Equivariant Self-Supervision For Musical Tempo Estimation

Elio Quinton [1]

[1]*Music, Audio and Machine Learning Lab*
Universal Music Group

Abstract

We propose to use **equivariance** as a self-supervision signal to learn audio tempo representations from unlabelled data.

How does it work?

1. **Timestretching** with random ratio applied twice to each training sample
2. **Forward pass** to get representation
3. **Equivariant objective**: representation constrained to reflect change in tempo

What do we get out of it?

- **Meaningful representations** can be learnt with this method
- **Simple loss function** that prevents the collapse on a trivial solution during training, without requiring regularisation or negative sampling
- **Light weight & accessible** Small model (33k params) + no need for large batch sizes = light compute requirement (trains on 1 GPU)
- **Robust** to hyper-parameter choices

What could it be useful for?

- **Low resource genres**. Typically underserved by supervised methods because most annotated datasets are western
- **General music representations**. This equivariant objective is simple enough to be used as complement to other self-supervised objectives
- **Inspiration** for further exploration of equivariant self-supervision. Well suited to regression problems

Background

Equivariance

An *invariant* representation $q(w)$ remains unchanged for a transformation k of the input w:

$$q(k \cdot w) = q(w) \tag{1}$$

An *equivariant* representation reflects the transformation applied to the input:

$$q(k \cdot w) = k \cdot q(w) \tag{2}$$

State of the art

- Tempo estimation is treated as supervised learning. Requires large quantity of labelled data.
- Self-supervised approaches based on siamese networks usually rely on *invariance* to data augmentations.
- SPICE [4] was first musical application of equivariant self-supervision for pitch estimation. But requires strong regularisation.

We propose an approach to learn tempo representations from unlabelled data, using *equivariance* as a simple self-supervision supervision objective, which does not require regularisation.

Method

Model

Temporal Convolutional Network (TCN) [1] with 33k parameters. Mel Spectrogram as input and outputs a 16-d embedding $\mathbf{h}$.

Training Strategy

For each training sample x, apply time-stretching transformations t_i and t_j, with rate α_i and α_j drawn at random from $[1-r, 1+r]$ where r is a hyper-parameter, and feed through the TCN and projection head, as depicted in Fig.1. The equivariant objective is then applied via the proposed loss function:

$$\mathcal{L} = \left| \frac{z_i}{z_j} - \frac{\alpha_i}{\alpha_j} \right| \tag{3}$$

Datasets

MagnaTagaTune (MTT) for self-supervised pre-training. GTZAN, Hainsworth, Giantsteps, and ACM Mirum for fine-tuning and evaluation.

Evaluation

Discard projection head $g(\cdot)$ and freeze network weight. Attach a linear classification head with 300 units and softmax layer to represent the range [0,300] BPM and fine-tune with cross-entropy loss. We perform cross-dataset evaluation and report *Accuracy 1* and *Accuracy 2* scores with a $\pm 4\%$ tolerance.

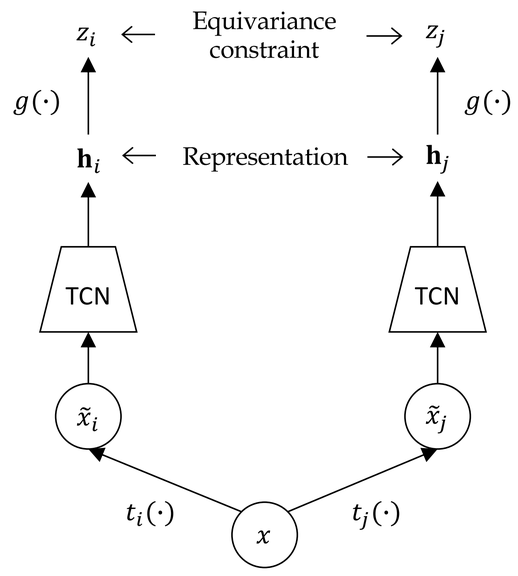

Figure 1. Equivariant self-supervision framework. Random time-stretching transformations t_i and t_j are applied to a training sample x. The TCN and projection head $g(\cdot)$ are trained to produce a pseudo-tempo scalar z equivariant to the time stretching transformation of the input. The projection head is discarded after training.

Results

Trivial solutions Loss function $\mathcal{L}$ does not collapse to trivial solutions

Influence of pre-training augmentation Random audio augmentations yield not notable difference. Timestretching strength (r_p) only hurts performance for extreme values. Cf. Fig 2.

Influence of fine-tuning augmentation Applying time-stretching ($r_f > 0$) generally improves performance. Optimal value varies from one dataset to the next. Cf. Table 1.

Comparison to supervised benchmarks Accuracy 2 performance is comparable with supervised benchmarks on all datasets. Accuracy 1 performance is more inconsistent.

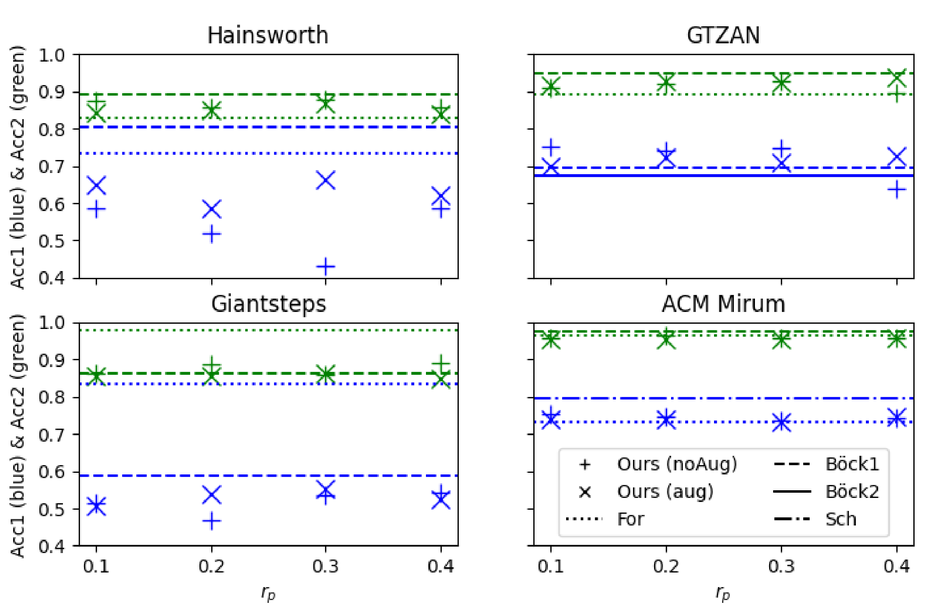

Figure 2. Performance against supervised benchmarks. Acc1 in blue and Acc2 in green. Results for combinations of using augmentations ("aug") or not ("noAug") and of $r_p \in \{0.1, 0.2, 0.3, 0.4\}$. In all cases the model is fine-tuned with a time-stretching of strength $r_f = 0.2$. Horizontal lines are supervised benchmarks "Böck1" [2], "Böck2" [1], "For" [3], "Sch" [5]. We only display the highest and lowest performing baselines.

Method	r_f	Hainsworth		GTZAN		Giantsteps		ACM Mirum	
		Acc1	Acc2	Acc1	Acc2	Acc1	Acc2	Acc1	Acc2
Ours	0.0	0.604	*0.838*	*0.691*	0.887	0.512	0.809	0.704	0.943
Ours	0.1	0.586	0.824	*0.719*	0.881	0.456	0.791	*0.757*	*0.965*
Ours	0.2	0.518	*0.856*	*0.741*	*0.919*	0.470	*0.886*	*0.747*	*0.965*
Ours	0.3	0.550	*0.829*	**0.785**	*0.921*	0.438	0.846	0.700	0.958
Ours	0.4	0.541	*0.829*	*0.778*	*0.926*	0.472	*0.884*	0.724	0.952
Schreiber [5]	-	0.770	0.842	0.694	0.926	0.730	0.893	**0.795**	0.974
Foroughmand [3]	-	0.734	0.829	0.697	0.891	**0.836**	**0.979**	0.733	0.965
Böck 1 [2]	-	**0.806**	**0.892**	0.697	**0.950**	0.589	0.864	0.741	**0.976**
Böck 2 [1]	-	-	-	0.673	0.938	0.764	0.958	0.749	0.974

Table 1. Influence of fine-tuning time-stretching strength. Pre-trained on MTT with $r_p = 0.2$ and no audio augmentations. Highest performance for each metric and dataset shown in bold. Italics metrics where our model outperforms at least one baseline.

References

[1] Sebastian Böck, Matthew EP Davies, and Peter Knees. Multi-task learning of tempo and beat: learning one to improve the other. In *20th International Society for Music Information Retrieval Conference (ISMIR 2019)*, 2019.

[2] Sebastian Böck, Florian Krebs, and Gerhard Widmer. Accurate tempo estimation based on recurrent neural networks and resonating comb filters. In *International Society for Music Information Retrieval (ISMIR) Conference*, 2015.

[3] Hadrien Foroughmand and Geoffroy Peeters. Deep-rhythm for tempo estimation and rhythm pattern recognition. In *International Society for Music Information Retrieval (ISMIR)*, 2019.

[4] Beat Gfeller, Christian Frank, Dominik Roblek, Matt Sharifi, Marco Tagliasacchi, and Mihajlo Velimirović. SPICE: Self-supervised pitch estimation. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 28:1118–1128, 2020.

[5] Hendrik Schreiber and Meinard Müller. A Single-Step Approach to Musical Tempo Estimation Using a Convolutional Neural Network. In *International Society for Mu- sic Information Retrieval (ISMIR) Conference*, pages 98–105, 2018.