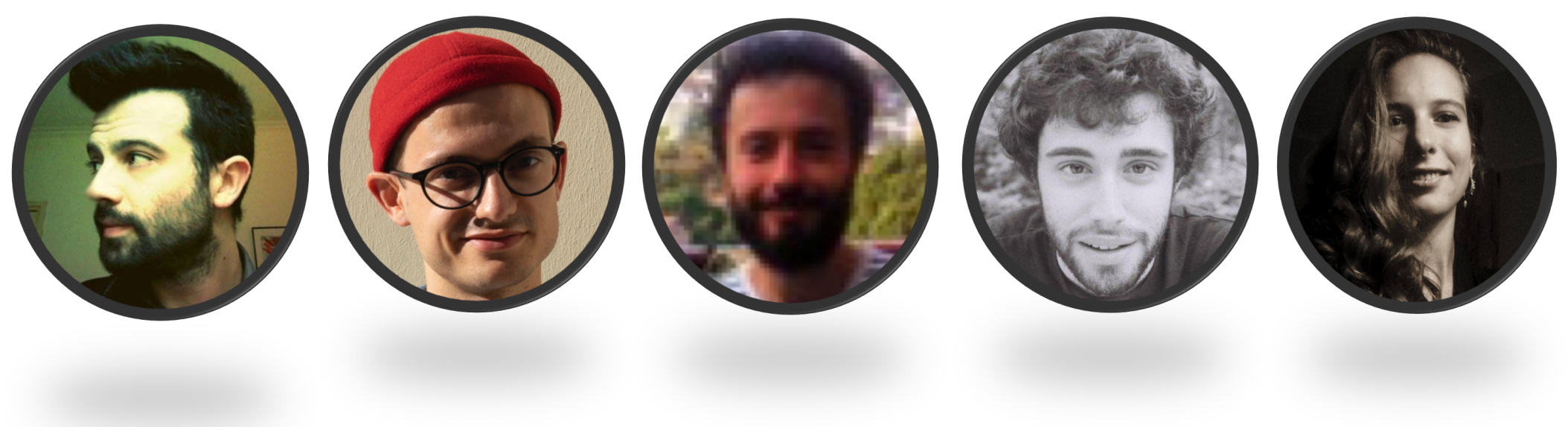
Ultra-light deep MIR by trimming lottery tickets

Philippe Esling, Théis Bazin, Adrien Bitton, Tristan Carsault and Ninon Devis

IRCAM CNRS – UMR 9912, 1 Place Igor Stravinsky, F-75004 Paris, France



Source code / paper

esling@ircam.fr

Abstract

State-of-the-art results in MIR are largely dominated by deep learning. Despite their unprecedented accuracy, their consistently overlooked downside is a stunningly massive complexity, which seems crucial to their success. Here, we address this issue by proposing a model pruning method based on the *lottery ticket hypothesis*. We modify the approach to allow for explicitly removing parameters, through *structured trimming* of entire units, instead of simply masking individual weights. This leads to models which are effectively lighter in terms of size, memory and number of operations.

We show that we can remove up to 90% of the model parameters without loss of accuracy, leading to ultra-light deep MIR models. We confirm the surprising result that, at smaller compression ratios (removing up to 85% of a network), lighter models consistently outperform their heavier counterparts. We exhibit these results on a large array of MIR tasks including *audio classification*, *pitch recognition*, *chord extraction*, *drum transcription* and *onset estimation*.

Deep accuracy madness

Deep learning holds most state-of-art results in MIR

However, some key problems of deep machine learning

- Networks can have up to **billions of parameters**
- **Gains in accuracy** now appear always **linked to increased size**
- Extremely demanding in **computation, energy and memory**.
- Huge **environmental issues** of deep models

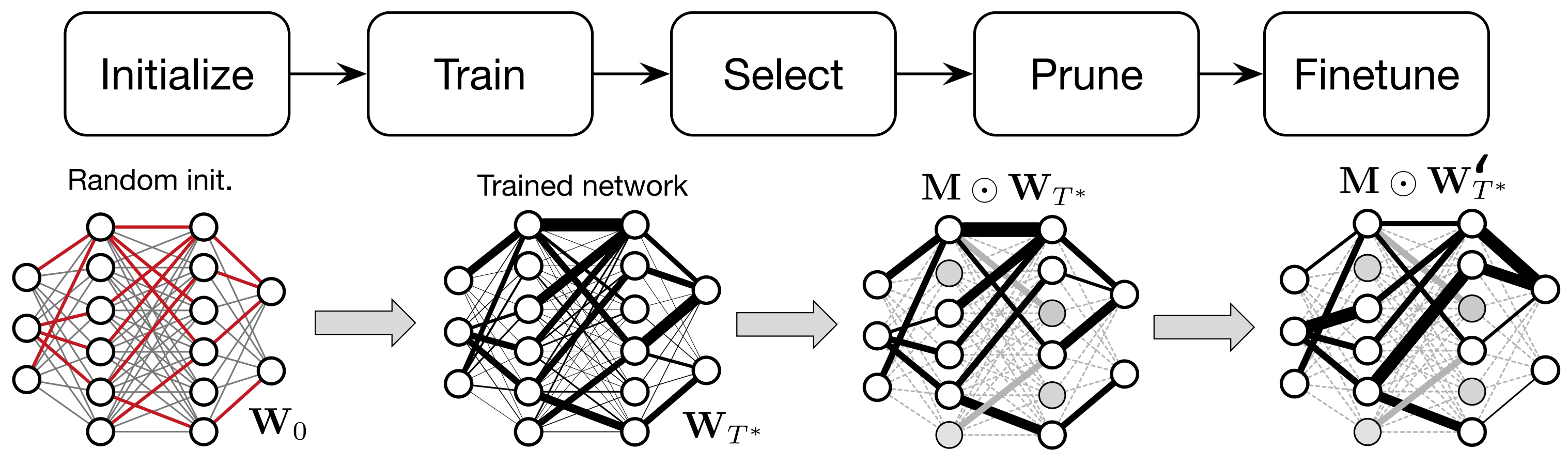
Showering example of the MegatronML model

- **8.3 billion** parameters trained on **512 V100 GPUs** for **9.2 days**
- Now **GPT-3** even goes to **175 billion** parameters
- Trend is starting in MIR (CREPE for pitch has 22 million parameters)

The quest for **smaller** models that **have the same accuracy**

Compressing networks by pruning

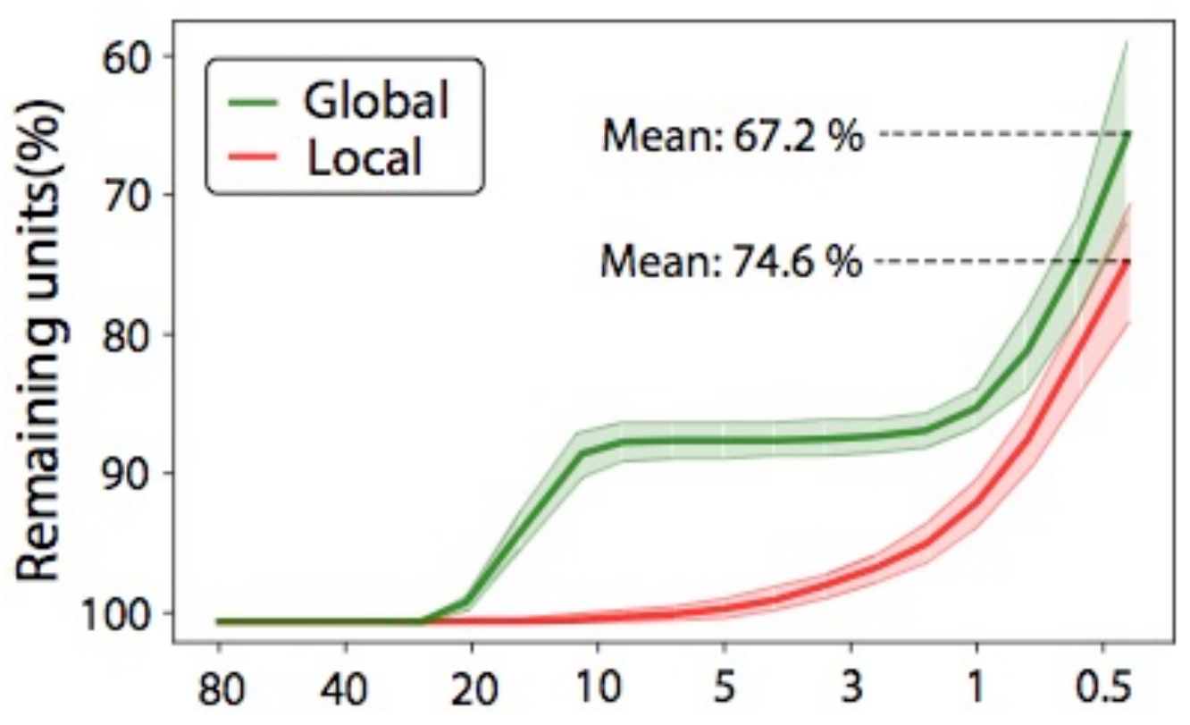
- The seminal approach for reducing models is *pruning*.
- Remove the **low-magnitude weights** and then finetune the masked net



- Finetuning **mostly unable** to keep accuracy and **poor compression**

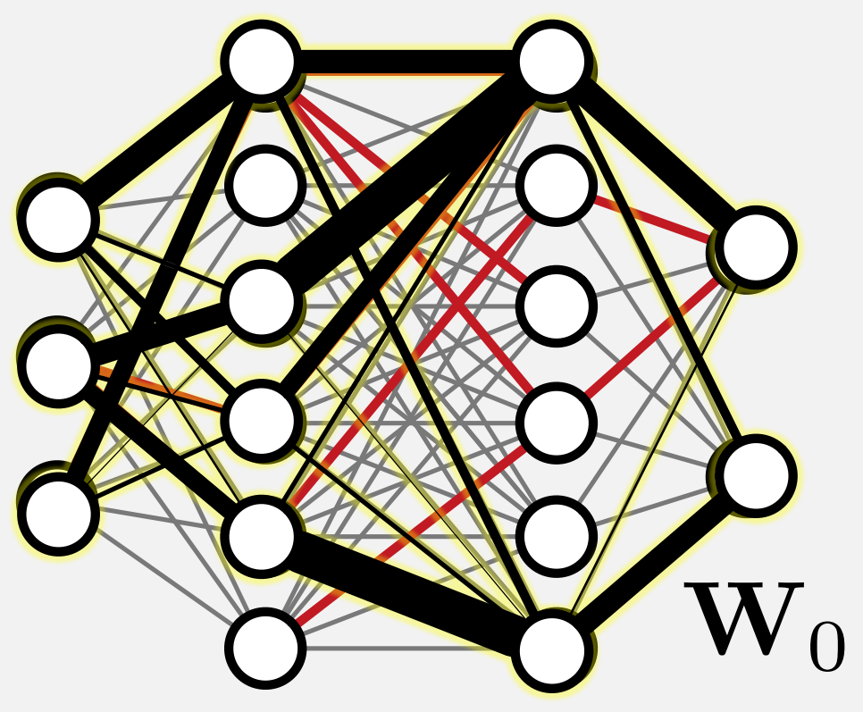
Limitations of the lottery

- Masking is not very efficient
- Masking 99% allows to remove ~25%
- No straightforward gains in
  - Computation nor memory



The *lottery ticket hypothesis*

The recently introduced *lottery ticket hypothesis* states that



- Inside randomly-initialized (untrained) neural networks
- Already exist **powerful very small subnetworks** (*winning tickets*)
- If trained in isolation, achieve **higher accuracy than large models**

Stunning results in the image domain

- **Higher accuracy** with models **removing 90% of the network**
- **Maintain accuracy** even when pruning up to **99% of the weights**

Obtaining these very light networks requires *weight rewinding* and *iterative pruning*

- Initialize weights randomly
- Train network to completion
- Select weights based on magnitude
- Mask low-magnitude weights
- Rewind the weights to prior iteration
- Retrain the resulting small network

Algorithm 1 Lottery ticket training with rewinding

```
$\boldsymbol{W}_0 \sim p(\boldsymbol{W})$                                  ▷ Random initialization
$\boldsymbol{M} = \mathbf{1}_{|\boldsymbol{W}|}$                                   ▷ Initial mask
$\boldsymbol{W}_k = \mathcal{A}(k, \boldsymbol{W}_0 \odot \boldsymbol{M})$                    ▷ Training for $k$ iterations
while $\mathcal{C}(\boldsymbol{M}, a, \boldsymbol{W})$ do                      ▷ Stopping criterion $\mathcal{C}$
    $\boldsymbol{W}_T = \mathcal{A}(T, \boldsymbol{W}_k \odot \boldsymbol{M})$                ▷ Train until completion
    $r = \mathcal{R}(\{\boldsymbol{W}_{T^*}\})$                          ▷ Ranking criterion $\mathcal{R}$
    $\boldsymbol{M} = \mathcal{M}(r, \{\boldsymbol{W}_{T^*}\})$                      ▷ Masking update $\mathcal{M}$
```

Our proposed structured trimming

- To obtain truly lighter networks, we propose *structured trimming*
- Consider layers as matrix $\mathbf{W}^{(l)} \in \mathbb{R}^{N_{out} \times N_{in}}$ and remove units $\mathbf{W}_i^{(l)}$
- We evaluate three different *selection criteria*

Magnitude-based
Find the units with maximal summed magnitude

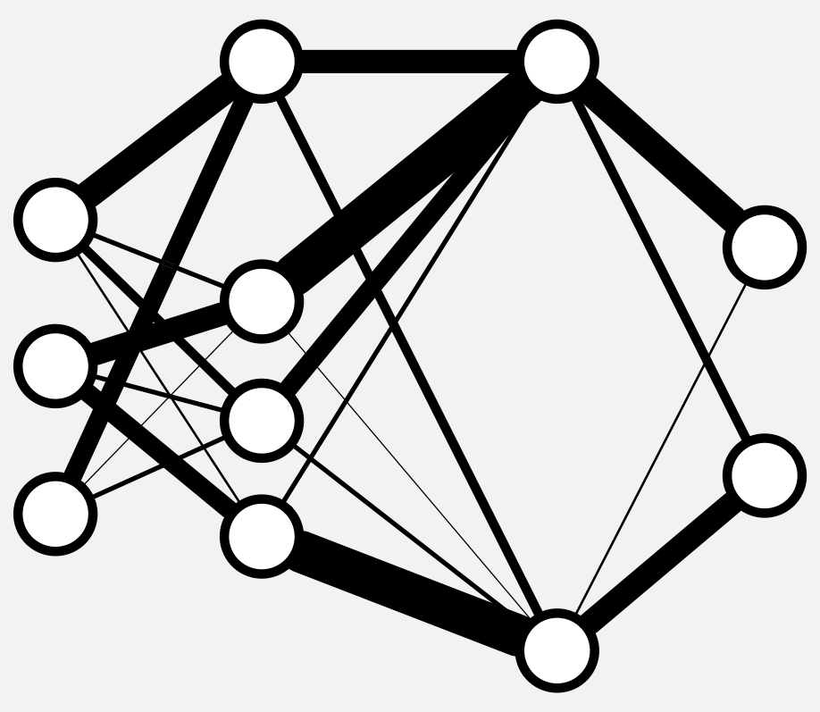
$$\mathcal{C}_{mag}(\mathbf{W}_I^{(l)}) = \sum_{j=1}^{N_{in}} \left| W_{i,j}^{(l)} \right| .$$

Activation-based
Find units maximally activated by the train set

$$\mathcal{C}_{act}(\mathbf{W}_I^{(l)}) = \sum_{k=1}^{\mathcal{D}_v} \left| f(\mathbf{x}_k, \mathbf{W}^{(l)})_I \right|$$

Normalization-based
Use the *scaling* factor as a proxy for usage

$$\mathcal{C}_{norm}(\mathbf{W}_I^{(l)}) = \left| \gamma_i^{(l)} \right|$$



Structured lottery
$\bar{\mathbf{W}}_{T'}^{[\mathbf{M}]}$

Experiments

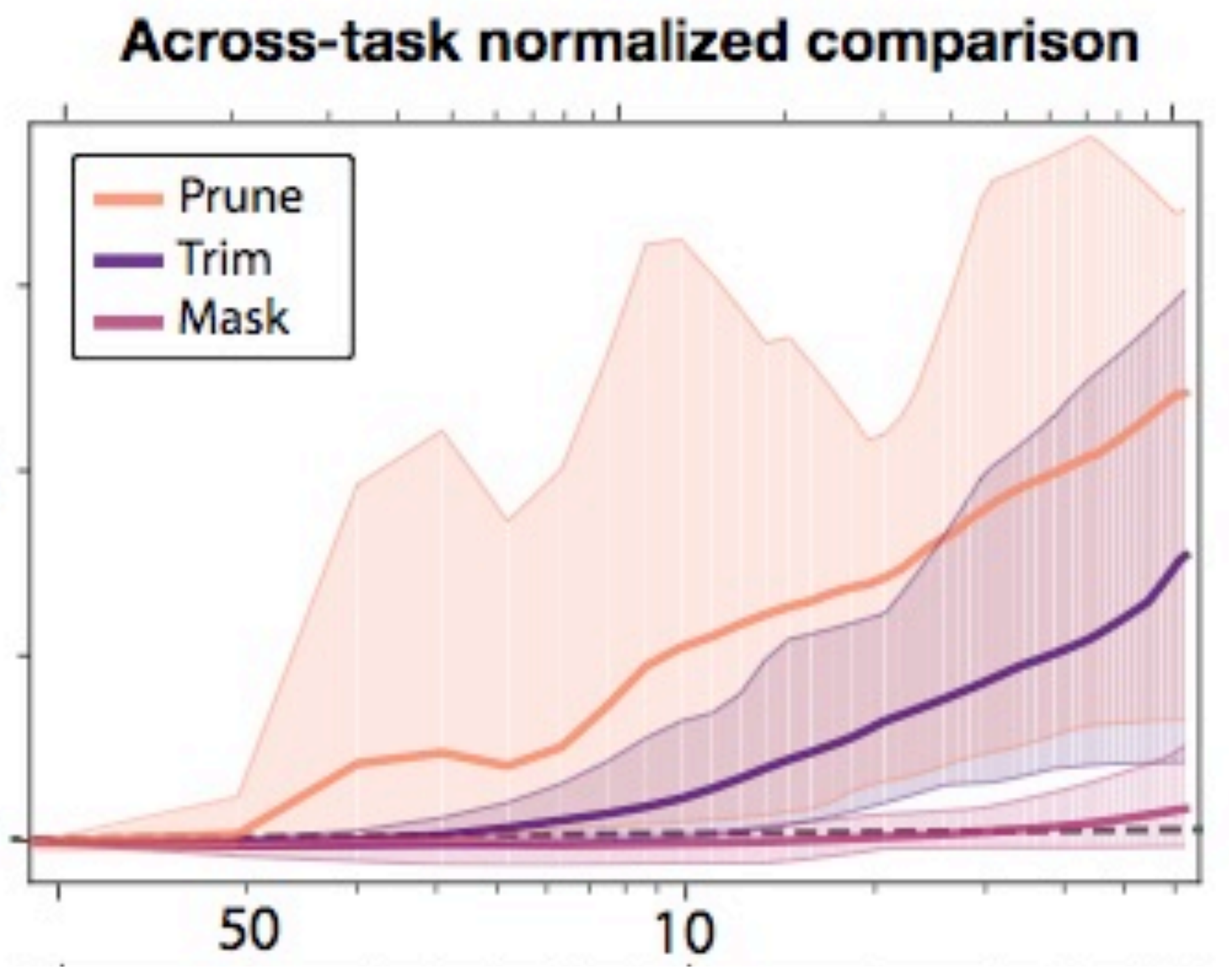
We test our *structured trimming* on several MIR tasks

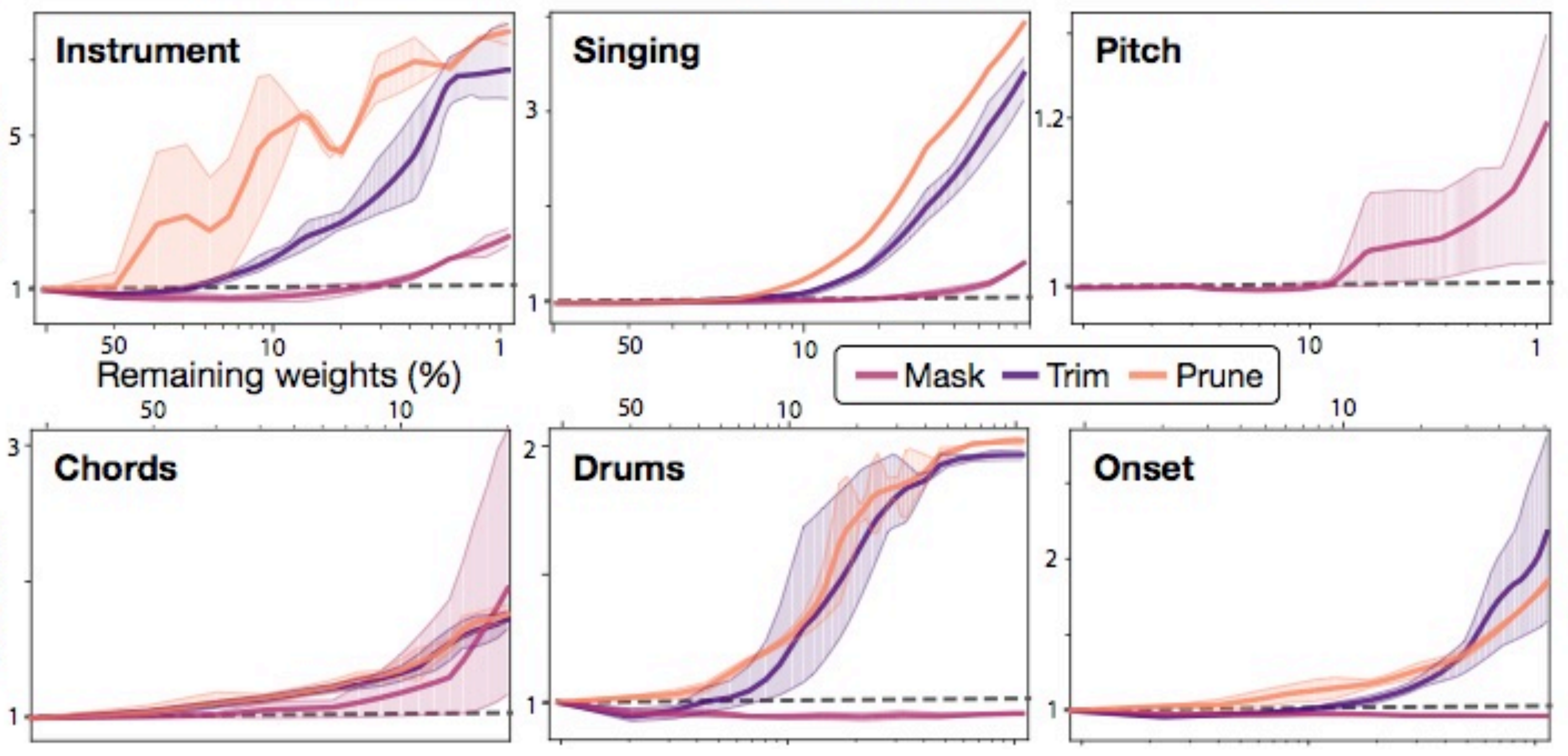
- **Singing voice classification** (Dilated CNN – MLP model on waveform)
- **Instrument classification** (Same dilated CNN – MLP on different set)
- **Pitch estimation** (CREPE – Large 6-layers 1d-CNN on *waveform*)
- **Drum transcription** (2d-CNN on Mel with 3-layer MLP per drum type).
- **Onset detection** (Same as *drum transcription* with a single output)
- **Automatic Chord Extraction** (2d-CNN architecture based on CQT)

All are deep approaches with various types of input data and tasks

Methods and accuracy results

- *Trimming also* finds efficient **smaller nets with higher accuracy** when removing up to 75% weights
- Allows models **up to 10 times smaller** that **maintains the accuracy** of large nets
- *Masking* still outperforms *trimming* (but corresponding weights are not removed)
- Similar results hold for all tasks (classification, pitch, chords, drums and onset)





Efficiency results

Most interesting results can be seen in

- Relationships *disk size, FLOPS* and *memory*

task	param	size	FLOPS	mem
inst.	797 K 93.4 K	10 M 2.3 M	572 M 38.3 M	190 M 41.9 M
sing.	1.4 M 144 K	19 M 2.7 M	663 M 94.4 M	194 M 53.2 M
pitch	5.9 M 224 K	49 M 1.0 M	2.8 G 2.8 M	256 M 9.6 M
chord	416 K 91.9 K	1.4 M 0.2 M	27.2 M 1.72 M	22.1 M 589 K
drum	8.1 M 1.0 M	22 M 3.7 M	3.54 G 87.5 M	667 M 10.2 M
onset	4.7 M 522 K	21 M 3.7 M	2.66 G 87.1 M	532 M 8.2 M

Networks that are 10 times smaller …

… are 100 times more efficient

- Logical due to quadratic complexity

Musical Representations team
IRCAM/STMS/UPMC/Sorbonne Université
acids.ircam.fr