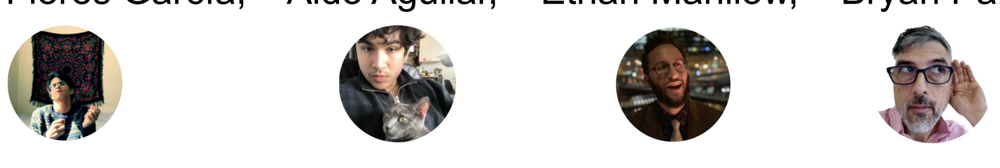
Leveraging Hierarchical Structures for Few-Shot Musical Instrument Recognition

Hugo Flores García, Aldo Aguilar, Ethan Manilow, Bryan Pardo



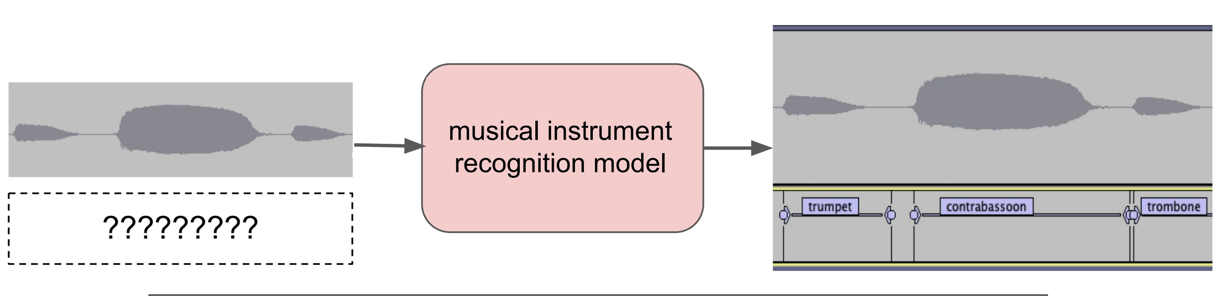
Interactive Audio Lab, Northwestern University

Takeaways

- Musical instrument recognition using few-shot learning
- A simple extension to **prototypical networks** that incorporates a class **hierarchy**
- Significantly better classification performance than a **non-hierarchical** baseline
- Makes **less severe mistakes**

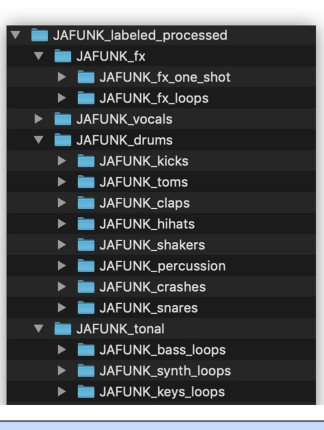
- Enables musical instrument recognition for **classes** defined by the **end user**
- A step towards deployment of instrument recognition in audio editing software

1. Musical Instrument Recognition

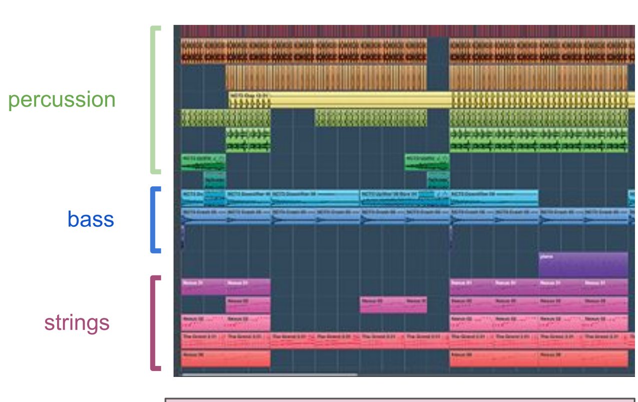


Identifying and locating musical instruments in an audio recording.

Useful for:



Organizing large sample libraries



Grouping tracks in a DAW

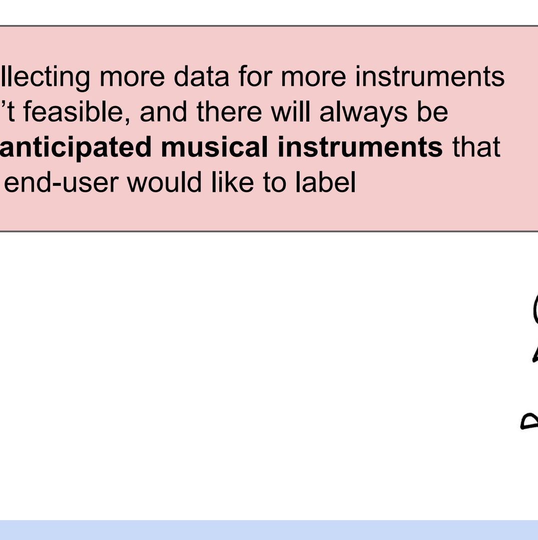
would facilitate eyes-free navigation in DAWs!

Navigating large music recordings by content

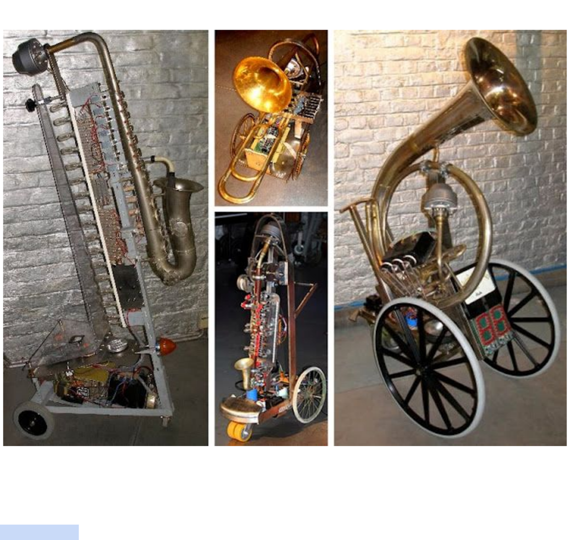
2. Motivation

Current musical instrument recognition models can only handle ~10 musical instruments. ☹

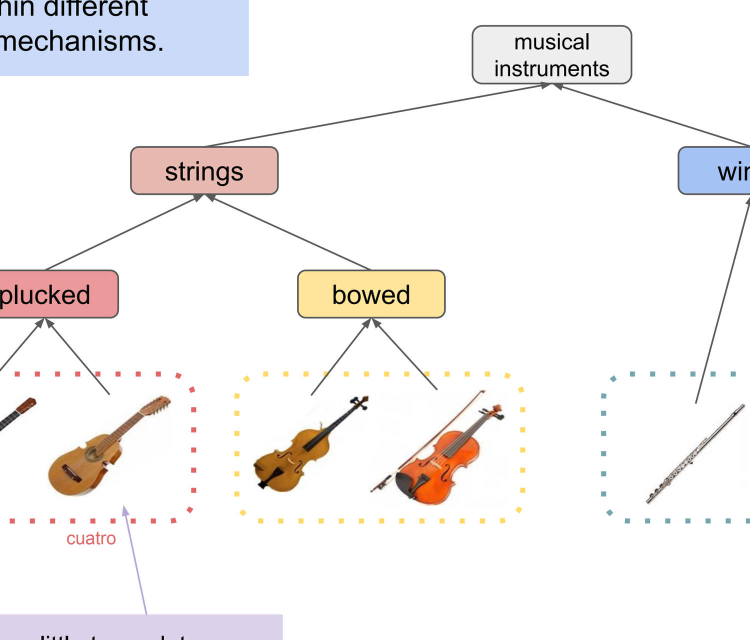
Current datasets



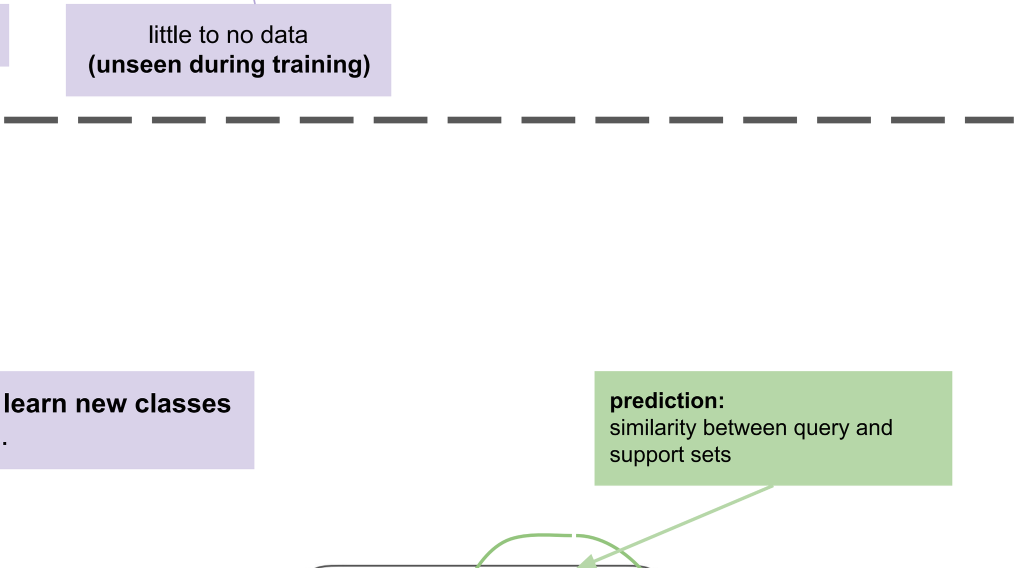
Real world



Collecting more data for more instruments isn't feasible, and there will always be **unanticipated musical instruments** that an end-user would like to label



Musicologists **categorize musical instruments** within different **hierarchical frameworks**, like **sound production** mechanisms.



Can we leverage this hierarchy by **learning** a **feature space** meaningful for **unseen classes** that **share hierarchical ancestry** with classes that have **abundant data**?

lots of data!

little to no data **(unseen during training)**

3. Background

Few Shot Learning

Using few shot learning techniques lets us **learn new classes on the fly** with only a **few** (~10) **examples**.

prediction: similarity between query and support sets

user provides **support** examples for each class, at inference

unlabeled **query** examples

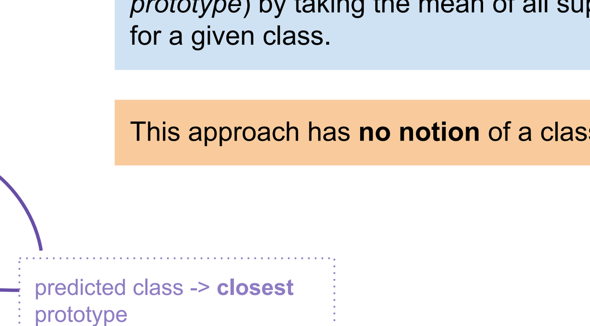
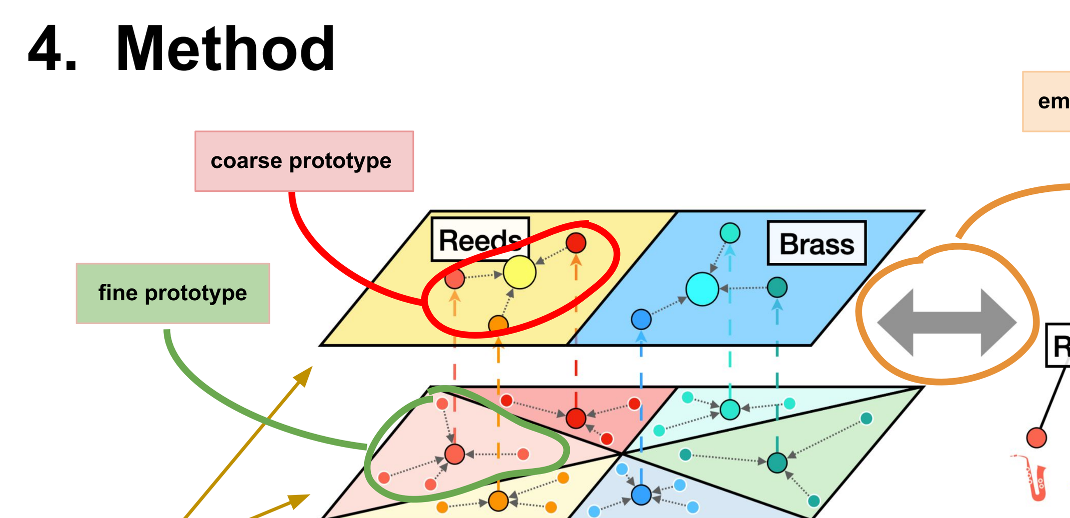


Fig. 2. Metric-based few-shot learning model (*5-way 2-shot*).
(Wang et al., 2020)

Prototypical Networks (Snell et al., 2017)

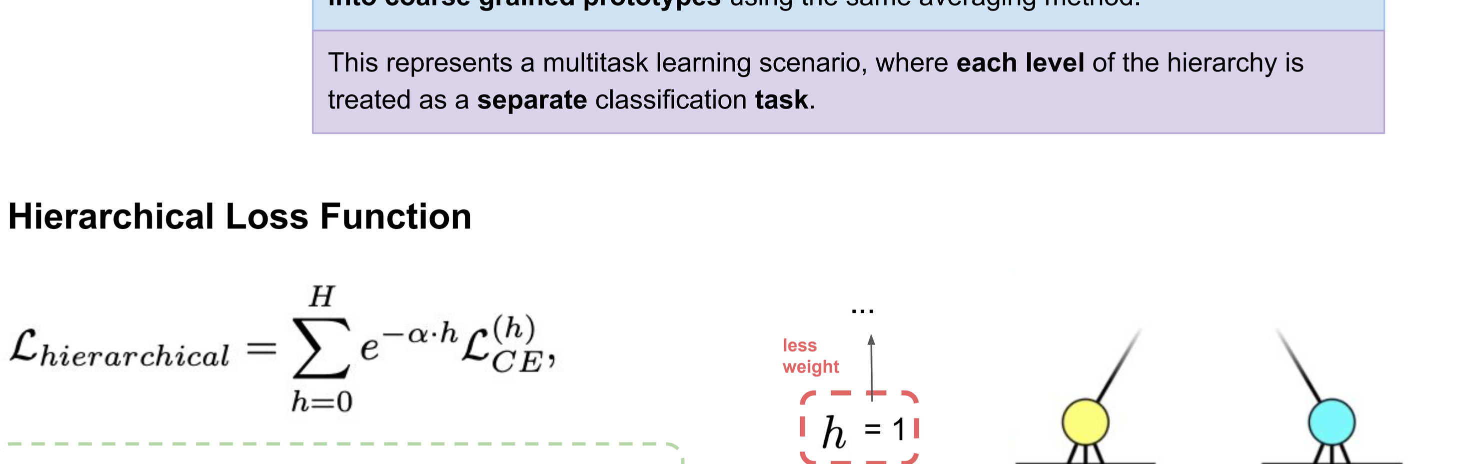


(Snell et al., 2017)

Prototypical networks form a class representation (i.e. *prototype*) by taking the mean of all support examples for a given class.

This approach has **no notion** of a class **hierarchy!**

4. Method



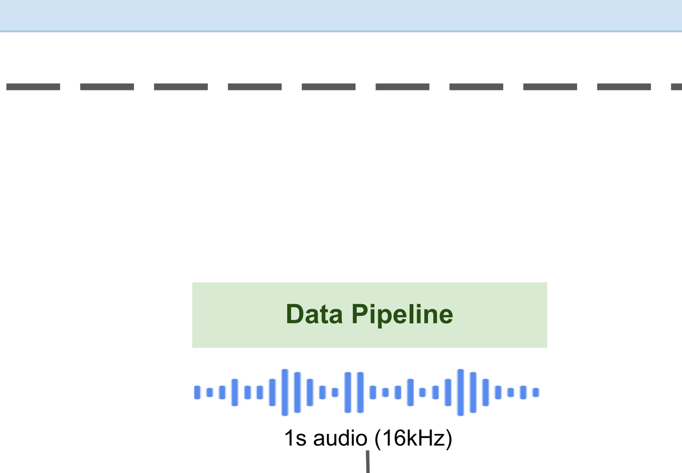
We extend upon prototypical networks by incorporating a **hierarchical label structure during training**, designed to mimic a musical instrument hierarchy. We achieve this by computing a set of prototypes from the support set, and **aggregating** these **prototypes into coarse grained prototypes** using the same averaging method.

This represents a multitask learning scenario, where **each level** of the hierarchy is treated as a **separate** classification **task**.

Hierarchical Loss Function

$$\mathcal{L}_{hierarchical} = \sum_{h=0}^{H} e^{-\alpha \cdot h} \mathcal{L}_{CE}^{(h)},$$

- $\mathcal{L}_{CE}^{(h)}$ cross entropy loss at level h
- α loss decay w.r.t. height
- h height of current level of tree



To **aggregate** these classification **tasks** at different levels of the hierarchy, we design a **weighted hierarchical loss** function that places the **most weight** on the **fine grained** classification **task**, and places **exponentially decreasing** weights on **coarse grained tasks** up the class hierarchy.

5. Experimental Design

Dataset: MedleyDB (Bittner et al. 2014)
- Train on 63 instrument classes
- Test on 24 instrument classes

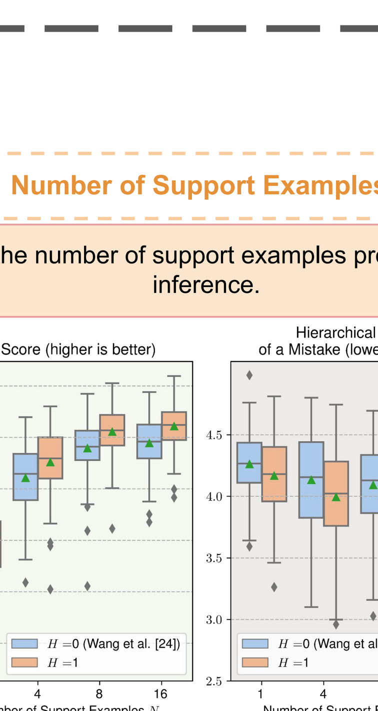
No overlap between train and test classes!
(all classes are unseen at evaluation)

Episodic Training:
- 12-way, 4-shot classification task.
- Train for 60k episodes.

Evaluation (for each experiment)
- 100 episodes
- 12-way, N-shot, with 120 query examples.

Baseline: Non-hierarchical prototypical network (Wang et al. 2020)

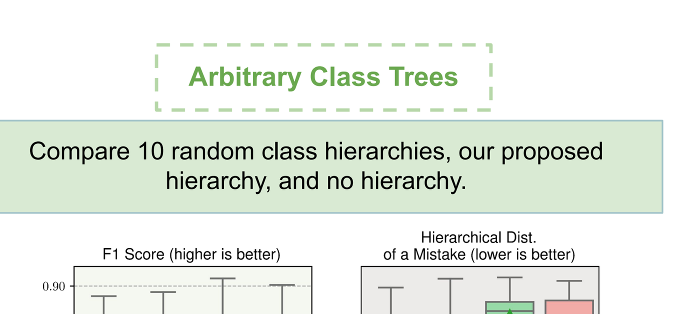
Data Pipeline



6. Experiments

Tree Height

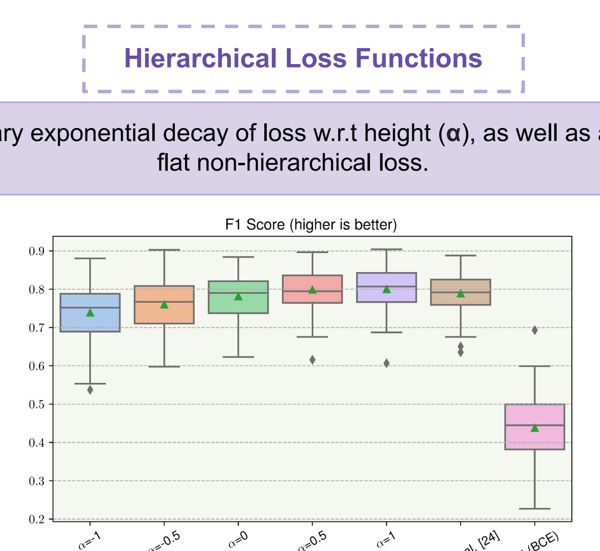
Train using trees with different heights.



- Single-level tree (**H = 1**) is best.
- Deeper trees (**H > 1**) may have less meaningful groupings if the number of classes is small, but may work better if number of classes is larger.

Number of Support Examples

Vary the number of support examples provided at inference.



- **Higher** gains in **F1** when **more support examples** are given.
- Hierarchy results in a **lower severity** of **mistakes**.

Arbitrary Class Trees

Compare 10 random class hierarchies, our proposed hierarchy, and no hierarchy.

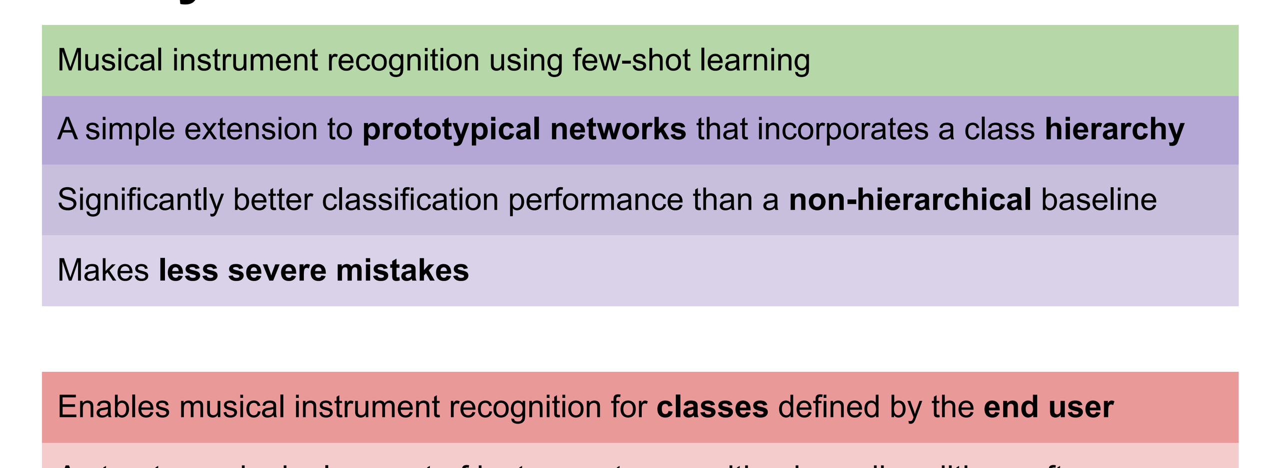
- **random-best** achieves comparable F1 to our proposed hierarchy, but makes severe mistakes.
- **random-worst** causes a severe degradation in classification performance.
- Choice of **hierarchy** is **not arbitrary,** but **other** meaningful **structures** exist!

Hierarchical Loss Functions

Vary exponential decay of loss w.r.t height (**α**), as well as a flat non-hierarchical loss.

- Only **α** > 0 improves over Wang et al. 2020
- "Flat" loss causes a degradation in classification performance.

Examining All Instrument Classes



our model is better at identifying a wider range of instrument classes!

Takeaways

- Musical instrument recognition using few-shot learning
- A simple extension to **prototypical networks** that incorporates a class **hierarchy**
- Significantly better classification performance than a **non-hierarchical** baseline
- Makes **less severe mistakes**

- Enables musical instrument recognition for **classes** defined by the **end user**
- A step towards deployment of instrument recognition in audio editing software

useful links

- arxiv.org/abs/2107.07029
- github.com/hugofloresgarcia/music-trees
- hugofloresgarcia.github.io

thank you!