PESTO: PITCH ESTIMATION WITH SELF-SUPERVISED TRANSPOSITION-EQUIVARIANT OBJECTIVE

Alain Riou\textsuperscript{1,2} Stefan Lattner\textsuperscript{2} Gaëtan Hadjeres\textsuperscript{3} Geoffroy Peeters\textsuperscript{1}

\textsuperscript{1} LTCI, Télécom-Paris, Institut Polytechnique de Paris, France
\textsuperscript{2} Sony Computer Science Laboratories - Paris, France
\textsuperscript{3} Sony AI

alain.riou@sony.com

ABSTRACT

In this paper, we address the problem of pitch estimation using Self Supervised Learning (SSL). The SSL paradigm we use is equivariance to pitch transposition, which enables our model to accurately perform pitch estimation on monophonic audio after being trained only on a small unlabelled dataset. We use a lightweight (< 30k parameters) Siamese neural network that takes as inputs two different pitch-shifted versions of the same audio represented by its Constant-Q Transform. To prevent the model from collapsing in an encoder-only setting, we propose a novel class-based transposition-equivariant objective which captures pitch information. Furthermore, we design the architecture of our network to be transposition-preserving by introducing learnable Toeplitz matrices.

We evaluate our model for the two tasks of singing voice and musical instrument pitch estimation and show that our model is able to generalize across tasks and datasets while being lightweight, hence remaining compatible with low-resource devices and suitable for real-time applications. In particular, our results surpass self-supervised baselines and narrow the performance gap between self-supervised and supervised methods for pitch estimation.

1. INTRODUCTION

Pitch estimation is a fundamental task in audio analysis, with numerous applications, e.g., in Music Information Retrieval (MIR) and speech processing. It involves estimating the fundamental frequency of a sound, which allows to estimate its perceived pitch. Over the years, various techniques have been developed for pitch estimation, ranging from classical methods (based on signal processing) [1–4] to machine learning approaches [5, 6].

In recent years, deep learning has emerged as a powerful tool for a wide range of applications, outperforming classical methods in many domains. This is notably true in MIR, where deep learning has led to significant advances in tasks such as music transcription [7–9], genre classification [10–12], and instrument recognition [13–15]. Pitch estimation has also benefited greatly from deep learning techniques [16, 17]. However, these deep learning models often require a large amount of labelled data to be trained, and can be computationally expensive, hindering their practical applications in devices with limited computing power and memory capabilities. Additionally, these models are often task-specific and may not generalize well to different datasets or tasks [18]. Therefore, there is a need for a lightweight and generic model that does not require labelled data to be trained. We address this here.

We take inspiration from the equivariant pitch estimation [19] and the equivariant tempo estimation [20] algorithms which we describe in part 2. As those, we use a SSL paradigm based on Siamese networks and equivariance to pitch transpositions (comparing two versions of the same sound that have been transposed by a random but known pitch shift). We introduce a new equivariance loss that enforces the model to capture pitch information specifically.

This work has the following contributions:

• we formulate pitch estimation as a multi-class problem (part 3.1); while [19, 20] model pitch/tempo estimation as a regression problem,

• we propose a novel class-based equivariance loss (part 3.1) which prevents collapse; while [19] necessitates a decoder,

• the architecture of our model is lightweight and transposition-equivariant by design. For this, we introduce Toeplitz fully-connected layers (part 3.4).

We evaluate our method on several datasets and show that it outperforms self-supervised baselines on single pitch estimation (part 4.4.1). We demonstrate the robustness of our method to domain-shift and background music, highlighting its potential for real-world applications (part 4.4.2).

Our proposed method requires minimal computation resources and is thus accessible to a wide range of users for both research and musical applications. In consideration of accessibility and reproducibility, we make our code and pretrained models publicly available\textsuperscript{1}.

\textsuperscript{1}https://github.com/SonyCSLParis/pesto

© A. Riou, S. Lattner, G. Hadjeres and G. Peeters. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0).

2. RELATED WORKS

2.1 SSL to learn invariant representations.

Siamese networks. Most common techniques for SSL representation involve Siamese networks [21]. The underlying idea is to generate two views of an input, feed them to a neural network, and train the network by applying a criterion between the output embeddings. Various techniques have been developed for generating views\(^2\).

Collapse. However, a major issue with these methods is “collapse”, when all inputs are mapped to the same embedding. To address this, various techniques have been proposed. One of the most common is SimCLR [22] which also uses negative samples to ensure that embeddings are far apart through a contrastive loss. Additionally, several regularization techniques have been developed that minimize a loss over the whole batch. Barlow Twins [23] force the cross-correlation between embeddings to be identity, while VICReg [24] add loss terms on the statistics of a batch to ensure that dimensions of the embeddings have high enough variance while remaining independent of each other. On the other hand, [25] explicitly minimize a loss over the hypersphere to distribute embeddings uniformly. Furthermore, incorporating asymmetry between inputs has been shown to improve performance. [26, 27] uses a momentum encoder, while [28] and [29] add a projection head and a stop-gradient operator on top of the network, with [28] also using a teacher network. Finally, [30] incorporates asymmetry to contrastive- and clustering-based representation learning.

Application to audio. While originally proposed for computer vision, these methods have been successfully adapted to audio and music as well. For example, [31], [32], and [33] respectively adapted [22], [23], and [28] to the audio domain. By training their large models on AudioSet [34], they aim at learning general audio representations that are suited for many downstream tasks. More specifically, [35] successfully adapts contrastive learning to the task of music tagging by proposing more musically-relevant data augmentations.

2.2 SSL to learn equivariant representations.

The purpose of the methods described above is to learn a mapping \( f : X \rightarrow Y \) that is invariant to a set of transforms \( T_X \), i.e. so that for any input \( x \in X \) and transform \( t \in T_X \)
\[
f(t(x)) \approx f(x) \tag{1}
\]

However, recent approaches [36–38] try instead to learn a mapping \( f \) that is equivariant to \( T_X \), i.e. that satisfies
\[
f(t'x) \approx t'(f(x)) \tag{2}
\]
where \( t' \in T_Y \) with \( T_Y \) a set of transforms that acts on the output space \( Y \). In other words, if the input is transformed, the output should be transformed accordingly. Representation collapse is hence prevented by design.

Equivariant representation learning has mostly been applied to computer vision and usually combines an invariance and an equivariance criterion. E-SSL [36] trains two projection heads on top of an encoder, one to return projections invariant to data augmentations while the other predicts the parameters of the applied data augmentations. [37] predicts separately a semantic representation and a rotation angle of a given input and optimizes the network with a reconstruction loss applied to the decoded content representation rotated by the predicted angle. Finally, SIE [38] creates a pair of inputs by augmenting an input and learns equivariant representations by training a hypernetwork conditioned on the parameters of the augmentation to predict one embedding of the pair from the other.

Application to audio. Finally, a few successful examples of equivariant learning for solving MIR tasks recently emerged [19,20]. In particular, [20] introduces a simple yet effective equivariance criterion for tempo estimation while preventing collapse without any decoder or regularization: pairs are created by time-stretching an input with two different ratios, then the output embeddings are linearly projected onto scalars and the network is optimized to make the ratio of the scalar projections match the time-stretching ratio within a pair.

2.3 Pitch estimation.

Monophonic pitch estimation has been a subject of interest for over fifty years [39]. The earlier methods typically obtain a pitch curve by processing a candidate-generating function such as cepstrum [39], autocorrelation function (ACF) [40], and average magnitude difference function (AMDF) [41]. Other functions, such as the normalized cross-correlation function (NCCF) [1,2] and the cumulative mean normalized difference function [3,42], have also been proposed. On the other hand, [4] performs pitch estimation by predicting the pitch of the sawtooth waveform whose spectrum best matches the one of the input signal.

Recently, methods involving machine learning techniques have been proposed [5,6]. In particular, CREPE [16] is a deep convolutional network trained on a large corpus to predict pitch from raw audio waveforms. SPICE [19] is a self-supervised method that takes as inputs individual Constant-Q Transform (CQT) frames of pitch-shifted inputs and learns the transposition between these inputs. It achieves quite decent results thanks to a decoder that takes as input the predicted pitch and tries to reconstruct the original CQT frame from it.

Finally, some works [43,44] aim at disentangling the pitch and timbre of an input audio, thus predicting pitch as a side effect. In particular, DDSN-inv [45] is a DDSP-based approach that relies on inverse synthesis to infer pitch in a self-supervised way.

3. SELF-SUPERVISED PITCH ESTIMATION

3.1 Transposition-equivariant objective

We focus on the problem of monophonic pitch estimation and model it as a classification task. Our model is com-
Figure 1. Example of $k$-transpositions. Visually, $y$ and $y'$ are just translated versions of each other. The sign of $k$ and its absolute value respectively indicate the direction and the distance of the translation.

posed of a neural network $f_0$ that takes as input an audio signal $x$ and returns a vector $y = (y_0, \ldots, y_i, \ldots, y_{d-1}) \in [0, 1]^d$, which represents the probability distribution of each pitch $i$. $y_i$ represents the probability that $i$ is the pitch of $x$. We propose here to train $f_0$ in a SSL way. For this, similarly to [22, 24, 26, 28, 29], we use data augmentations and Siamese networks.

Given $x$, we first generate $x^{(k)}$ by pitch-shifting $x$ by a known number $k$ of semitones. Then, both $x$ and $x^{(k)}$ are fed to $f_0$ which is trained to minimize a loss function between $y = f_0(x)$ and $y^{(k)} = f_0(x^{(k)})$.

**Definition.** For two vectors $y, y' \in \mathbb{R}^d$ and $0 \leq k < d$, $y'$ is a $k$-transposition of $y$ if and only if for all $0 \leq i < d$

$$y'_i = \begin{cases} y_i & \text{if } 0 \leq i \leq d-k \\ 0 & \text{if } i < k \\ y_{i+k} & \text{if } i \geq d-k+1 \end{cases}$$

Similarly, for $-d < k \leq 0$, $y'$ is a $-k$-transposition of $y$ if and only if $y$ is a $-k$-transposition of $y'$.

The concept of $k$-transposition is illustrated in Figure 1. Note also that for a vector $y \in \mathbb{R}^d$, exists at most one vector $y' \in \mathbb{R}^d$ that is a $k$-transposition of $y$. We can therefore refer to $y'$ as the $k$-transposition of this vector $y$.

**Equivariance loss.** We then design our criterion based on the following assumption: the probability of $x$ to have pitch $i$ is equal to the probability of $x^{(k)}$ to have pitch $i+k$, i.e. $y_i$ should be equal to $y_{i+k}^{(k)}$. In other words, if $x^{(k)}$ is a pitch-shifted version of $x$, their respective pitch probability distributions should be shifted accordingly, i.e. $y^{(k)}$ should be the $k$-transposition of $y$.

We take inspiration from [20] to design our equivariance loss. However, in our case, the output of our network $f_0$ is not a generic representation but a probability distribution. We therefore adapt our criterion by replacing the learnable linear projection head from [20] by the following deterministic linear form:

$$\phi : \mathbb{R}^d \rightarrow \mathbb{R} \quad y \mapsto (\alpha, \alpha^2, \ldots, \alpha^d)y$$

where $\alpha$ is a fixed hyperparameter. \footnote{For example, if $k = 2$ semitones, the probability of $x$ to be C4 is exactly the probability of $x^{(k)}$ to be a D4, and the same holds for any pitch independently of the actual pitch of $x$.}

Indeed, with this formulation, for any $k$ if $y'$ is a $k$-transposition of $y$ then $\phi(y') = \alpha^k \phi(y)$. Hence we define our loss as

$$L_{\text{equiv}}(y, y^{(k)}, k) = h_\tau \left( \frac{\phi(y^{(k)})}{\phi(y)} - \alpha^k \right)$$

where $h_\tau$ is the Huber loss function \footnote{We found $\alpha = 2^{3/36}$ to work well in practice.} defined by

$$h_\tau(x) = \begin{cases} \frac{x^2}{2} & \text{if } |x| \leq \tau \\ \frac{\tau^2}{2} + \tau (|x| - \tau) & \text{otherwise} \end{cases}$$

**Regularization loss.** Note that if $y^{(k)}$ is the $k$-transposition of $y$ then $L_{\text{equiv}}(y, y^{(k)}, k)$ is minimal. However, the converse is not always true. In order to actually enforce pitch-shifted pairs of inputs to lead to $k$-transpositions, we further add a regularization term which is simply the shifted cross-entropy (SCE) between $y$ and $y^{(k)}$, i.e. the cross-entropy between the $k$-transposition of $y$ and $y^{(k)}$:

$$L_{\text{SCE}}(y, y^{(k)}, k) = \sum_{i=0}^{d-1} y_i \log \left( y_{i+k}^{(k)} \right)$$

with the out-of-bounds indices replaced by 0. The respective contribution of $L_{\text{equiv}}$ and $L_{\text{SCE}}$ is studied in part 4.4.3.

**Invariance loss.** $L_{\text{equiv}}$ and $L_{\text{SCE}}$ allow our model to learn relative transpositions between different inputs and learn to output probability distributions $y$ and $y^{(k)}$ that satisfy the equivariance constraints. However, these distributions may still depend on the timbre of the signal. This is because our model actually never observed at the same time two different samples with the same pitch.

To circumvent this, we rely on a set $T$ of data augmentations that preserve pitch (such as gain or additive white noise). We create augmented views $\tilde{x} = t(x)$ of our inputs $x$ by applying random transforms $t \sim T$.

Similarly to [35], we then train our model to be invariant to those transforms by minimizing the cross-entropy between $y = f_0(x)$ and $\tilde{y} = f_0(\tilde{x})$.

$$L_{\text{inv}}(y, \tilde{y}) = \text{CrossEntropy}(y, \tilde{y})$$

**Combining the losses.** For a given input sample $x$ and a given set of augmentations $T$,

- we first compute $x^{(k)}$ by pitch-shifting $x$ by a random number of bins $k$ (the precise procedure is described in section 3.2);
- we then generate two augmented views $\tilde{x} = t_1(x)$ and $\tilde{x}^{(k)} = t_2(x^{(k)})$, where $t_1, t_2 \sim T$;
- we compute $y = f_0(x)$, $\tilde{y} = f_0(\tilde{x})$ and $\tilde{y}^{(k)} = f_0(\tilde{x}^{(k)})$.

Our final objective loss is then:

$$L(y, \tilde{y}, \tilde{y}^{(k)}, k) = \lambda_{\text{inv}} L_{\text{inv}}(y, \tilde{y})$$

$$+ \lambda_{\text{equiv}} L_{\text{equiv}}(\tilde{y}, \tilde{y}^{(k)}, k)$$

$$+ \lambda_{\text{SCE}} L_{\text{SCE}}(\tilde{y}, \tilde{y}^{(k)}, k)$$

We illustrate this in Figure 2. To set the weights $\lambda$, we use the gradient-based method proposed by [48–50].
3.2 Audio-frontend
The inputs $x$ are the individual frames of the CQT. We have chosen the CQT as input since its logarithmic frequency scale, in which bins of the CQT exactly correspond to a fixed fraction $b$ of pitch semitones, naturally leads to pitch-shifting by translation. CQT is also a common choice made for pitch estimation [17, 19, 51].

To compute the CQT, we use the implementation provided in the nnAudio library [52] since it supports parallel GPU computation. We choose $f_{\text{min}} = 27.5$ Hz, which is the frequency of A0 the lowest key of the piano and select a resolution of $b = 3$ bins per semitone. Our CQT has in total $K = 996$ log-frequency bins, which corresponds to the maximal number of bins for a 16kHz signal.

3.3 Simulating translations.
To avoid any boundary effects, we perform pitch-shift by cropping shifted slices of the original CQT input frame as in [19] \(^5\). From a computational point of view, it is indeed significantly faster than applying classical pitch shift algorithms based on phase vocoder and resampling.

3.4 Transposition-preserving architecture
The architecture of $f_\theta$ is illustrated in Figure 3. It is inspired by [17]. Each input CQT frame is processed independently: first layer-normed [53] then preprocessed by two 1D-Conv (convolution in the log-frequency dimension) with skip-connections [54], followed by four 1D-Conv layers. As in [17], we apply a non-linear leaky-ReLU (slope 0.3) [55] and dropout (rate 0.2) [56] between each convolutional layer. Importantly, the kernel size and padding of each of these layers are chosen so that the frequency resolution remains unchanged. We found in practice that it helps the model to distinguish close but different pitches. The output is then flattened, fed to a final fully-connected layer and normalized by a softmax layer to become a probability distribution of the desired shape.

Note that all layers (convolutions \(^6\), elementwise non-linearities, layer-norm and softmax), except the last final fully-connected layer, preserve transpositions. To make the final fully-connected layer also transposition-equivariant, we propose to use \textbf{Toeplitz fully-connected layers}. It simply consists of a standard linear layer without bias but whose weights matrix $A$ is a Toeplitz matrix, i.e. each of its diagonals is constant.

\[ A = \begin{pmatrix} a_0 & a_{-1} & a_{-2} & \cdots & a_{-n+2} & a_{-n+1} \\ a_1 & a_0 & a_{-1} & \cdots & a_{-n+2} & \vdots \\ a_2 & a_1 & a_0 & \cdots & \vdots & \vdots \\ \vdots & \vdots & a_{m-1} & \cdots & \cdots & a_m \\ \vdots & \vdots & \vdots & \cdots & a_{m-n} & \vdots \\ a_n & a_{n-1} & a_{n-2} & \cdots & a_{n-n} & a_{n-n+1} \end{pmatrix} \]

Contrary to arbitrary fully-connected layers, Toeplitz matrices are transposition-preserving operations and only have $m + n - 1$ parameters instead of $mn$. Furthermore, they are mathematically equivalent to convolutions, making them straightforward to implement.

\(^5\) Specifically, we sample an integer $k$ uniformly from the range \([-k_{\text{max}}, \ldots, k_{\text{max}}]\), then generate two CQT outputs, denoted as $x$ and $x^{(k)}$, where $x$ is obtained by cropping the input CQT at indices $[k_{\text{max}}, K - k_{\text{max}} - 1]$, and $x^{(k)}$ is obtained by cropping the input CQT at indices $[k_{\text{max}} - k, K - k_{\text{max}} + k - 1]$, with $K$ the total number of bins of the original CQT frame and $k_{\text{max}} = 16$ in practice (see Figure 2).

\(^6\) Convolutions roughly preserve transpositions since the kernels are applied locally, meaning that if two transposed inputs are convolved by the same kernel, then the output results will be almost transpositions of each other as well.
3.5 Absolute pitch inference from y

Our encoder \( f_0 \) returns a probability distribution over (quantized) pitches. From an input CQT frame \( x \), we first compute the probability distribution \( f_0(x) \), then we infer the absolute pitch \( \hat{p} \) by applying the affine mapping:

\[
\hat{p}(x) = \frac{1}{b} \left( \arg \max_b f_0(x) + p_0 \right)
\]

where \( b = 3 \) is the number of bins per semitones in the CQT and \( p_0 \) is a fixed integer shift that only depends on \( f_0 \). As in [19], we set the integer shift \( p_0 \) by relying on a set of synthetic data\(^7\) with known pitch.

4. EXPERIMENTS

4.1 Datasets

To evaluate the performance of our approach, we consider the two following datasets:

1. **MIR-1K** [57] contains 1000 tracks (about two hours) of people singing Chinese pop songs, with separate vocal and background music tracks provided.

2. **MDB-stem-synth** [58] contains re-synthesized monophonic music played by various instruments.

The pitch range of the **MDB-stem-synth** dataset is wider than the one of **MIR-1K**. The two datasets have different sampling rates and granularity for the annotations.

We conduct separate model training and evaluation on both datasets to measure overfitting and generalization performance. In fact, given that our model is lightweight and does not require labelled data, overfitting performance is particularly relevant for real-world scenarios, as it is easy for someone to train on their own dataset, e.g. their own voice. However, we also examine generalization performance through cross-evaluation to ensure that the model truly captures the underlying concept of pitch and does not merely memorize the training data.

4.2 Training details

From an input CQT (see part 3.2), we first compute the pitch-shifted CQT (see part 3.3). Then two random data augmentations \( t_1, t_2 \sim T \) are applied with a probability of 0.7. We used white noise with a random standard deviation between 0.1 and 2, and gain with a random value picked uniformly between -6 and 3 dB. The overall architecture of \( f_0 \) (see part 3.4) is implemented in PyTorch [59].

For training, we use a batch size of 256 and the Adam optimizer [60] with a learning rate of \( 10^{-4} \) and default parameters. The model is trained for 50 epochs using a cosine annealing learning rate scheduler. Our architecture being extremely lightweight, training requires only 545MB of GPU memory and can be performed on a single GTX 1080Ti.

4.3 Performance metrics

We measure the performances using the following metrics.

1. **Raw Pitch Accuracy (RPA):** corresponds to the percentage of voiced frames whose pitch error\(^8\) is less than 0.5 semitone [61].

2. **Raw Chroma Accuracy (RCA):** same as RPA but considering the mapping to Chroma (hence allowing octave errors) [61].

RCA is only used in our ablation studies.

4.4 Results and discussions

4.4.1 Clean signals

We compare our results with three baselines: CREPE [16], SPICE [19] and DDSP-inv [45]. CREPE is fully-supervised while SPICE and DDSP-inv are two SSL approaches. To measure the influence of the training set, we train PESTO on the two datasets (**MIR-1K** and **MDB-stem-synth**) and also evaluate on the two. This allows to test model generalization.

We indicate the results in Table 1. We see that PESTO significantly outperforms the two SSL baselines (SPICE and DDSP-inv) even in the cross-dataset scenario (93.5% and 94.6%). Moreover, it is competitive with CREPE (-1.7% and -1.2%) which has 750 times more parameters and is trained in a supervised way on the same datasets.

<table>
<thead>
<tr>
<th>Model</th>
<th># params</th>
<th>Trained on</th>
<th>Raw Pitch Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>SPICE [19]</td>
<td>2.38M</td>
<td>private data</td>
<td>90.6% 91.1%</td>
</tr>
<tr>
<td>DDSP-inv [45]</td>
<td>-</td>
<td><strong>MIR-1K</strong> / <strong>MDB-stem-synth</strong></td>
<td>91.8% 88.5%</td>
</tr>
<tr>
<td>PESTO (ours)</td>
<td>28.9k</td>
<td><strong>MIR-1K</strong></td>
<td>96.1% 94.6%</td>
</tr>
<tr>
<td>PESTO (ours)</td>
<td>28.9k</td>
<td><strong>MDB-stem-synth</strong></td>
<td>93.5% 95.5%</td>
</tr>
<tr>
<td>CREPE [16]</td>
<td>22.2M</td>
<td>many (supervised)</td>
<td>97.8% 96.7%</td>
</tr>
</tbody>
</table>

Table 1. Evaluation results of PESTO compared to supervised and self-supervised baselines. CREPE has been trained in a supervised way on a huge dataset containing in particular **MIR-1K** and **MDB-stem-synth**. It is grayed out as a reference. For DDSP-inv, we report the results when training and evaluating on the same dataset.

4.4.2 Robustness to background music

Background noise and music can severely impact pitch estimation algorithms, making it imperative to develop robust methods that can handle real-world scenarios where background noise is often unavoidable.

We therefore test the robustness of PESTO to background music. For this, we use the **MIR-1K** dataset, which contains separated vocals and background tracks

\(^7\) synthetic harmonic signals with random amplitudes and pitch

\(^8\) i.e. distance between the predicted pitch and the actual one
Table 2. Robustness of PESTO and other baselines to background music with various Signal-to-Noise ratios. Adding background music to training samples significantly improves the robustness of PESTO (see section 4.4.2).

<table>
<thead>
<tr>
<th>Model</th>
<th>Raw Pitch Accuracy (MIR-1K)</th>
<th>PESTO</th>
<th>CREPE [16]</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>clean</td>
<td>20 dB</td>
<td>10 dB</td>
</tr>
<tr>
<td>SPICE [19]</td>
<td>91.4%</td>
<td>91.2%</td>
<td>90.0%</td>
</tr>
<tr>
<td>PESTO</td>
<td>β = 0</td>
<td>94.8%</td>
<td>90.7%</td>
</tr>
<tr>
<td></td>
<td>β = 1</td>
<td>94.5%</td>
<td>92.9%</td>
</tr>
<tr>
<td></td>
<td>β ~ U(0, 1)</td>
<td>94.7%</td>
<td>94.4%</td>
</tr>
<tr>
<td></td>
<td>β ~ N(0, 1)</td>
<td>94.8%</td>
<td>94.5%</td>
</tr>
<tr>
<td>CREPE [16]</td>
<td>97.8%</td>
<td>97.3%</td>
<td>95.3%</td>
</tr>
</tbody>
</table>

Table 3. Respective contribution of various design choices of PESTO for a model trained on MIR-1K.

5. CONCLUSION

In this paper, we presented a novel self-supervised learning method for pitch estimation that leverages equivariance to musical transpositions. We propose a class-based equivariant objective that enables Siamese networks to capture pitch information from pairs of transposed inputs accurately. We also introduce a Toeplitz fully-connected layer to the architecture of our model to facilitate the optimization of this objective. Our method is evaluated on two standard benchmarks, and the results show that it outperforms self-supervised baselines and is robust to background music and domain shift.

From a musical perspective, our lightweight model is well-suited for real-world scenarios, as it can run on resource-limited devices without sacrificing performance. Moreover, its SSL training procedure makes it convenient to fine-tune on a small unlabeled dataset, such as a specific voice or instrument. Additionally, the resolution of the model is a sixth of a tone but could eventually be increased by changing the resolution of the CQT. Moreover, despite modelling pitch estimation as a classification problem, we make no assumption about scale or temperament.

These features make our method still a viable solution, e.g. for instruments that use quartertones and/or for which no annotated dataset exists. We therefore believe that it has many applications even beyond the limitations of Western music.

Overall, the idea of using equivariance to solve a classification problem is a novel and promising approach that enables the direct return of a probability distribution over the classes with a single, potentially synthetic, labelled element. While our paper applies this approach to pitch estimation, there are other applications where this technique could be useful, such as tempo estimation.

Moreover, modelling a regression task as a classification problem can offer greater interpretability as the output of the network is not a single scalar but a whole probability distribution. Finally, it can generalize better to multi-label scenarios.

Our proposed method hence demonstrates the potential of using equivariance to solve problems that are beyond the scope of our current work. In particular, it paves the way towards self-supervised multi-pitch estimation.
6. ACKNOWLEDGEMENTS

This work has been funded by the ANR CIFRE convention n°2021/1537 and Sony France. This work was granted access to the HPC/AI resources of IDRIS under the allocation 2022-AD011013842 made by GENCI. We would like to thank the reviewers and meta-reviewer for their valuable and insightful comments.

7. REFERENCES


