables. This view of SPN leaves subsumes the definition using indicator nodes in [21], since an indicator function is a special case of a PMF, assigning all probability mass to a single state.

Concerning some internal node N, i.e. a sum or a product node, we define $\mathbf{sc}(N) := \bigcup_{C \in \mathbf{ch}(N)} \mathbf{sc}(C)$, where $\mathbf{ch}(N)$ denotes the children of N. Let R denote the root node of the SPN, and assume w.l.o.g. that $\mathbf{sc}(R) = \{1, \dots, M\}$. Then an SPN defines a probability distribution over $\mathbf{X}$ as $p_{SPN}(\mathbf{x}) \propto R(\mathbf{x})$, i.e. by its normalized output. In order to perform efficient inference (e.g. marginalization, most-probable explanation, conditional marginals) using forward and backprop passes [66], an SPN should be *valid* [21]. A sufficient condition for validity is when the SPN is *complete* and *decomposable*, defined as follows [21]:

- *Completeness*: For any two children C, C' of any sum node, it must hold that $\mathbf{sc}(C) = \mathbf{sc}(C')$.
- *Decomposability*: For any two children C, C' of any product node, it must hold that $\mathbf{sc}(C) \cap \mathbf{sc}(C') = \emptyset$.

When an SPN is complete and decomposable, and when the non-negative weights are normalized to 1 for each sum node [2] then the output is already normalized and $p_{SPN}(\mathbf{x}) = R(\mathbf{x})$. A complete and decomposable SPN can be naturally interpreted as a recursively defined distribution: product nodes serve as *cross-overs* of distributions with non-overlapping scope, where the product represents a local (context-specific) independence assumption; sum nodes represent *mixtures* of distributions, dissolving independence assumptions [24], [31]. Since sum nodes represent mixture distributions, one can associate a latent discrete random variable with each sum node, selecting the mixture component, where the associated weights can be interpreted as component priors [21]. The latent variable interpretation opens the door for the expectation-maximization algorithm, and variants thereof.

We use the approach in [21] to train the SPNs. Starting with the spectrogram data (i.e. the root rectangle), the algorithm recursively performs all decompositions into two sub-rectangles along the t and f dimensions, respectively, using a certain step size (resolution). Rectangles of size 1 (i.e. pixels) are not split further. The root rectangle is equipped with a single sum node, representing the distribution over all variables. Each non-root rectangle $\mathcal{R}$, containing more than one variable, is equipped with ρ sum nodes, representing ρ mixture distributions over the variables contained in $\mathcal{R}$. Each rectangle containing exactly one variable is equipped with γ *Gaussian* probability density nodes, which are the leaves of the SPN. The means of the Gaussian nodes are set to the γ quantile means of the corresponding variables, calculated from the training set, and the standard deviation is set to 1. If $\mathcal{R}'$ and $\mathcal{R}''$ are two rectangles generated by some split of $\mathcal{R}$, then for each combination of nodes N', N'', where N' comes from $\mathcal{R}'$ and N'' comes from $\mathcal{R}''$, a product node is generated and connected as parent of N' and N''. The so-generated product nodes are connected as child of each sum node in $\mathcal{R}$. The weights of this SPN are trained by a type of hard (winner-take-all) EM, with a sparseness penalty, penalizing evocation of non-zero weights.

III. Experiments: SCSS

A. Setup

We evaluate various models on a speaker dependent separation task (SD), a speaker independent separation task (SI), a matched noise separation task (MN), and an unmatched noise separation task (UN) using utterances of the 2nd CHiME speech separation challenge database [52] and the NOISEX corpus [68]. The 2nd CHiME database contains GRID corpus utterances [69] mixed with background noise, i.e. noise recorded in the kitchen and lounge. The noise contains speech and other sound patterns such as music, TV, and other noise types. CHiME consists of 34 speakers with 500 training samples each, and a validation- and test-set with 600 samples. In the SD and SI task original CHiME samples were used as data source. However, due to the lack of isolated noise signals needed to compute the source-specific spectrograms of the validation- and test set for evaluation purposes, disjoint subsets of the original training corpus were used for training and testing. In the SD task a model is applied only to speech mixtures from a specific speaker in the training and test case. In the SI setup, a model is trained on multiple speakers and tested on a disjoint subset of speakers. In this case the model is agnostic of the specific characteristic of the speaker. The MN and UN task, CHiME speech signals were mixed with noise variants from the NOISEX corpus i.e. for MN the same

[2] As shown in [67], this can be assumed w.l.o.g.

Ids {1,...,12} were chosen for both training and testing. In the UN task, the Ids {1,...,12} and {13,...,17} were selected for the training and testing, respectively. The MN task is rather unrealistic, but nevertheless it has been included to be comparable to [49]. Details about the task specific setup are listed in Table I. In [34], [49], a similar setup has been used.

task	database	speakers	# utterances/speaker		
			train	valid	test
SD	CHiME	4	400	50	50
SI	CHiME	10	50	5	5
MN	CHiME, NOISEX	10	40	5	5
UN	CHiME, NOISEX	10	40	5	5

TABLE I: Number of utterances used for training, validation and test.

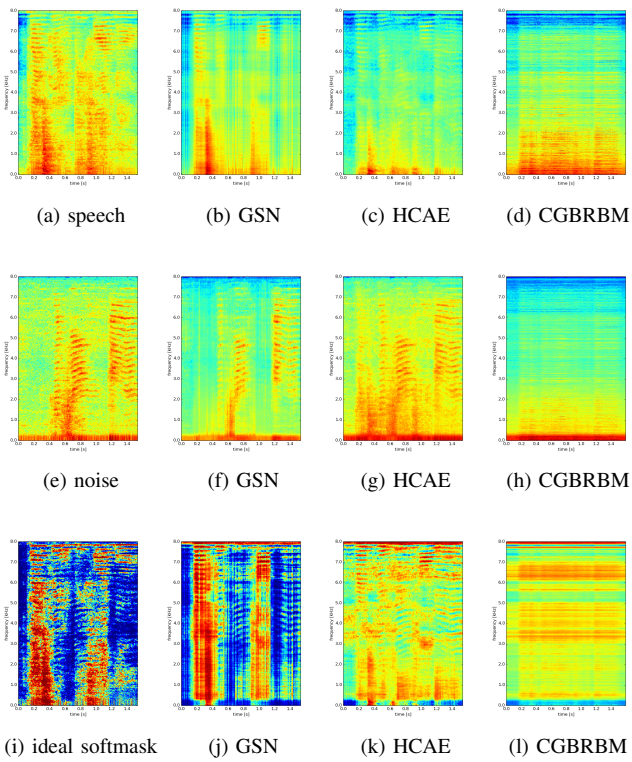
The time frequency representation was computed by a 1024 point Fourier transform using a Hamming window of 32ms length and a steps size of 10ms. The test set of all tasks was generated from 2 male, i.e. Ids {1, 2} and 2 female speakers, i.e. Ids {18, 20}. In the SD case different utterances from the same speakers were selected. In case of the SI, MN and UN setup, the training set consists of utterances selected from 5 male and 5 female i.e. Ids {3,5,6,9,10,4,7,8,11,15}. The training data was mixed at dB levels of {-6,-3,0,+3,+6,+9}. In the test case each model was evaluated separately using the whole test data remixed for every dB level.

For objective evaluation the overall perceptual score (OPS) [54] and the PESQ measure [53] are used. The OPS ranges between 0 and 100, where 100 is the best. Both, OPS and PESQ correlate rather well with subjective speech quality [54].

To find the optimal GSN configuration, a grid test on SD data was performed over $M \times d$, where $d \in \{1, 2, 3\}$ denotes the number of layers and $M \in \{256, 500, 1000, 2000, 3000\}$ is the number of neurons per layer. Sigmoid RBM- and HCAE variants were configured with network size of 2000×1. The optimal GSN is a 2000×2 network using rectifier activation functions and zero-mean Gaussian pre- and post activation noise with $\sigma = 0.1$, trained with $k = 2 \times d$ *walkback* steps. All models used linear downward activations in the first layer allowing to fully generate the zero-mean and unit variance normalized data. The network weights were initialized with an uniform distribution [70] and trained with early stopping. The mean-square-error was used as objective function for training all models using spectrogram frames as input, i.e. frame-wise processing is performed. The MLP uses rectifier activation functions.

B. Results

Figure 3 shows a reverberated clean speech spectrogram of the utterance "Place green in b 5 now", spoken by s20 (3a), a noise spectrogram (3e), and the computed optimal softmask (3i). Speech and noise are mixed at 0dB. Figure (3b), (3f) and (3j) show the reconstructions of speech and noise generated by two GSNs given the mixed signal (cf. Figure (3a)), and the resulting softmask. Figure (3c), (3g) and (3k) show the



(a) speech (b) GSN (c) HCAE (d) CGBRBM

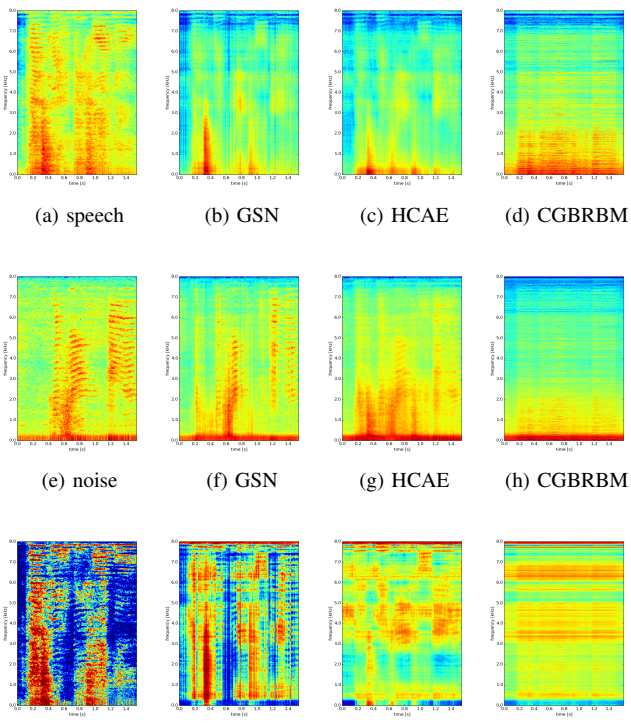
(e) noise (f) GSN (g) HCAE (h) CGBRBM

(i) ideal softmask (j) GSN (k) HCAE (l) CGBRBM

Fig. 3: Log-spectrograms of the utterance "Place green in b 5 now" spoken by s20, the noise, and the resulting soft-mask recovered by various frame-wise SD deep representation models. The first column shows the ideal softmask and the original noise and speech utterance. The remaining columns depict the reconstructions by GSNs, HCAEs, and CGBRBMs, respectively.

reconstructions of the HCAE and Figure (3d), (3h), (3l) the reconstructions generated by the CGBRBM, respectively. All models are optimized in a SD fashion. The harmonics in the noise shown in Figure (3e) and (4e) come from the lounge background, i.e. a second speech source is introduced from a TV in the background. The GSN obtains the most similar softmask compared to the optimal mask visually and in terms of the mean square error. The richer representation capability of GSNs is achieved by training multiple layers at the same time, i.e. the higher layers help the bottom layer to better model the input distribution. This is important if we have multi-modal input data, since a single layer might not be able to cover the higher-order statistics appropriately. The CGBRBM is not able to recover a meaningful temporal structure in the spectrogram.

Figure 4 shows a reverberated clean speech spectrogram of the same utterance as above (4a), a noise spectrogram (4e), and the computed optimal softmask (4i). Speech and noise are mixed at 0dB. In contrast to Figure 3 we apply SI models for SCSS. Figure (4b), (4f) and (4j) show the reconstructions of speech and noise generated by the GSN, given the mixed signal and the resulting softmask. Figure (4c), (4g) and (4k) show the reconstructions of the HCAE



(a) speech (b) GSN (c) HCAE (d) CGBRBM

(e) noise (f) GSN (g) HCAE (h) CGBRBM

(i) ideal softmask (j) GSN (k) HCAE (l) CGBRBM

Fig. 4: Log-spectrograms of the utterance "Place green in b 5 now" spoken by s20, the noise, and the resulting soft mask recovered by various frame-wise SI deep representation models. The first column shows the ideal softmask and the original noise and speech utterance. The remaining columns depict the reconstructions by GSNs, HCAEs, and CGBRBMs, respectively.

and Figure (4d), (4h), (4l) the reconstructions generated by the CGBRBM, respectively. Also for SI models the GSN outperforms the remaining representation models. GSNs have a richer model structure compared to the remaining representation models. Compared to SD models, the spectrograms obtained by HCAEs and GSNs appear to be more blurred, i.e. the harmonic structure of the original signals is not so well recovered. This is also reflected in the softmask.

The GSN mostly outperforms the other models with respect to the objective evaluation scores OPS and PESQ. This is shown in Table II, III, IV, V for the SD, SI, MN, and UN task and different dB conditions, respectively. Furthermore, we present OPS and PESQ scores for the mixed signal and the optimally separated signal using the ideal softmask (ISM). The *filtering* approach using two models is referenced via the subscript $M_1 + M_2$ (cf. Figure 1b). Approaches using a single model, e.g. the direct learning method, are referenced via the subscript M_1 (cf. Figure 1a). In all cases the ISM was inferred from the predicted components, except for the $\text{MLP}_{M_1}^{IRM}$ and $\text{MLP}_{M_1}^{IBM}$, where the ideal ratio mask (IRM) and ideal binary mask (IBM) was used. The ISM, IRM and IBM, studied in detail in [33], are defined as:

$$\text{ISM}(t,f) = \frac{\mathcal{S}(t,f)}{\mathcal{S}(t,f)+\mathcal{N}(t,f)}$$

$$\text{IRM}(t,f) = \left(\frac{\mathcal{S}^2(t,f)}{\mathcal{S}^2(t,f)+\mathcal{N}^2(t,f)}\right)^{\beta} \text{ with } \beta \in \mathbb{R}_+$$

$$\text{IBM}(t,f) = \begin{cases} 1 & \text{if ISM}(t,f) \geq 0.5 \\ 0 & \text{if ISM}(t,f) < 0.5 \end{cases}$$

In general, networks with multiple layers, i.e. 3-layer MLPs and 2-layer GSNs, outperform single layer networks. Having a closer look on the SD results in Table II we see that the frame-wise GSNs were able to outperform any other model including the discriminatively trained $\text{MLP}_{M_1}^{IRM}$ and also a MLPs predicting both $\mathcal{S}$ and $\mathcal{N}$ components within one single model, i.e. $\text{MLP}_{M_1}^{S+N}$ [71]. Interestingly the 3-layer $\text{MLPs}_{M_1+M_2}$, trained on $\mathcal{S}$ and $\mathcal{N}$ achieved lower OPS and PESQ scores than the generative 2-layer GSNs, i.e. GSN_{M_1+M2}. This indicates that the proposed *filtering* approach using two generative multi-layer networks slightly increases the perceptual quality in SCSS. Similar observations hold for Table III, IV and Table V.

Model	-6dB	-3dB	0dB	3dB	6dB	9dB
			PESQ			
mixed signal	1.11	1.19	1.25	1.37	1.54	1.77
MLP_{M_1}	1.29	1.43	1.62	1.85	2.14	2.47
$\text{MLP}_{M_1}^{IRM}$	1.27	1.38	1.55	1.77	2.04	2.35
$\text{MLP}_{M_1}^{S+N}$	1.25	1.39	1.58	1.80	2.06	2.35
$\text{MLP}_{M_1+M_2}$	1.72	1.96	2.22	2.42	2.64	2.84
$\text{CGBRBM}_{M_1+M_2}$	1.74	1.98	2.21	2.44	2.66	2.85
$\text{GBRBM}_{M_1+M_2}$	1.75	1.99	2.22	2.46	2.67	2.87
$\text{HCAE}_{M_1+M_2}$	1.77	2.01	2.38	2.60	2.80	3.01
$\text{GSN}_{M_1+M_2}$	**2.09**	**2.30**	**2.53**	**2.75**	**2.94**	**3.14**
optimal mask	4.50	4.50	4.50	4.50	4.50	4.50
			OPS			
mixed signal	9.07	9.86	11.17	13.56	17.14	21.62
MLP_{M_1}	32.99	36.13	39.39	42.14	44.73	46.63
$\text{MLP}_{M_1}^{IRM}$	30.59	33.93	35.51	36.22	37.14	39.58
$\text{MLP}_{M_1}^{S+N}$	31.83	36.39	39.66	41.37	44.32	47.38
$\text{MLP}_{M_1+M_2}$	25.25	26.76	29.31	30.47	32.32	35.54
$\text{CGBRBM}_{M_1+M_2}$	15.68	17.05	18.69	20.81	23.22	27.28
$\text{GBRBM}_{M_1+M_2}$	9.93	10.59	12.02	14.80	17.95	23.26
$\text{HCAE}_{M_1+M_2}$	12.20	24.42	25.72	26.69	27.92	31.07
$\text{GSN}_{M_1+M_2}$	**33.11**	**37.44**	**42.08**	**45.34**	**47.59**	**50.34**
optimal mask	98.89	98.89	98.89	98.89	98.89	98.89

TABLE II: PESQ and OPS results of SD task; Bold numbers denote best results for each specific noise level.

On the downside, far more computational resources are needed to compute an output reconstruction of a single frame with a GSN. In particular, the model needs $2 \cdot k \cdot d$ matrix-vector computations to pass the input information to higher layers and to compute the output reconstruction of a single frame, whereas a single layer HCAE needs 2 matrix-vector calculations, i.e. one for the upward pass and one for the downward pass. Nevertheless, reducing the bit-width [72], [73], [74] might allow to implement the proposed method on computationally constrained systems.

We also performed informal listening tests confirming the results, i.e. utterances processed by GSNs sound more natural and suppress the noise in a better way than the other methods. Nevertheless, the performance gap to the optimal mask reveals that there is still significant improvement possible.

Model	-6dB	-3dB	0dB	3dB	6dB	9dB
			PESQ			
mixed signal	1.37	1.65	1.81	2.07	2.38	2.59
MLP_{M_1}	1.13	1.21	1.35	1.50	1.70	1.97
$\text{MLP}_{M_1+M_2}$	1.50	1.69	1.90	2.12	2.43	2.64
$\text{CGBRBM}_{M_1+M_2}$	1.37	1.64	1.90	2.12	2.43	2.64
$\text{GBRBM}_{M_1+M_2}$	1.45	1.70	1.93	2.16	2.44	2.65
$\text{HCAE}_{M_1+M_2}$	1.51	1.75	1.99	2.22	2.52	2.71
$\text{GSN}_{M_1+M_2}$	**1.62**	**1.87**	**2.06**	**2.29**	**2.55**	**2.75**
optimal mask	4.50	4.50	4.50	4.50	4.50	4.50
			OPS			
mixed signal	10.02	10.59	12.45	14.20	16.70	21.88
MLP_{M_1}	29.08	32.91	34.91	36.73	38.06	39.34
$\text{MLP}_{M_1+M_2}$	10.40	11.02	12.27	14.29	17.44	22.74
$\text{CGBRBM}_{M_1+M_2}$	10.16	11.14	12.64	14.24	17.13	22.57
$\text{GBRBM}_{M_1+M_2}$	9.81	11.15	12.60	14.21	16.96	22.13
$\text{HCAE}_{M_1+M_2}$	13.06	13.51	14.68	15.63	17.23	20.28
$\text{GSN}_{M_1+M_2}$	**29.25**	**33.50**	**38.39**	**42.22**	**43.21**	**45.84**
optimal mask	98.89	98.89	98.89	98.89	98.89	98.89

TABLE III: PESQ and OPS results of SI task; Bold numbers denote best results for each specific noise level.

Model	-6dB	-3dB	0dB	3dB	6dB	9dB
			PESQ			
mixed signal	1.50	1.70	1.90	2.12	2.43	2.64
MLP_{M_1}	1.48	1.67	1.86	2.07	2.31	2.63
$\text{MLP}_{M_1+M_2}$	1.85	2.04	2.20	2.40	2.61	2.84
$\text{CGBRBM}_{M_1+M_2}$	1.63	1.85	2.05	2.28	2.48	2.66
$\text{GBRBM}_{M_1+M_2}$	1.72	1.90	2.09	2.32	2.52	2.70
$\text{HCAE}_{M_1+M_2}$	1.82	1.96	2.19	2.36	2.55	2.72
$\text{GSN}_{M_1+M_2}$	**2.23**	**2.44**	**2.63**	**2.85**	**3.06**	**3.24**
optimal mask	4.50	4.50	4.50	4.50	4.50	4.50
			OPS			
mixed signal	9.44	10.39	12.36	14.23	17.03	22.15
MLP_{M_1}	33.50	**37.43**	**41.84**	**45.82**	**51.73**	56.46
$\text{MLP}_{M_1+M_2}$	30.44	32.98	35.37	38.24	41.35	48.31
$\text{CGBRBM}_{M_1+M_2}$	9.91	10.78	12.60	14.87	19.16	33.00
$\text{GBRBM}_{M_1+M_2}$	10.05	10.67	12.76	15.45	19.99	33.92
$\text{HCAE}_{M_1+M_2}$	21.79	23.42	25.27	29.90	36.66	45.15
$\text{GSN}_{M_1+M_2}$	**34.34**	36.92	40.53	44.82	50.93	**59.32**
optimal mask	98.89	98.89	98.89	98.89	98.89	98.89

TABLE IV: PESQ and OPS results of MN task; Bold numbers denote best results for each specific noise level.

Model	-6dB	-3dB	0dB	3dB	6dB	9dB
			PESQ			
mixed signal	1.61	1.83	1.95	2.15	2.35	2.56
MLP_{M_1}	1.09	1.13	1.20	1.33	1.52	1.80
$\text{MLP}_{M_1+M_2}$	1.64	1.84	2.02	2.20	2.40	2.63
$\text{CGBRBM}_{M_1+M_2}$	1.67	1.86	2.04	2.21	2.40	2.62
$\text{GBRBM}_{M_1+M_2}$	1.65	1.85	2.05	2.23	2.43	2.64
$\text{HCAE}_{M_1+M_2}$	**1.68**	1.79	2.00	2.21	2.38	2.58
$\text{GSN}_{M_1+M_2}$	**1.68**	**1.88**	**2.07**	**2.40**	**2.51**	**2.75**
optimal mask	4.50	4.50	4.50	4.50	4.50	4.50
			OPS			
mixed signal	13.93	16.08	19.58	22.54	27.89	33.77
MLP_{M_1}	24.45	28.33	32.24	37.41	42.12	46.72
$\text{MLP}_{M_1+M_2}$	26.24	30.80	35.93	39.60	42.93	47.17
$\text{CGBRBM}_{M_1+M_2}$	13.06	14.83	18.28	22.23	27.71	34.26
$\text{GBRBM}_{M_1+M_2}$	12.98	14.91	18.49	22.43	28.31	34.96
$\text{HCAE}_{M_1+M_2}$	22.99	26.64	30.43	33.47	36.59	40.09
$\text{GSN}_{M_1+M_2}$	**26.63**	**31.44**	**36.46**	**40.86**	**45.39**	**50.42**
optimal mask	98.89	98.89	98.89	98.89	98.89	98.89

TABLE V: PESQ and OPS results of UN task; Bold numbers denote best results for each specific noise level.

In many speech applications the use of context information is beneficial to further improve the classification performance [35]. Therefore, we present results for the SD task using GSNs with varying number of spectrogram frames as input in Table VI. While we limited models of the previous experiments to a single input frame, we use here also 3, 5, and 7 spectrogram frames as input. This provides additional temporal information for the deep GSN representation model. However, we still do not model the temporal information explicitly, only up to 7 consecutive frames are used as input to *filter* the output frame. For the SD task, the best results are obtained with 5 frames closely followed by 3 frames. In [34], we also discovered that 5 input frames lead to the best estimation of the softmask, when using a STFT with a Hamming window of 32ms length and a steps size of 10ms.

Model	Frames	-6dB	-3dB	0dB	3dB	6dB	9dB
				PESQ			
$\text{GSN}_{M_1+M_2}$	1	2.09	2.30	2.53	2.75	2.94	3.14
$\text{GSN}_{M_1+M_2}$	3	2.18	2.38	2.59	2.80	2.98	3.17
$\text{GSN}_{M_1+M_2}$	5	**2.19**	**2.41**	**2.62**	**2.82**	**2.99**	**3.18**
$\text{GSN}_{M_1+M_2}$	7	1.97	2.21	2.42	2.62	2.81	3.00
				OPS			
$\text{GSN}_{M_1+M_2}$	1	33.11	37.44	42.08	45.34	47.59	50.34
$\text{GSN}_{M_1+M_2}$	3	33.85	38.40	43.11	**47.84**	**52.59**	**56.30**
$\text{GSN}_{M_1+M_2}$	5	**34.62**	**39.31**	**43.69**	47.15	51.09	54.72
$\text{GSN}_{M_1+M_2}$	7	33.79	36.99	39.45	40.22	42.41	44.41

TABLE VI: PESQ and OPS results of SD task using 1, 3, 5, or 7 frames as input in $\text{GSN}_{M_1+M_2}$; Bold numbers denote best results for each specific noise level.

In order to evaluate different masks $\mathcal{G}$ we trained a 3-layer MLP_{M_1} using IBMs, ISMs and IRMs. Table VII shows the results for SD data using 5 consecutive spectrogram frames. The softmask achieved the best PESQ and the IRM mask using $\beta = 2$ achieved the best OPS results. Table VIII shows PESQ and OPS scores for IRM masks and different β values.

Mask	-6dB	-3dB	0dB	3dB	6dB	9dB
			PESQ			
mixed signal	1.11	1.19	1.25	1.37	1.54	1.77
IBM	1.35	1.51	1.72	1.99	2.03	2.63
IRM	1.26	1.40	1.60	1.87	2.20	2.56
ISM	**1.38**	**1.57**	**1.79**	**2.05**	**2.34**	**2.64**
optimal mask	4.50	4.50	4.50	4.50	4.50	4.50
			OPS			
mixed signal	9.07	9.86	11.17	13.56	17.14	21.62
IBM	29.77	33.05	36.05	40.95	45.19	49.39
IRM	**33.30**	**38.35**	**42.83**	**45.84**	**48.47**	**52.67**
ISM	32.50	35.91	39.18	41.01	42.35	44.40
optimal mask	98.89	98.89	98.89	98.89	98.89	98.89

TABLE VII: PESQ and OPS results of a 3-layer MLP trained on different masks on a SD task using an 5 consecutive spectrogram frames; Bold numbers denote best results for each specific noise level.

β	-6dB	-3dB	0dB	3dB	6dB	9dB
			PESQ			
0.10	1.12	1.17	1.28	1.41	1.62	1.89
0.25	1.17	1.25	1.38	1.56	1.81	2.13
0.50	1.21	1.32	1.48	1.69	1.98	2.32
0.75	1.24	1.38	1.58	1.86	2.19	2.55
1.00	1.25	**1.40**	**1.61**	**1.88**	**2.21**	**2.57**
1.50	**1.26**	**1.40**	1.60	1.87	2.20	2.56
2.00	**1.26**	**1.40**	1.60	1.87	2.20	2.56
			OPS			
0.10	9.41	9.23	10.58	12.17	15.22	20.34
0.25	22.67	23.50	23.52	22.61	24.05	26.84
0.50	30.65	32.64	34.49	34.02	35.57	37.04
0.75	32.71	36.66	40.61	43.61	45.95	48.64
1.00	33.32	38.00	41.79	45.43	47.46	49.17
1.50	**33.91**	**38.56**	41.86	44.71	48.15	50.56
2.00	33.30	38.35	**42.83**	**45.84**	**48.47**	**52.67**

TABLE VIII: PESQ and OPS results of a 3-layer MLP trained with the IRM mask on a SD task using an 5 consecutive spectrogram frames and different β values; Bold numbers denote best results for each specific noise level.

IV. Experiments: ABE

A. Setup

We performed our experiments on the GRID corpus [69], where we used the test speakers with numbers 1, 2, 18, and 20 (similar as in the SCSS task), referred to as s1, s2, s18, and s20, respectively. Speakers s1 and s2 are male, and s18 and s20 are female. We trained SD and SI models. For speaker dependent models we used 10 minutes of speech of the respective speaker. For speaker independent models we used 10 minutes of speech obtained from the remaining 30 speakers of the corpus, each speaker providing approximately 20 seconds of speech. For testing we used 50 utterances per test speaker, not included in the training set. We simulate narrow-band telephone speech [75] by applying a bandpass filter with stop frequencies 50 Hz and 4000 Hz.

As baseline we use the method proposed in [31]. We used HMM-GMMs with 256 components per state dependent GMM with diagonal covariance matrices and $L = 64$ states. For training GMMs, we used the EM algorithm for maximal 100 iterations and 3 random restarts. Inference using the GMM model works the same way as for SPNs (see below), since a GMM can be formulated as an SPN with a single sum node [31]. We refer to this baseline as HMM-GMM. The SPNs are used in a similar fashion.[3] The observation probabilities in the HMM [55] are replaced by SPNs, i.e. each state has an individual SPN. An illustration of this modified HMM is shown in Figure 5. To recover the full-band log-magnitude spectrogram from the narrow-band telephone spectrogram inference in the HMM is required. Let $\mathcal{S}(t,f)$ be the f^{th} frequency bin of the t^{th} time-frame of the full-band signal, $t \in \{1,\dots,T\}$, $f \in \{1,\dots,F\}$, where F is the number of frequency bins and $\mathcal{S}_t = (\mathcal{S}(t,1),\dots,\mathcal{S}(t,F))^T$. Further, $\bar{\mathcal{S}}(t,f)$ are the time-frequency bins of the telephone filtered signal, and $\bar{\mathcal{S}}_t = (\bar{\mathcal{S}}(t,1),\dots,\bar{\mathcal{S}}(t,F))^T$. Within the telephone band, we can assume that $\mathcal{S}(t,f) \approx \bar{\mathcal{S}}(t,f)$, while some of the lowest and the upper half of the frequency bins in $\bar{\mathcal{S}}_t$ are lost. We use *most-probable-explanation* (MPE) [21] inference for recovering the missing spectrogram content, where we reconstruct the high-band only. Let $\hat{\mathcal{S}}_{t,l} = (\hat{\mathcal{S}}_{t,l}(1),\dots\hat{\mathcal{S}}_{t,l}(F))^T$ be the MPE-reconstruction of the t^{th} time frame, using the SPN depending on the l^{th} HMM-state. Then we use the following bandwidth-extended log-magnitude spectrogram

$$\hat{\mathcal{S}}(t,f) = \begin{cases} \bar{\mathcal{S}}(t,f) & \text{if } f \in \text{telephone band} \\ \sum_{l=1}^{L} p(Y_t = l|\mathbf{e}_t)\hat{\mathcal{S}}_{t,l}(f) & \text{otherwise} \end{cases}. \tag{1}$$

The recovered spectrogram bins are a weighted sum over all l MPE reconstructions $\hat{\mathcal{S}}_{t,l}$. The weights are the marginals $p(Y_t|\mathbf{e}_t)$ provided by the forward-backward algorithm [57], where Y_t is the hidden state variable at the t^{th} time frame and $\mathbf{e}_t$ is the observed data up to time frame t plus look-ahead λ, i.e. all frequency bins in the telephone band, for all time frames $1,\dots,(t+\lambda)$. The observation likelihoods provided by the SPNs for each state Y_t are determined requiring marginalization of the missing frequency bins in the narrow-band telephone signal $\bar{\mathcal{S}}_t$. This can be efficiently done in SPNs, i.e Gaussian density function nodes corresponding to unobserved frequency bins, constantly return value 1 and these variables are marginalized in the upward-pass.

[3]Results of the HMM-GMM and the HMM-SPN approach have previously been published in [25].

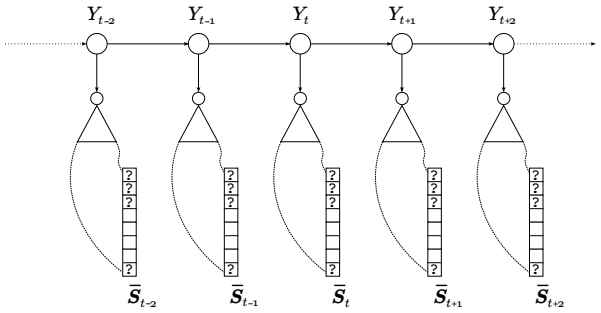


Fig. 5: HMM with SPN observation models. State-dependent SPNs are symbolized by triangles with a circle on top. For the forward-backward algorithm, frequency bins marked with "?" (missing) are marginalized out by the SPNs.

For the other representational models we take a different approach for learning; here we interpret the telephone band as a noisy version of the full band. The telephone band is fed into the network and the reconstruction is compared to the signal of full bandwidth, i.e. a mapping from telephone band to the full band is modeled. This leads to a matching setup for training and testing and also prevents the networks from learning the identity function of clean data. In addition, we only consider frame-wise models trained on isolated frames. In particular, we again use the GSN, the RBM variants, the HCAE and also a discriminatively trained 3-layer MLP as a baseline system. The optimal configuration of the representation models has been determined in the same way as for the SCSS task.

We used time-frames of 512 samples length, with 75% overlap, which using a sampling frequency of $f_s = 16\,\text{kHz}$ corresponds to a frame length of $32\,\text{ms}$ and a frame rate of $125\,\text{Hz}$. Before applying the FFT, the frames were weighted with a Hamming window. For the forward-backward algorithm in the HMMs we used a look-ahead of $\lambda = 3$ frames, which corresponds to the minimal delay introduced by the 75% frame-overlap.

For re-synthesizing time signals from the log-magnitude spectrogram reconstructions we use the same approach as in [25], using 100 iterations of the Griffin&Lim algorithm [76] to synthesize phase for the reconstructed time-frequency bins. We replace the predicted telephone band by the original telephone band in all models and initialize Griffin&Lim with the corresponding phase. In [55], the phase from the telephone band is also taken for the high-band. This is also an alternative for computationally constrained systems.

B. Evaluation Objectives

For objective evaluation, we use the log-spectral distortion (LSD) in the high-band, similar as in [55], [25]. Using 9^{th} order LPC analysis of each frame we get the spectral envelope

as

$$E_{\mathbf{a}}(e^{j\Omega}) = \frac{\sigma}{|\sum_{k=0}^{9} a_k e^{-jk\Omega}|}, \tag{2}$$

where σ is the square-root of the variance of the LPC-analyzed signal and $\mathbf{a} = (a_0, \ldots, a_9)$ are the LPC coefficients. The high-band LSD in [dB] for the τ^{th} frame is computed as

$$\text{LSD}_\tau = \sqrt{\frac{\int_\nu^\pi \left(20 \log E_{\mathbf{a}_\tau}(e^{j\Omega}) - 20 \log E_{\hat{\mathbf{a}}_\tau}(e^{j\Omega})\right)^2 d\Omega}{\pi - \nu}}, \tag{3}$$

where $\nu = \pi \frac{4000}{f_s/2}$, f_s is the sampling frequency, and $\mathbf{a}_\tau$ and $\hat{\mathbf{a}}_\tau$ are the LPC coefficients of the τ^{th} frame of the original and reconstructed signal, respectively. We report the utterance LSD given as the average of LSD_τ over all frames. Furthermore, we compute the unweighted segmental SNR (sSNR) in the frequency-domain [77], limited in the range of [-10dB, 35dB].

In a detailed analysis we also considered wide-band PESQ [53] (WB-PESQ), which provides an instrumental prediction for the mean opinion score (MOS) to show the improvement obtained by the proposed methods compared to the narrow-band telephone signal and original wide-band signal. It was reported that WB-PESQ correlates well with subjective test results [78].

All evaluation measures are determined from the re-synthesized time-domain signals.

C. Results

Figure 6 shows log-magnitude spectrograms of the clean test utterance of speaker s20 in (6a), the bandwidth extended signals of SD deep frame-wise representation models, i.e. HCAEs (6b), GSNs (6c), and CGBRBMs (6d) and the reconstructions of the HMM-GMM (6e), HMM-SPN (6f). These log-magnitude spectrograms are determined from the synthesized time-domain signals of the bandwidth extended log-magnitude spectrogram, i.e. artifacts from recovering the phase and of the weighted overlap-add method are included.

The frame-wise GSN model is able to reproduce the missing high frequency components in a better way than HCAE and CGBRBM. The HCAE produces a strongly smoothed spectrogram of the high frequency bands. The CGBRBM fails to produce a meaningful reconstruction of the high frequency band. This has also impact on the telephone band during recovery of the time-domain signal. The reconstruction by HMM-GMM is blurry and does not recover the harmonic structure of the original signal well, but partly recovers high-frequency content related to consonants. The HMM-SPN approach is obtaining a more natural high frequency reconstruction. Similarly, the GSN method recovers a high frequency structure, which largely resembles the original full-band signal: the harmonic structure appears more natural than the one delivered by HMMs and consonant sounds seem to be better detected and reconstructed than by HMM-GMMs. Interestingly, in this example the frame-wise GSN recovers the most similar spectrogram without explicit temporal modeling. According to informal listening tests, the visual impression corresponds to the listening experience: the signals delivered by HMM-SPNs and GSNs clearly enhance the high-frequency content and sound more natural than the signals delivered by HMM-GMMs, HCAEs, and RBMs. The SPN and GSN variants produce a more realistic extension for fricative and plosive sounds.

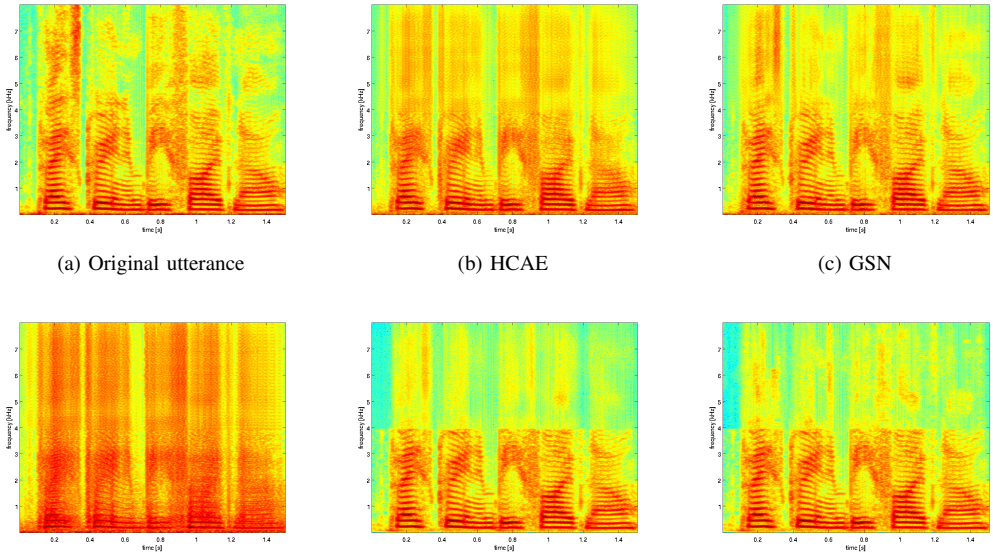
Model	s1	s2	s18	s20	avg.
			LSD [dB]		
narrow-band	6.98	7.58	6.66	6.48	6.93
MLP_{M_1}	5.15	6.44	5.50	4.16	5.31
GBRBM_{M_1}	16.34	10.66	6.50	6.11	9.90
CGBRBM_{M_1}	3.72	3.91	3.74	3.50	3.71
$\text{HMM} - \text{GMM}$	3.18	2.93	2.28	2.82	2.80
$\text{HMM} - \text{SPN}$	3.12	**2.84**	2.15	**2.59**	**2.68**
HCAE_{M_1}	**3.07**	3.70	3.43	3.38	3.39
GSN_{M_1}	3.50	2.88	**2.12**	2.75	2.81
			sSNR [dB]		
narrow-band	1.62	1.64	8.23	6.77	4.56
MLP_{M_1}	3.52	3.60	9.50	9.74	6.59
GBRBM_{M_1}	-4.02	-2.75	1.12	10.07	1.10
CGBRBM_{M_1}	3.82	3.94	10.53	9.95	7.06
$\text{HMM} - \text{GMM}$	3.95	4.14	10.33	9.81	7.06
$\text{HMM} - \text{SPN}$	3.52	3.86	9.73	9.67	6.95
HCAE_{M_1}	3.82	3.95	10.45	9.97	7.05
GSN_{M_1}	**7.95**	**8.43**	**13.72**	**13.35**	**10.85**

TABLE IX: Log spectral distortion (LSD) and segmental signal-to-noise-ratio (sSNR) for SD frame-wise and HMM-based models. The narrow-band baseline is included. Bold numbers denote best results for each speaker.

Model	s1	s2	s18	s20	avg.
			LSD [dB]		
narrow-band	6.98	7.58	6.66	6.48	6.96
MLP_{M_1}	5.05	6.88	6.09	4.71	5.68
GBRBM_{M_1}	6.67	7.29	6.50	6.15	6.65
CGBRBM_{M_1}	4.18	4.87	4.26	6.15	4.87
$\text{HMM} - \text{GMM}$	3.62	4.46	3.82	3.60	3.88
$\text{HMM} - \text{SPN}$	**3.42**	3.85	**3.05**	**3.36**	**3.32**
HCAE_{M_1}	3.81	4.05	3.95	**3.36**	3.79
GSN_{M_1}	3.88	**3.77**	3.17	3.55	3.59
			sSNR [dB]		
narrow-band	-0.32	0.57	3.91	5.19	2.34
MLP_{M_1}	3.55	3.44	8.74	9.38	6.27
GBRBM_{M_1}	3.92	4.13	10.73	10.07	7.21
CGBRBM_{M_1}	3.89	4.08	10.59	10.00	7.14
$\text{HMM} - \text{GMM}$	3.69	3.88	10.23	9.81	6.90
$\text{HMM} - \text{SPN}$	3.54	3.80	9.94	9.65	6.73
HCAE_{M_1}	3.85	3.97	10.55	9.96	7.08
GSN_{M_1}	**7.97**	**8.27**	**13.73**	**13.31**	**10.81**

TABLE X: Log spectral distortion (LSD) and segmental signal-to-noise-ratio (sSNR) for SI frame-wise and HMM-based models. The narrow-band baseline is included. Bold numbers denote best results for each speaker.

The LSD and sSNR values of the frame-wise and HMM-based models for the SD and SI tasks averaged over the 50 test sentences are shown in Table IX and X, respectively. Furthermore, we added the LSD and sSNR for the narrow-band signal. When looking at the LSD scores on both SI and SD tasks, the HMM-SPN slightly outperforms the GSN. However a pair-wise t-test with $p = 0.05$ reveals no significance in this difference. Furthermore, as mentioned above we replaced the predicted telephone band by the original one. This results in slightly lower LSD values for GSNs, but removes audible artifacts. In particular, the predicted full bandwidth signals of GSNs achieve an average LSD of 2.15dB and 2.93dB compared to 2.81dB and 3.59dB for the SD and SI task. The GSN achieved an average improvement of 3.90dB



(a) Original utterance (b) HCAE (c) GSN

(d) CGBRBM (e) HMM-GMM (f) HMM-SPN

Fig. 6: Log-magnitude spectrogram of the utterance "Place green in b 5 now", spoken by s20 recovered by various frame-wise SD deep representation models and hybrid HMM variants: (6a) original full bandwidth signal, (6b) HCAE, (6c) GSN, (6d) CGBRBM, (6e) HMM-GMM, (6f) HMM-SPN.

in sSNR compared to the HMM-SPN in the SD task. In case of the SI task the average improvement between both models in sSNR is 4.08dB. Both improvements are statistically significant when using a t-test with $p = 0.1$. Most notably, the HMM-SPN consists of 64 sub-models connected to one HMM, whereas the GSN uses a single model and simple frame-wise processing. The GSN also clearly outperforms the GBRBM and CGBRBM variants, due to its ability to handle multi-modal input data distributions. The GSN jointly optimizes multiple network layers at the same time. Higher layers contribute to the modeling process of the lower layers. We conjecture that this is a key reason for the good overall performance of the model when compared to single layer network variants, such as HCAEs and CGBRBMs.

The average WB-PESQ is above 4.36 for the SPN-HMM, HCAE and GSN models, with no statistical significant differences. The narrow-band signal achieved an average WB-PESQ of 4.35 and the full-band signal 4.5. Therefore, we do not report detailed WB-PESQ scores as the improvements in the high frequencies are not well covered in the score and differences are neglectable.

Similar as for SCSS, we show results for the SI task using GSNs with varying number of spectrogram frames as input in Table XI. We use in addition to single-frame input also 3 and 5 spectrogram frames. This provides implicit temporal information for the GSN model. We use for all cases the same GSN setup as in the single-frame experiment. The sSNR performance slightly improves with additional frames. We assume that the underlying reason is that the optimal GSN has been determined for only one input frame. However a t-test with p=0.05 reveals that all differences are not statistically significant.

Model	Frames	s1	s2	s18	s20	avg.
		LSD [dB]				
GSN_{M_1}	1	3.88	3.77	3.17	3.55	**3.59**
GSN_{M_1}	3	**3.68**	3.57	**3.07**	**3.05**	3.34
GSN_{M_1}	5	3.73	**3.57**	3.17	3.15	3.40
		sSNR [dB]				
GSN_{M_1}	1	7.97	8.27	13.73	13.31	10.82
GSN_{M_1}	3	**8.77**	**9.03**	**14.43**	**14.00**	**11.55**
GSN_{M_1}	5	**8.77**	9.05	14.35	13.99	11.54

TABLE XI: Log spectral distortion for frame-based SI GSN models using 1, 3, and 5 frames as input; Bold numbers denote best results for each specific noise level.

V. Conclusion and Future Work

We compared several generative representation learning approaches for two speech processing applications: single-channel source separation (SCSS) and artificial bandwidth extension (ABE). We use popular models from representation learning including Gauss Bernoulli restricted Boltzmann machines (RBMs), conditional Gauss Bernoulli restricted Boltzmann machines, higher order contractive autoencoders (AEs), sum-product networks (SPNs), and generative stochastic networks (GSNs). Furthermore, a rectifier multilayer perceptron is applied to the SCSS and ABE tasks. For SCSS we applied a two model *filtering* approach, i.e. we train each model separately on mixed spectrograms *and* the corresponding source representations. GSNs outperform the other models on speaker dependent and speaker independent SCSS. The GSN uses a hierarchical layer structure which is jointly optimized. For ABE we use in addition to simple frame-wise reconstruction of the missing high frequency part in the spectrograms also SPNs and GMMs embedded in an HMM. The frame-wise GSN significantly outperforms the hybrid HMM-based models and

the other representation models in terms of segmental SNR.

Future work includes the extension of the SCSS task by modeling the temporal information explicitly. Furthermore, we aim to use more realistic data and larger data sets to obtain a better generalization of the learned representations. This also includes models with more hidden layers. This is in particular important for the SI tasks, where the speakers in the test set are not included in the training data.

REFERENCES

[1] Y. Bengio, "Learning deep architectures for AI," *Foundations and Trends in Machine Learning*, vol. 2, no. 1, pp. 1–127, 2009.

[2] D. Yu and L. Deng, "Deep learning and its applications to signal and information processing," *IEEE Signal Processing Magazine*, vol. 28, no. 1, pp. 145–154, 2011.

[3] G. Hinton, L. Deng, G. E. Dahl, A. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. Sainath, and B. Kingsbury, "Deep neural networks for acoustic modeling in speech recognition." *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 82–97, 2012.

[4] Y. Bengio, A. C. Courville, and P. Vincent, "Representation learning: A review and new perspectives." *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 8, pp. 1798–1828.

[5] G. Dahl, M. Ranzato, A. Mohamed, and G. Hinton, "Phone recognition with the mean-covariance restricted boltzmann machine," in *Neural Information Processing Systems (NIPS)*, 2010, pp. 469–477.

[6] L. Deng, M. Seltzer, D. Yu, A. Acero, A. Mohamed, and G. Hinton, "Binary coding of speech spectrograms using a deep auto-encoder." in *Interspeech*, 2010, pp. 1692–1695.

[7] F. Seide, G. Li, and D. Yu, "Conversational speech transcription using context-dependent deep neural networks." in *Interspeech*, 2011, pp. 437–440.

[8] R. Sarikaya, G. Hinton, and A. Deoras, "Application of deep belief networks for natural language understanding," *IEEE Transactions on Audio Speech and Language Processing*, vol. 22, no. 4, pp. 778–784, 2014.

[9] Y. Bengio, E. Thibodeau-Laufer, and J. Yosinski, "Deep generative stochastic networks trainable by backprop," in *International Conference on Machine Learning (ICML)*, 2014.

[10] M. Zöhrer and F. Pernkopf, "Representation models in single channel source separation," in *International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2015.

[11] G. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors," *CoRR*, 2012.

[12] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets." *Neural computation*, vol. 18, no. 7, pp. 1527–1554, 2006.

[13] K. Cho, A. Ilin, and T. Raiko, "Improved learning of Gaussian-Bernoulli restricted Boltzmann machines," in *International Conference on Artificial Neural Networks (ICANN)*, 2011, pp. 10–17.

[14] Y. Bengio, P. Lamblin, D. Popovici, and H. Larochelle, "Greedy layer-wise training of deep networks," in *Neural Information Processing Systems (NIPS)*, 2007, pp. 153–160.

[15] P. Vincent, H. Larochelle, Y. Bengio, and P. Manzagol, "Extracting and composing robust features with denoising autoencoders," in *International Conference on Machine learning (ICML)*, 2008, pp. 1096–1103.

[16] S. Rifai, G. Mesnil, P. Vincent, X. Muller, Y. Bengio, Y. Dauphin, and X. Glorot, "Higher order contractive auto-encoder," in *European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML PKDD)*, 2011.

[17] H. Lee, A. Battle, R. Raina, and A. Ng, "Efficient sparse coding algorithms," in *Neural Information Processing Systems (NIPS)*, 2007, p. 801.

[18] M. Ranzato, C. Poultney, S. Chopra, and Y. LeCun, "Efficient learning of sparse representations with an energy-based model," in *Neural Information Processing Systems (NIPS)*, vol. 19, 2007, pp. 1137–1144.

[19] G. Hinton and R. Salakhutdinov, "Reducing the dimensionality of data with neural networks." *Science*, vol. 313, no. 5786, pp. 504–507, 2006.

[20] M. Zöhrer and F. Pernkopf, "General stochastic networks for classification," *Neural Information Processing Systems (NIPS)*, 2014.

[21] H. Poon and P. Domingos, "Sum-product networks: A new deep architecture," in *Uncertainty in Artificial Intelligence (UAI)*, 2011, pp. 337–346.

[22] A. Dennis and D. Ventura, "Learning the architecture of sum-product networks using clustering on variables," in *Neural Information Processing Systems (NIPS)*, 2012, pp. 2042–2050.

[23] R. Gens and P. Domingos, "Discriminative learning of sum-product networks," in *Neural Information Processing Systems (NIPS)*, 2012, pp. 3248–3256.

[24] ——, "Learning the structure of sum-product networks," in *International Conference on Machine learning (ICML)*, 2013, pp. 873–880.

[25] R. Peharz, G. Kapeller, P. Mowlaee, and F. Pernkopf, "Modeling speech with sum-product networks: Application to bandwidth extension," in *International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2014.

[26] A. Rooshenas and D. Lowd, "Learning sum-product networks with direct and indirect variable interactions," *International Conference on Machine learning (ICML)*, pp. 710–718, 2014.

[27] Y. Bengio, L. Yao, G. Alain, and P. Vincent, "Generalized denoising auto-encoders as generative models," in *Neural Information Processing Systems (NIPS)*, vol. 26, 2013, pp. 899–907.

[28] S. Ozair, L. Yao, and Y. Bengio, "Multimodal transitions for generative stochastic networks." *CoRR*, vol. abs/1312.5578, 2013.

[29] D. Koller and N. Friedman, *Probabilistic Graphical Models: Principles and Techniques*. MIT Press, 2009.

[30] F. Pernkopf, R. Peharz, and S. Tschiatschek, *Introductin to Probabilistic Graphical Models*. Academic Press Library in Signal Processing, vol. 1, ch. 18,, 2014.

[31] R. Peharz, B. Geiger, and F. Pernkopf, "Greedy part-wise learning of sum-product networks," in *European Conference on Machine Learning (ECML)*, 2013, pp. 612–627.

[32] G. Taylor and G. Hinton, "Factored conditional restricted Boltzmann machines for modeling motion style," *International Conference on Machine Learning (ICML)*, pp. 1025–1032, 2009.

[33] Y. Wang, A. Narayanan, and D. Wang, "On training targets for supervised speech separation," *IEEE/ACM Transactions on Audio, Speech and Language Processing*, vol. 22, no. 12, pp. 1849–1858, 2014.

[34] M. Zöhrer and F. Pernkopf, "Single channel source separation with general stochastic networks," in *International Conference on Spoken Language Processing (Interspeech)*, 2014, pp. –.

[35] Y. Xu, J. Du, L.-R. Dai, and C.-H. Lee, "A regression approach to speech enhancement based on deep neural networks," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 23, no. 1, pp. 7–19, 2015.

[36] K. Li and C.-H. Lee, "A deep neural network approach to speech bandwidth expansion," in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2015.

[37] S. Roweis, "Factorial models and refiltering for speech separation and denoising," in *EUROSPEECH*, 2003, pp. 1009–1012.

[38] D. D. Lee and H. Seung, "Learning the parts of objects by nonnegative matrix factorization," *Nature*, vol. 401, pp. 788–791, 1999.

[39] P. Smaragdis, "Convolutive speech bases and their application to supervised speech separation," *IEEE Trans. Audio Speech and Language Processing*, vol. 15, no. 1, pp. 1–12, 2007.

[40] R. Peharz and F. Pernkopf, "Sparse Nonnegative Matrix Factorization with ℓ^0 Constraints," *Neurocomputing, accepted*, vol. 80, pp. 38–46, 2012.

[41] R. Peharz, M. Stark, F. Pernkopf, and Y. Stylianou, "A factorial sparse coder model for single channel source separation," in *Interspeech*, 2010, pp. 386–389.

[42] C. Leggetter and P. Woodland, "Maximum likelihood linear regression for speaker adaptation of continuous density hidden Markov models," *Computer Speech and Language*, vol. 9, no. 2, pp. 171 – 185, 1995.

[43] M. Gales and P. Woodland, "Mean and variance adaptation within the MLLR framework," *Computer Speech & Language*, vol. 10, no. 4, pp. 249 – 264, 1996.

[44] J.-L. Gauvain and C.-H. Lee, "Maximum a posteriori estimation for multivariate gaussian mixture observations of markov chains," *IEEE Transactions on Speech and Audio Processing*, vol. 2, pp. 291–298, 1994.

[45] R. Kuhn, J.-C. Junqua, P. Nguyen, and N. Niedzielski, "Rapid speaker adaptation in eigenvoice space," *IEEE Transactions on Speech and Audio Processing*, vol. 8, no. 6, pp. 695 –707, 2000.

[46] A. Nadas, D. Nahamoo, and M. Picheny, "Speech recognition using noise-adaptive prototypes," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 37, no. 10, pp. 1495–1503, 1989.

[47] R. Rose, E. Hofstetter, and D. Reynolds, "Integrated models of signal and background with application to speaker identification in noise," *IEEE Transactions on Speech and Audio Processing*, vol. 2, no. 2, pp. 245–257, 1994.

[48] M. Wohlmayr, L. Mohr, and F. Pernkopf, "Self-adaption in single-channel source separation," in *International Conference on Spoken Language Processing (Interspeech)*, 2014, pp. –.

[49] Y. Wang and D. Wang, "Cocktail party processing via structured prediction," in *Neural Information Processing Systems (NIPS)*, 2012, pp. 224–232.

[50] M. Stark, M. Wohlmayr, and F. Pernkopf, "Source-filter based single channel speech separation using pitch information," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 2, pp. 242–255, 2011.

[51] M. Stark and F. Pernkopf, "On optimizing the computational complexity for vq-based single channel source separation," in *International Conference on Acoustics Speech and Signal Processing (ICASSP)*, 2010, pp. 237 –240.

[52] E. Vincent, J. Barker, S. Watanabe, J. Le Roux, F. Nesta, and M. Matassoni, "The second CHiME speech separation and recognition challenge: An overview of challenge systems and outcomes," in *Automatic Speech Recognition and Understanding (ASRU) Workshop*, 2013.

[53] "ITU-T recommendation P.862. Perceptual evaluation of speech quality (PESQ): an objective method for end-to-end speech quality assessment of narrow-band telephone networks and speech codecs," 2000.

[54] V. Emiya, E. Vincent, N. Harlander, and V. Hohmann, "Subjective and objective quality assessment of audio source separation," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 7, pp. 2046–2057, 2011.

[55] P. Jax and P. Vary, "On artificial bandwidth extension of telephone speech," *Signal Processing*, vol. 83, pp. 1707–1719, 2003.

[56] Y. Linde, A. Buzo, and R. Gray, "An algorithm for vector quantizer design," *IEEE Transaction on Communication*, vol. 28, no. 1, pp. 84–95, 1980.

[57] L. Rabiner, "A tutorial on Hidden Markov Models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–286, 1989.

[58] G.-B. Song and P. Martynovich, "A study of HMM-based bandwidth extension of speech signals," *Signal Processing*, vol. 89, pp. 2036–2044, 2009.

[59] D. Ackley, G. Hinton, and T. Sejnowski, "A learning algorithm for Boltzmann machines," *Cognitive Science*, vol. 9, no. 2, pp. 147–169, 1985.

[60] P. Smolensky, *Information processing in dynamical systems: Foundations of harmony theory*. MIT Press, 1986, vol. 1, pp. 194–281.

[61] T. Tieleman, "Training restricted Boltzmann machines using approximations to the likelihood gradient," in *International Conference on Machine learning (ICML)*, 2008, pp. 1064–1071.

[62] G. Hinton, "A Practical Guide to Training Restricted Boltzmann Machines," Department of Computer Science, University of Toronto, Tech. Rep., 2010.

[63] Y. Bengio, P. Lamblin, D. Popovici, and H. Larochelle, "Greedy layer-wise training of deep networks," in *Neural Information Processing Systems (NIPS)*, 2007.

[64] S. Rifai and X. Muller, "Contractive auto-encoders: Explicit invariance during feature extraction," in *International Conference on Machine learning (ICML)*, 2011, pp. 833–840.

[65] G. Alain, Y. Bengio, and S. Rifai, "Regularized auto-encoders estimate local statistics," *CoRR*, pp. 1–17, 2012.

[66] A. Darwiche, "A Differential Approach to Inference in Bayesian Networks," *ACM*, vol. 50, no. 3, pp. 280–305, 2003.

[67] R. Peharz, S. Tschiatschek, F. Pernkopf, and P. Domingos, "On theoretical properties of sum-product networks," in *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2015.

[68] V. Emiya, E. Vincent, N. Harlander, and V. Hohmann, "Subjective and objective quality assessment of audio source separation," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 7, pp. 2046–2057, 2011.

[69] M. Cooke, J. Barker, S. Cunningham, and X. Shao, "An audio-visual corpus for speech perception and automatic speech recognition." *The Journal of the Acoustical Society of America*, vol. 120, no. 5, pp. 2421–2424, 2006.

[70] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2010, pp. 249–256.

[71] P.-S. Huang, M. Kim, M. Hasegawa-Johnson, and P. Smaragdis, "Deep learning for monaural speech separation," in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2014, pp. 1562–1566.

[72] S. Tschiatschek and F. Pernkopf, "On Reduced Precision Bayesian Network Classifiers." *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 37, no. 4, pp. 774–785, 2015.

[73] S. Tschiatschek, K. Paul, and F. Pernkopf, "Integer Bayesian networks," in *European Conference on Machine Learning (ECML)*, 2014, pp. 209–224.

[74] S. Tschiatschek and F. Pernkopf, "Learning of bayesian network classifiers under computational constraints," in *European Conference on Machine Learning (ECML)*, 2015.

[75] "ETSI: Digital cellular telecommunications system (phase 2+); enhanced full rate (EFR) speech transcoding, ETSI EN 300 726 v8.0.1," Nov. 2000.

[76] D. Griffin and J. Lim, "Signal estimation from modified short-time Fourier transform," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 32, no. 2, pp. 236–243, 1984.

[77] P. C. Loizou, *Speech Enhancement: Theory and Practice*. Boca Raton, FL, USA: CRC Press, Inc., 2013.

[78] S. Möller, E. Kelaidi, F. Köster, N. Côté, P. Bauer, T. Fingscheidt, T. Schlien, H. Pulakka, and P. Alku, "Speech quality prediction for artificial bandwidth extension algorithms," in *Interspeech*, 2013, pp. 3439–3443.

Matthias Zöhrer received his MSc (Dipl. Ing.) degree in Telematik at Graz University of Technology, Austria, in summer 2013. Since 2013 he is a Research Associate at the Laboratory of Signal Processing and Speech Communication, Graz University of Technology, Austria. His research interests include machine learning, representation learning, deep learning architectures, GPU optimized processing, and image- and speech processing applications.

Robert Peharz received his PhD degree in 2015. His research interests include machine learning in general, sparse coding and nonnegative matrix factorization, classical probabilistic graphical models, sum-product networks, large margin methods, discriminative/generative hybrid methods, classification, semi-supervised learning, sequential data modeling and structured prediction, with applications to speech/audio engineering and computer vision. Currently he is with iDN, Institute of Physiology, at the Medical University of Graz.

Franz Pernkopf received his MSc (Dipl. Ing.) degree in Electrical Engineering at Graz University of Technology, Austria, in summer 1999. He earned a PhD degree from the University of Leoben, Austria, in 2002. In 2002 he was awarded the Erwin Schrödinger Fellowship. He was a Research Associate in the Department of Electrical Engineering at the University of Washington, Seattle, from 2004 to 2006. Currently, he is Associate Professor at the Laboratory of Signal Processing and Speech Communication, Graz University of Technology, Austria. His research interests include machine learning, discriminative learning, graphical models, deep learning architectures, feature selection, finite mixture models, and image- and speech processing applications.