HIERARCHICAL MUSICAL INSTRUMENT SEPARATION

Ethan Manilow
Northwestern University
ethanm@u.northwestern.edu

Gordon Wichern **Jonathan Le Roux**
Mitsubishi Electric Research Laboratories (MERL)
wichern@merl.com leroux@merl.com

ABSTRACT

Many sounds that humans encounter are hierarchical in nature; a piano note is one of many played during a performance, which is one of many instruments in a band, which might be playing in a bar with other noises occurring. Inspired by this, we re-frame the musical source separation problem as hierarchical, combining similar instruments together at certain levels and separating them at other levels. This allows us to deconstruct the same mixture in multiple ways, depending on the appropriate level of the hierarchy for a given application. In this paper, we present various methods for hierarchical musical instrument separation, with some methods focusing on separating specific instruments (like guitars) and other methods that determine what to separate based on a user-supplied audio example. We additionally show that separating all hierarchy levels is possible even when training data is limited at fine-grained levels of the hierarchy.

1. INTRODUCTION

The field of source separation has seen notable performance improvements with the introduction of deep learning techniques, most notably in the areas of speech enhancement [1–4], speech separation [5–8], and music separation [9–12]. These techniques succeed in cases where the notion of a source is well defined; in the case of speech enhancement or separation, the target is always defined as the speech of a single speaker. However, real-world scenarios can have more complicated definitions of a source. Consider the case where a band is playing on the radio while two people are having a conversation: how does one segment this audio scene? Is the radio one source and the talkers each a source? Or are each of the instruments in the band on the radio a source as well? Clearly, there are many correct answers to this question, but one way to understand this auditory scene is to apply a hierarchical structure to its parts. In this work, we re-frame the source separation problem as hierarchical, focusing on the example of music source separation, where we use musical instruments as elements in a complex hierarchical auditory scene.

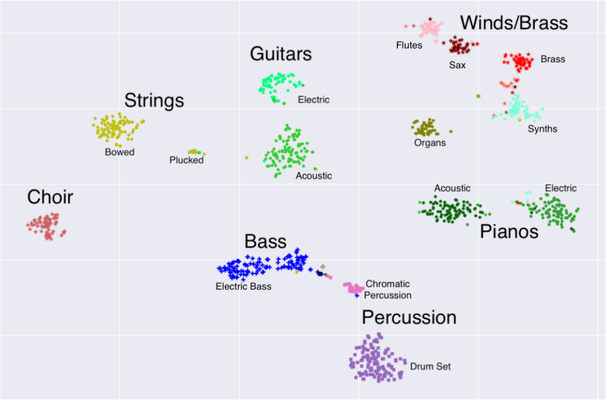

Figure 1. Annotated t-SNE [13] projection of the learned anchors from a hierarchical query-by-example separation model on a test set.

When considering music separation, determining what constitutes a target source is not well defined. Even in a well-studied problem like singing voice separation [9–11], in which the singer is isolated from non-vocal background music, the definition of what is "singing voice" is somewhat muddled. Many popular songs often contain a lead vocal part, possibly several additional background vocal parts, and sometimes additional vocal effect tracks. This is a simple case; when we consider instrument categories with a larger variety of possible timbres, like synthesizers or guitars, deciding what particular instrument part to isolate can become even harder to nail down. One may want to go even further and separate each instrument into unique notes or chord instances.

Framing a musical scene as hierarchical has precedent in fields that study human audition. Evidence shows that human auditory perception has many hierarchical characteristics [14–17]. As Bregman notes in *Auditory Scene Analysis* [18]: "It makes sense [...] to think of the auditory perceptual organization of [a musical] duet as having a hierarchical structure [...]. This argument implies that there are levels of perceptual belongingness intermediate between 'the same thing' and 'unrelated things'". While perceptual auditory hierarchies can involve timing, timbre, rhythm, and much more, in this paper we focus on the task of building hierarchical source separation systems via an instrument hierarchy.

In the field of musicology, musical instruments have long been thought of as hierarchical. Almost all human cultures throughout history have created musical instrument classification systems [19], many of which are inherently hierarchical. One prominent example is the

Attribution: E. Manilow, G. Wichern, J. Le Roux, "Hierarchical Musical Instrument Separation", in *Proc. of the 21st Int. Society for Music Information Retrieval Conf.,* Montréal, Canada, 2020.

Hornbostel-Sachs system [20], which classifies musical instruments by their sound production mechanisms in a hierarchical manner similar to the Dewey Decimal System [21]. Another system widely used in Western music classifies instruments by their musical range, with terms named after singing voice classifications: *soprano*, *alto*, etc. We use a hierarchy inspired by both of these approaches for hierarchical source separation.

There is also an element of a musical instrument hierarchy when making recordings in the recording studio. Each track is assumed to be an isolated recording of a single instrument, or part of one instrument. At the mixing board, a sound engineer can mix together multiple tracks into a "submix", which acts as a single unit in the recording session, having effects and other signal routing configurations be specific to the submix rather than the individual tracks therein [22]. The submixes are then manipulated alongside other tracks, which may contain only a single instrument. For example, a standard practice is to record every separate piece of a drum kit with a single microphone and then combine those into a drum submix. This configuration is hierarchical; the engineer can choose to manipulate all of the drum sounds (the drum submix) or manipulate individual drum tracks within it.

In this paper, we re-frame the problem of musical source separation as hierarchical. We propose two main strategies for hierarchical source separation, one solely based on the well-studied source-specific mask inference approach to source separation [3], and another based on more recently proposed query-by-example source separation systems [23,24]. In both cases, we learn to simultaneously separate submixes of instruments corresponding to multiple levels of an instrument label hierarchy. By learning to separate sources at multiple levels of granularity, we observe performance benefits even in cases where training data is limited for the most fine-grained source types.

2. RELATED WORK

Music source separation has recently seen a great deal of success. Most of this success is owed to the availability of the MUSDB18 [25] dataset. This dataset has avoided the "source definition ambiguity" by grouping all audio within a track into four target sources: vocals, bass, drums, and other. The "other" source contains a variety of different instruments, like guitars, pianos, strings, and synthesizers. While MUSDB18 has undoubtedly helped to advance the field of music source separation, its source groupings remain overly coarse for many real-world remixing applications. In this work, we propose systems to separate sources historically grouped as the "other" source.

Our proposed work is related to source separation algorithms that attempt to estimate multiple musical sources with one network. Some works accomplish this by outputting a set of masks for each target source, improving performance via specialized training techniques [26,27] or by giving the networks additional tasks to solve, like music transcription [28]. Other works accomplish this by conditioning a network to output different sources depending on the desired source [23, 29, 30]. None of these approaches have any requirements that the sources they separate have any inherent structure in relation to other sources, especially not in a hierarchical manner as we propose here.

This work also draws inspiration from query-by-example (QBE) networks. Within the speech separation literature, the task of using a query to separate a specific speaker from a mixture with many speakers is called speaker extraction, and this task has garnered much attention recently [31–33]. Specifically, this work builds off of work [34] that extends deep attractor networks [35] for the QBE case. Deep attractor networks have been successfully used for music separation [23,36], where QBE music separation was considered as an auxiliary benefit of the learned embedding space. Although systems specifically tailored to QBE separation of musical instruments have also been proposed [24], none of these systems assume or enforce any hierarchical structure on an auditory scene.

3. AUDITORY HIERARCHIES

In this work, we are interested in hierarchies of sound producing objects, where top levels of the hierarchy correspond to broad groups (e.g., midrange stringed instruments) and lower levels are more specific (e.g., acoustic guitar). With regard to source separation, we can define an auditory hierarchy such that sources at higher levels in the hierarchy are composed of mixtures of sources at lower levels of the hierarchy. Each source node can potentially be further separated into child sources and combined with its siblings to create parent sources. Considering a hierarchy with L levels, we denote by $\mathcal{S}_{l,c}$ the c-th source type node at hierarchy level l, for $l = 1, \ldots, L$, where we assume that the set of leaf source types $\mathcal{S}_{1,c}$ cannot be decomposed into further source types, and $\mathcal{S}_{L,1}$ is the sole source type at the top of the hierarchy and includes all source types. Further denoting by $\mathcal{C}_{l,c}$ the set of indices of the child sources at level $l-1$ of $\mathcal{S}_{l,c}$, the hierarchy can be defined as

$$\mathcal{S}_{l,c} = \bigcup_{c' \in \mathcal{C}_{l,c}} \mathcal{S}_{l-1,c'}, \forall l = 2, \ldots, L. \qquad (1)$$

We define a *path* down the hierarchy as a sequence of source types from a beginning source type node $\mathcal{S}_a$ to a destination source type node $\mathcal{S}_b$ at a lower level.

When using this hierarchy to decompose a mixture x, we denote by $\boldsymbol{S}_{l,c}$ the corresponding source component in x whose source type is $\mathcal{S}_{l,c}$, where the submix of all signals of the same type are considered as a single component. By definition, $\boldsymbol{S}_{L,1} = x$. Each c-th source component $\boldsymbol{S}_{l,c}$ at a level l can be decomposed into source components $\boldsymbol{S}_{l-1,c'}$, such that $\boldsymbol{S}_{l-1,c'}$ is the signal corresponding to all sources belonging to the child source type $\mathcal{S}_{l-1,c'}$:

$$\boldsymbol{S}_{l,c} = \sum_{c' \in \mathcal{C}_{l,c}} \boldsymbol{S}_{l-1,c'}, \text{ s.t. } \boldsymbol{S}_{l-1,c'} \in \mathcal{S}_{l-1,c'}, \qquad (2)$$

for $l = 2, \ldots, L$. For simplicity, we use the sum operator to denote mixing, although the mixing process is often more complex than a simple summation of signals.

In this paper, we specifically examine auditory hierarchies composed of mixtures of musical instruments, but we note that this hierarchical formulation can be applied to mixtures with any type of source content.

4. HIERARCHICAL SOURCE SEPARATION

In general, source separation is formulated as trying to estimate C complex spectrograms, $\boldsymbol{S}_c \in \mathbb{C}^{F\times T}$ for $c = 1, \ldots, C$, that represent a set of desired sources within the spectrogram $X \in \mathbb{C}^{F\times T}$ of an audio mixture. In this general formulation, there is no requirement that source $\boldsymbol{S}_c$ have any relationship to source $\boldsymbol{S}_d$, for $c \neq d$.

Given an audio mixture X, a hierarchical separation algorithm under a given hierarchy may attempt to extract a submix of all sources belonging to some source type $\mathcal{S}_{l,c}$ at a level l. For instance, separating out all guitars (acoustic and electric) from a mixture that includes electric guitar, acoustic guitar, piano, and drums (as depicted in Fig. 2).

4.1 Hierarchical Source-Specific Separation

Conventional source-specific separation (SSS) networks based on mask inference typically attempt to estimate a real-valued mask $\hat{M}_c \in \mathbb{R}^{F\times T}$ for a single target source c by minimizing some distortion measure between the source estimate obtained from the mask and a reference $\boldsymbol{S}_c$. A commonly used example of such an objective function, which we use in this work, is the truncated phase sensitive approximation (tPSA) objective [3]:

$$\mathcal{L}_{\text{tPSA}} = \left\| \hat{M}_c \odot |X| - \mathrm{T}_0^{|X|}\left(|\boldsymbol{S}_c| \odot \cos(\angle \boldsymbol{S}_c - \angle X)\right) \right\|_1, \tag{3}$$

where $\odot$ denotes element-wise product, $|Y|$ and $\angle Y$ denote the magnitude and phase of a spectrogram Y, and $\mathrm{T}_0^{|X|}(x) = \min(\max(x, 0), |X|)$ is a truncation function ensuring the target can be reached with a sigmoid activation function. The estimated mask $\hat{M}_c$ is element-wise multiplied with the original mixture spectrogram X to obtain an estimate for the target source $\boldsymbol{S}_c$.

As a first naive strategy for building hierarchical SSS networks, we can train single networks which output a single node $\mathcal{S}_{n,c}$ at a given level of the hierarchy. Each such single-level network can be trained to minimize the tPSA objective above, where the target source is $\boldsymbol{S}_{n,c}$, the component corresponding to the targeted source type in the hierarchy within the mixture X. Each of these networks outputs one mask $\hat{M}_{n,c}$ for its targeted source type, and they are trained independently of each other.

In order to make further use of the hierarchical structure of the data, we propose a multi-level strategy in which we train a network to output multiple levels of the hierarchy at once. A potential advantage of this strategy is that the network may be able to leverage learned knowledge about a mask $\hat{M}_{n,c}$ to reinforce and improve its estimate for another mask $\hat{M}_{n',c'}$ in the hierarchy. A trivial implementation of this strategy would be to output a mask for each leaf node in the hierarchy, and recompose the leaf sources as we travel through the hierarchy, training the network by combining loss functions for all nodes in the hierarchy. However, any sufficiently realistic hierarchy likely contains dozens of leaf nodes, leading to memory and computation issues as well as difficulties balancing the contributions of all the losses. To avoid these issues, we consider instead a single network that outputs N masks for N levels along a single path down the hierarchy, e.g., [strings/keys] → [guitars] → [clean guitars] ("Clean" indicates acoustic and electric guitars with no overdrive or distortion applied).

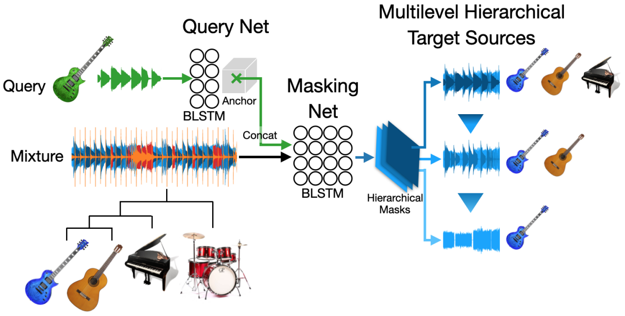

Figure 2. One of the proposed methods for hierarchical source separation. We assume that a mixture contains a hierarchy of musical instruments (bottom left), and use an audio query (the green electric guitar, top left) to separate instruments at multiple levels of the hierarchy, with the closest target at the lowest level (blue electric guitar).

4.2 Hierarchical Query-by-Example

The approaches described above cannot capture many instruments in an instrument hierarchy: using one network per level only allows the network to learn one node in the hierarchy at a time, and using a multilevel network only learns one path down the instrument hierarchy. If we want to capture relationships between different instruments in a hierarchy, we need a method for separating multiple instruments at different levels with a single network.

A successful recent strategy involves query-by-example (QBE) networks that ingest a mixture and an example of the desired source to separate from the mixture [24]. By extending this to a hierarchical case, we can model an entire instrument hierarchy for source separation. Note that, instead of conditioning on a query, we could alternatively condition the separation on the leaf node label, leading to a hierarchical extension of conditional source separation methods [23, 29, 30]. We here focus on QBE, as an audio query can be considered as a generalization of a class label, and QBE may further provide the ability to interpolate to unseen source types during inference.

Our proposed realization of hierarchical QBE relies on two networks, a query net and a masking net. The query net calculates a query anchor $A_q \in \mathbb{R}^k$ for some input query $Q \in \mathbb{R}^{F\times T_q}$ as a weighted sum of k-dimensional query embeddings $V_{q,i}$ produced by the network at each time-frequency bin $i = (f, t)$ of the query spectrogram space:

$$A_q = \frac{\sum_i P_{q,i} V_{q,i}}{\sum_i P_{q,i}}, \tag{4}$$

where $P_q \in \mathbb{R}^{FT_q}$ is a query presence vector for query Q, defined such that $P_{q,i} = 1$ if the magnitude at bin $i = (f, t)$ is above a threshold (set to -60 dB from the maximum in our experiments), and 0 otherwise. The query anchor A_q is concatenated with the frequency vector of the mixture X_t at each frame t, and used as input to the masking network, which produces, for each hierarchy layer n of interest, a mask $\hat{M}_{n,c}$ for a target source $S_{n,c}$ which is in the same node $\mathcal{S}_{n,c}$ of the hierarchy as the query Q. This architecture is depicted in Fig. 2.

This QBE system is trained to minimize the tPSA objective in Eq. 3 based on a target source $S_{n,c}$, where the target source used to train the network is here determined both by the query and a given level in the hierarchy. Other QBE systems [24] apply a loss directly on the query embedding space; while we leave this direction to future work, we note that we are already able to learn some form of hierarchical structure without introducing a specific loss on the embedding space, as exemplified in Fig. 1.

Using an acoustic guitar query as example, the training procedure for a hierarchical QBE system is as follows: an acoustic guitar query is used to train a network that attempts to extract the corresponding sources at the leaf node level, in which case the target will consist of the submix of all clean guitars in the mixture. Note that we leave the problem of separating instruments of the same fine-grained type (e.g., multiple clean guitars) using techniques such as permutation-invariant training [5, 6] for future work. The same acoustic guitar query may also be used to train a network that attempts to extract the corresponding sources one level up, in which case the target will consist of the submix of all guitars in the mixture, regardless of whether they are clean guitars or not. When there is no target in the mixture corresponding to the query at the given level of the hierarchy, the target is set to silence.

As with hierarchical SSS networks, we can make a single-level QBE network for each separate level in the hierarchy and only separate instruments at that level, as described in the above example, or we can make a single hierarchical multi-level QBE network that returns multiple (or even all) levels of the hierarchy. For the latter strategy, we can consider enforcing a hierarchical constraint on the masks, as described below.

4.3 Constraints on Hierarchical Masks

Assuming the components of a mixture exist in some hierarchy, we can leverage knowledge about its structure to impart constraints on the network. For instance, we can use the relationship defined in Eq. 2 to require the set of masks produced by a multi-level hierarchical network to follow the same structure as the hierarchy, namely that masks at higher levels be composed of masks at lower levels.

However, this would require us to output masks for every node in the hierarchy, which is infeasible for any sufficiently realistic hierarchy. Instead, we consider imposing a hierarchical constraint that does not depend on knowledge of the whole hierarchy. This hierarchical constraint requires that masks at higher levels in the hierarchy must apportion at least the same amount of energy as masks at lower levels. More precisely, the mask at level l is set as

$$\hat{M}_l = \max(\hat{M}'_l, \hat{M}_{l-1}), \tag{5}$$

where max is applied element-wise to every TF bin, and $\hat{M}'_l$ is the mask estimate output by the network for level l.

Level	Submixes to be separated
3	Keyboards, guitars, and orchestral strings
2	All guitars (both clean and effected)
1	Only clean guitars (both electric and acoustic)

Table 1. Contents of hierarchical levels used for training and testing the hierarchical single-instrument source-specific separation (SSS) networks[1]. Hierarchical SSS can only learn one path down the hierarchy at a time.

5. EXPERIMENTAL DESIGN

We design a set of experiments to determine the validity of our hierarchical source separation methods outlined above. We want to understand how well the proposed methods work in a hierarchical scenario. We look specifically at the case of a musical instrument hierarchy.

5.1 Dataset and Evaluation

To test the proposed methods in this paper, we required a large dataset with isolated sources of many instruments that could be combined in a hierarchical way. Specifically, we required a dataset with a wide variety of granular source labels, i.e., not only “guitars”, but “acoustic guitars”, “electric guitars”, “effected guitars”, and so on for every instrument in the dataset. Because of this, we chose Slakh2100 [37], which contains 2,100 musical mixtures along with isolated sources. This dataset has 145 hours of mixture data split into 34 instrument categories.

Before selecting excerpts from the dataset, we created a musical instrument hierarchy from Slakh’s included instrument categories [1]. For these experiments, we define a hierarchy with three levels (excluding the trivial level consisting of the mixtures of all sources). The top level contains four categories: mid-range strings and keys (guitars, keyboards, and orchestral strings), bass instruments (acoustic and electric basses), winds (flutes, reeds, and brass), and percussion (drum sets and chromatic percussion). The middle level has seven categories (e.g., from mid-range strings: orchestral strings, guitars, keyboards, and electric keyboards), and the lowest level has eighteen categories (e.g., from guitars: clean guitars, and effected guitars). We note that this is just one of many possible hierarchies and almost all of the instruments described here would be classified as “other” in MUSDB18 [25].

To select examples from the dataset, we create multiple instantaneous submixes for each track, corresponding to the different levels of the hierarchy. As an example illustrated in Table 1, at the highest level, all pianos, guitars, and strings are considered one source, while at the next

[1] The full hierarchy can be seen at: `https://git.io/JJ4gx`

level all guitars are considered one source, and at the lowest level only clean guitars are considered one source. For each mixture in the dataset, we compute the saliency of each hierarchical submix in 10 second chunks, with a hop size of 2.5 seconds. If the source in the submix has energy above -30 dB in that chunk, it is considered salient. For the experiments involving multiple levels, we ensure that for a given node, its parent (or grandparent) has energy from child nodes other than itself. In other words, we want to make sure that a parent is not exactly the same as the child, meaning that some of the child node's siblings or cousins are also salient.

For our experiments, we use the *Slakh2100-split2* stratification and downsample the audio to 16 kHz. We do the mixing on the fly and select chunks randomly from the pool of salient examples for the specific experiment. For training, the networks see 20,000 examples per epoch ($\approx$55.5 h), and we use 3,000 examples ($\approx$8.3 h) for the validation and test sets. To ensure we have enough examples and a rich enough hierarchy to train, for the hierarchical SSS experiments we choose to separate sources down a path of the hierarchy as shown in Table 1, although the proposed methods can be extended to other paths down this or other hierarchies. For the QBE networks, we separate every instrument type in the hierarchy. Query chunks are selected from the pool of salient chunks such that they are always leaf nodes along the same path as the target regardless of the target level, but originate from different tracks.

For all experiments, we use the scale-invariant source-to-distortion ratio (SI-SDR) [38] to determine the output quality of our models. For reference, we also report the SI-SDR when doing no processing on the mixes.

5.2 Experiments and Model Configurations

In this paper, we evaluate four types of hierarchical source separation models. We vary models along two dimensions: whether they are single-instrument (i.e., source-specific separation, or SSS) or multi-instrument (i.e., query-by-example, or QBE), and whether they output a single level, or multiple levels. We describe each configuration below:

- **Single-instrument, Single-level:** A trio of instrument-specific SSS models each corresponding to one level of the hierarchy along one hierarchical path.
- **Single-instrument, Multi-level:** One SSS model that outputs a hierarchical set of masks, separating at all levels of a single hierarchical path simultaneously.
- **Multi-instrument, Single-level:** A trio of multi-instrument QBE models outputting one mask at one level of the hierarchy as determined by an input query.
- **Multi-instrument, Multi-level:** One QBE model that outputs a hierarchical set of masks for every level of the hierarchy along a path determined by an input query.

For the single-instrument models, we separate along one path of the hierarchy as referenced in Table 1. The multi-instrument, multi-level model is trained to separate a source based on a query, and thus can learn the full hierarchy (i.e., all instruments) instead of just one path as in the single-instrument, multi-level case.

Model Type	HC	Level 3	Level 2	Level 1
SSS (Guitar)		3.5	4.0	4.0
SSS (Guitar)	✓	3.2	3.6	3.8
QBE		3.2	2.4	0.2
QBE	✓	3.3	2.1	1.6

Table 2. Improvement in SI-SDR (dB) for hierarchical SSS (Guitar) and QBE models. Each model is trained either with the hierarchical constraint (HC) described in Section 4.3 or with no constraints on the masks produced for sources at different levels of granularity.

For the multi-level models, we test the effect of the hierarchical constraint proposed in Section 4.3. We can also test how well they learn with limited data about the leaf source. To do this, we train the three-level SSS and QBE models under the assumption that the leaf ground truth is unavailable either 50% or 90% of the time, in which cases only the upper levels are directly involved in the objective function. For comparison, we also evaluate models where *all* nodes are missing either 50% or 90% of the time during training. These experiments can tell us how well the multi-level network can leverage higher (i.e., coarser) levels of the hierarchy at the leaf node. Such an ability would be particularly advantageous as it is typically more difficult to collect data with fine-grained ground truth sources compared to data with a mixture and only a few source components gathered in broad categories, and could potentially help breaking open the "other" category of MUSDB18 with limited annotations.

All single-level and multi-level networks we test have the same architecture. The SSS models are composed of 4 bidirectional long short-term memory (BLSTM) layers with 600 hidden units in each direction and dropout of 0.3, followed by a fully connected layer with sigmoid activation function that outputs a mask. As described in Section 4.2, the QBE models are composed of two sub-networks, a query net and a masking net. The query net is composed of 2 BLSTM layers with 600 nodes in each direction and dropout of 0.3, followed by a fully-connected layer with linear activation that maps each time-frequency bin to an embedding space with 20 dimensions. The masking net is the same as the SSS models, with a larger input feature vector to accommodate the concatenated query anchor.

All models were trained with the Adam optimizer at a learning rate of 1e-4 for 100 epochs and a batch size of 25. The learning rate was halved if the loss on the validation set did not decrease for 5 straight epochs. The gradient was clipped to the 10th percentile of historical gradient norms if the norm of the minibatch was above that value [39].

6. RESULTS

In Table 2, we examine the effect of the hierarchical constraint (HC) on multi-level hierarchical networks. We observe that, for the source-specific separation network (which in this case only separates guitars), the HC slightly diminishes performance at all levels, indicating that SSS models are able to learn the specific hierarchical relation-

		All Levels			Level 3			Level 2			Level 1		
Model Type	# lvls	Mix	SI-SDR	Δ	Mix	SI-SDR	Δ	Mix	SI-SDR	Δ	Mix	SI-SDR	Δ
SSS (Guitar)	1	−3.9	−2.1	1.8	0.9	4.1	3.2	−5.9	−3.2	2.7	−6.6	−7.3	−0.7
SSS (Guitar)	3	−3.9	0.0	3.9	0.9	4.3	3.4	−5.9	−1.9	4.0	−6.6	−2.6	4.0
QBE	1	−4.9	−3.9	1.0	−1.3	2.0	3.3	−5.3	−3.9	1.4	−8.0	−9.8	−1.9
QBE	3	−4.9	−2.5	2.3	−1.3	2.0	3.3	−5.3	−3.2	2.1	−8.0	−6.4	1.6

Table 3. SSS and QBE model results in terms of SI-SDR (dB), where Δ denotes improvement over the noisy mix. SSS networks are only trained to separate sources in the hierarchy containing clean guitars (See Table 1), whereas QBE networks separate any source in the hierarchy. Here we compare single-level networks (denoted by a "1") to multi-level networks (denoted "3"). There is only one multi-level network for all three levels, but three single-level networks (one for each level).

	Data Reduction		Levels			
	%	type	All	Level 3	Level 2	Level 1
SSS (Guitar)	0	-	3.8	3.5	4.0	4.0
	50	all	3.3	3.1	3.4	3.4
	50	leaf	3.5	3.3	3.6	3.6
	90	all	0.1	1.5	−0.7	−0.5
	90	leaf	3.6	3.4	3.7	3.7
Mix			−3.9	0.9	−5.9	−6.6
QBE	0	-	2.3	3.3	2.1	1.6
	50	all	−1.5	−2.1	−1.4	−1.1
	50	leaf	2.2	3.4	2.1	1.1
	90	all	−1.8	−2.1	−1.8	−1.5
	90	leaf	1.9	3.1	1.7	0.8
Mix			−4.9	−1.3	−5.3	−8.0

Table 4. SI-SDR improvement (dB) over the unprocessed mix ("Mix") for hierarchical SSS and QBE models (separated by the thick broken line). Each model is trained while removing either just the leaf ("leaf") or the whole example ("all") for a specified percentage of the data. Reducing just leaf nodes up to 90% shows only a 0.3 dB drop for SSS and 0.8 dB drop for QBE compared to using all of the leaves.

ship for a single source (in this case, guitar) at different levels without additional help. For the query-by-example network (which separates all types of instruments), the HC marginally hinders performance at Level 2, but helps considerably for the leaf node (Level 1). We hypothesize that QBE networks benefit more because they are unable to learn the specific mask "shapes" of any individual source, and thus need the additional help offered by the HC. Therefore, in all subsequent experiments we include the HC for QBE networks, but omit it for the SSS networks.

In Table 3, we expand on the results from Table 2 and compare the results from single-level and multi-level hierarchical models for both SSS and QBE separation models. In both cases, the multi-level hierarchical networks improve over the single-level models, with the largest gains occurring at lower hierarchy levels. This implies that the networks can leverage their shared knowledge of the hierarchy to aid themselves at the lower levels, where individual instruments are more difficult to discern in the mix.

From the Level 1 results in Table 3, we see that separating sources at this fine level of detail (e.g., clean electric guitars vs. distorted electric guitars) is extremely difficult, especially with a MIDI-synthesized data set such as Slakh2100, where several different instrument types may sound similar. In fact, when trying to train a single network to only separate these fine-grained sources, we are unsuccessful as noted by the negative SI-SDR improvements in the # lvls=1 (single level) rows for Level 1 sources. Training networks on multiple levels simultaneously mitigates this to some extent, although we have informally noticed the multi-level network sometimes outputting nearly identical separated sources between Level 1 and Level 2. We also note that the highest output SI-SDR values are obtained when separating Level 3 sources in Table 3, and we mention that Level 3 sources can be considered similar to the "other" source class in MUSDB18 [25]. Therefore, separating sources at the more fine-grained Levels (1 and 2) is more difficult than what is typically attempted in musical source separation.

In Table 4, we can observe the effect of removing leaf sources (Level 1 sources, see Table 1 for guitar example) from the training set. Compared to reducing *all* of the data by 50% or 90%, the performance of reducing only the leaves degrades very minimally. In cases where we have rich data at higher levels but sparse data at lower levels, hierarchical multi-level networks can do a respectable job at separating lower levels. We see the same story for both SSS and QBE networks: even a small amount of leaf data can help ward off a large drop in performance.

7. CONCLUSIONS

In this paper, we re-framed the source separation problem as hierarchical, and demonstrated the benefit of learning to simultaneously separate sources at different levels of granularity. In the present work, we considered network architectures that output masks for source separation at all relevant levels together. We showed that in doing so, we are still able to separate out the most granular source types when training data is severely limited. A major drawback of this work is the need for a large quantity of labeled and curated data, a limitation that we hope future work can address. Other future directions include architectures that output relevant levels sequentially, such as cascaded models [40], or directions inspired by hierarchical audio classification models [41,42].

8. REFERENCES

[1] Y. Xu, J. Du, L.-R. Dai, and C.-H. Lee, "An experimental study on speech enhancement based on deep neural networks," *IEEE Signal Processing Letters*, vol. 21, no. 1, pp. 65–68, 2014.

[2] F. J. Weninger, J. R. Hershey, J. Le Roux, and B. Schuller, "Discriminatively trained recurrent neural networks for single-channel speech separation," in *GlobalSIP Machine Learning Applications in Speech Processing Symposium*, Dec. 2014.

[3] H. Erdogan, J. R. Hershey, S. Watanabe, and J. Le Roux, "Phase-sensitive and recognition-boosted speech separation using deep recurrent neural networks," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Apr. 2015, pp. 708–712.

[4] D. Wang and J. Chen, "Supervised speech separation based on deep learning: An overview," *IEEE/ACM Transactions on Audio, Speech and Language Processing*, vol. 26, no. 10, pp. 1702–1726, 2018.

[5] J. R. Hershey, Z. Chen, and J. Le Roux, "Deep clustering: Discriminative embeddings for segmentation and separation," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Mar. 2016, pp. 31–35.

[6] M. Kolbæk, D. Yu, Z.-H. Tan, and J. Jensen, "Multitalker speech separation with utterance-level permutation invariant training of deep recurrent neural networks," *IEEE/ACM Transactions on Audio, Speech and Language Processing*, pp. 1901–1913, 2017.

[7] Z.-Q. Wang, J. Le Roux, and J. R. Hershey, "Alternative objective functions for deep clustering," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Apr. 2018.

[8] Y. Luo and N. Mesgarani, "TasNet: Surpassing ideal time-frequency masking for speech separation," *arXiv preprint arXiv:1809.07454*, Sep. 2018.

[9] A. Jansson, E. Humphrey, N. Montecchio, R. Bittner, A. Kumar, and T. Weyde, "Singing voice separation with deep U-Net convolutional networks," in *Proc. International Society for Music Information Retrieval Conference (ISMIR)*, Oct. 2017.

[10] N. Takahashi, N. Goswami, and Y. Mitsufuji, "MM-DenseLSTM: An efficient combination of convolutional and recurrent neural networks for audio source separation," in *Proc. IEEE International Workshop on Acoustic Signal Enhancement (IWAENC)*, Sep. 2018.

[11] Y. Luo, Z. Chen, J. R. Hershey, J. Le Roux, and N. Mesgarani, "Deep clustering and conventional networks for music separation: Stronger together," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Mar. 2017, pp. 61–65.

[12] F.-R. Stöter, S. Uhlich, A. Liutkus, and Y. Mitsufuji, "Open-Unmix - a reference implementation for music source separation," *Journal of Open Source Software*, 2019. [Online]. Available: https://doi.org/10.21105/joss.01667

[13] L. v. d. Maaten and G. Hinton, "Visualizing data using t-SNE," *Journal of machine learning research*, vol. 9, no. Nov., pp. 2579–2605, 2008.

[14] C. M. Wessinger, J. W. VanMeter, B. Tian, J. Van Lare, J. Pekár, and J. P. Rauschecker, "Hierarchical organization of the human auditory cortex revealed by functional magnetic resonance imaging," *Journal of cognitive neuroscience*, vol. 13, no. 1, pp. 1–7, 2001.

[15] J. E. Peelle, I. Johnsrude, and M. H. Davis, "Hierarchical processing for speech in human auditory cortex and beyond," *Frontiers in human neuroscience*, vol. 4, p. 51, 2010.

[16] G. G. Parras, J. Nieto-Diego, G. V. Carbajal, C. Valdés-Baizabal, C. Escera, and M. S. Malmierca, "Neurons along the auditory pathway exhibit a hierarchical organization of prediction error," *Nature communications*, vol. 8, no. 1, pp. 1–17, 2017.

[17] M. M. Farbood, D. J. Heeger, G. Marcus, U. Hasson, and Y. Lerner, "The neural processing of hierarchical structure in music and speech at different timescales," *Frontiers in neuroscience*, vol. 9, p. 157, 2015.

[18] A. S. Bregman, *Auditory scene analysis: The perceptual organization of sound.* MIT press, 1994.

[19] M. J. Kartomi, *On concepts and classifications of musical instruments.* University of Chicago Press Chicago, 1990.

[20] E. M. Von Hornbostel and C. Sachs, "Classification of musical instruments: Translated from the original German by Anthony Baines and Klaus P. Wachsmann," *The Galpin Society Journal*, pp. 3–29, 1961.

[21] M. Dewey, *A classification and subject index, for cataloguing and arranging the books and pamphlets of a library.* Brick row book shop, Incorporated, 1876.

[22] B. Owsinski, *The mixing engineer's handbook.* Nelson Education, 2013.

[23] P. Seetharaman, G. Wichern, S. Venkataramani, and J. Le Roux, "Class-conditional embeddings for music source separation," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2019, pp. 301–305.

[24] J. H. Lee, H.-S. Choi, and K. Lee, "Audio query-based music source separation," *arXiv preprint arXiv:1908.06593*, 2019.

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

[26] C. S. Doire and O. Okubadejo, “Interleaved multitask learning for audio source separation with independent databases,” *arXiv preprint arXiv:1908.05182*, 2019.

[27] V. S. Kadandale, J. F. Montesinos, G. Haro, and E. Gómez, “Multi-task U-Net for music source separation,” *arXiv preprint arXiv:2003.10414*, 2020.

[28] E. Manilow, P. Seetharaman, and B. Pardo, “Simultaneous separation and transcription of mixtures with multiple polyphonic and percussive instruments,” in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2020, pp. 771–775.

[29] O. Slizovskaia, G. Haro, and E. Gómez, “Conditioned source separation for music instrument performances,” *arXiv preprint arXiv:2004.03873*, 2020.

[30] G. Meseguer-Brocal and G. Peeters, “Conditioned-U-Net: Introducing a control mechanism in the U-Net for multiple source separations,” *arXiv preprint arXiv:1907.01277*, 2019.

[31] X. Xiao, Z. Chen, T. Yoshioka, H. Erdogan, C. Liu, D. Dimitriadis, J. Droppo, and Y. Gong, “Single-channel speech extraction using speaker inventory and attention network,” in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2019, pp. 86–90.

[32] Q. Wang, H. Muckenhirn, K. Wilson, P. Sridhar, Z. Wu, J. Hershey, R. A. Saurous, R. J. Weiss, Y. Jia, and I. L. Moreno, “Voicefilter: Targeted voice separation by speaker-conditioned spectrogram masking,” *arXiv preprint arXiv:1810.04826*, 2018.

[33] M. Delcroix, K. Zmolikova, K. Kinoshita, A. Ogawa, and T. Nakatani, “Single channel target speaker extraction and recognition with speaker beam,” in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Apr. 2018, pp. 5554–5558.

[34] J. Wang, J. Chen, D. Su, L. Chen, M. Yu, Y. Qian, and D. Yu, “Deep extractor network for target speaker recovery from single channel speech mixtures,” *arXiv preprint arXiv:1807.08974*, 2018.

[35] Z. Chen, Y. Luo, and N. Mesgarani, “Deep attractor network for single-microphone speaker separation,” in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Mar. 2017, pp. 246–250.

[36] R. Kumar, Y. Luo, and N. Mesgarani, “Music source activity detection and separation using deep attractor network.” in *Proc. ISCA Interspeech*, Sep. 2018, pp. 347–351.

[37] E. Manilow, G. Wichern, P. Seetharaman, and J. Le Roux, “Cutting music source separation some Slakh: A dataset to study the impact of training data quality and quantity,” in *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, Oct. 2019.

[38] J. Le Roux, S. T. Wisdom, H. Erdogan, and J. R. Hershey, “SDR – half-baked or well done?” in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2019.

[39] P. Seetharaman, G. Wichern, B. Pardo, and J. Le Roux, “AutoClip: Adaptive gradient clipping for source separation networks,” in *Proc. International Workshop on Machine Learning for Signal Processing (MLSP)*, Oct. 2020.

[40] M. Maciejewski, G. Wichern, E. McQuinn, and J. Le Roux, “WHAMR!: Noisy and reverberant single-channel speech separation,” in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2020, pp. 696–700.

[41] J. Cramer, V. Lostanlen, A. Farnsworth, J. Salamon, and J. P. Bello, “Chirping up the right tree: Incorporating biological taxonomies into deep bioacoustic classifiers,” in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2020, pp. 901–905.

[42] H. Shrivaslava, Y. Yin, R. R. Shah, and R. Zimmermann, “Mt-gcn for multi-label audio tagging with noisy labels,” in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2020, pp. 136–140.