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

Alain Riou[1,2] **Stefan Lattner**[2] **Gaëtan Hadjeres**[3] **Geoffroy Peeters**[1]
[1] LTCI, Télécom-Paris, Institut Polytechnique de Paris, France
[2] Sony Computer Science Laboratories - Paris, France
[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 unlabeled 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 [1] .

[1] https://github.com/SonyCSLParis/pesto

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 $f : \mathcal{X} \to \mathcal{Y}$ that is *invariant* to a set of transforms $\mathcal{T}_\mathcal{X}$, i.e. so that for any input $\mathbf{x} \in \mathcal{X}$ and transform $t \in \mathcal{T}_\mathcal{X}$

$$f(t(\mathbf{x})) \approx f(\mathbf{x}) \tag{1}$$

However, recent approaches [36–38] try instead to learn a mapping f that is *equivariant* to $\mathcal{T}_\mathcal{X}$, i.e. that satisfies

$$f(t(\mathbf{x})) \approx t'(f(\mathbf{x})) \tag{2}$$

where $t' \in \mathcal{T}_\mathcal{Y}$ with $\mathcal{T}_\mathcal{Y}$ a set of transforms that acts on the output space $\mathcal{Y}$. In other words, if the input is transformed, the output should be transformed accordingly. Representation collapse is hence prevented by design.

[2] The most common technique involves randomly applying data augmentations to inputs to create pairs of inputs that share semantic content.

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, DDSP-inv [45] is a DDSP-based approach [46] 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-

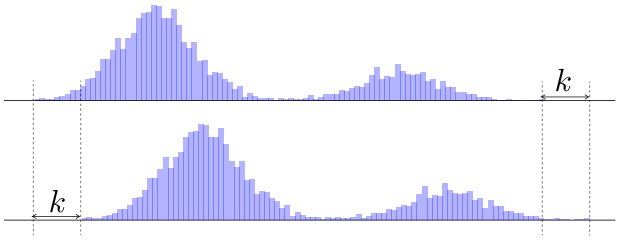

Figure 1. Example of k-transpositions. Visually, $\mathbf{y}$ and $\mathbf{y}'$ are just translated versions of each other. The sign of k and its absolute value respectively indicate the direction and the distance of the translation.

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

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

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

$$\begin{cases} y'_{i+k} = y_i & \text{when } \ 0 \leq i < d-k \\ y'_i = 0 & \text{when } \ i < k \\ y_i = 0 & \text{when } \ i \geq d-k-1 \end{cases} \tag{3}$$

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

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

Equivariance loss. We then design our criterion based on the following assumption: the probability of $\mathbf{x}$ to have pitch i is equal to the probability of $\mathbf{x}^{(k)}$ to have pitch $i+k$, i.e. y_i should be equal to $y^{(k)}_{i+k}$ [3]. In other words, if $\mathbf{x}^{(k)}$ is a pitch-shifted version of $\mathbf{x}$, their respective pitch probability distributions should be shifted accordingly, i.e. $\mathbf{y}^{(k)}$ should be the k-transposition of $\mathbf{y}$.

We take inspiration from [20] to design our equivariance loss. However, in our case, the output of our network f_θ 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:

$$\begin{array}{rccc} \phi & : & \mathbb{R}^d & \to & \mathbb{R} \\ & & \mathbf{y} & \mapsto & (\alpha, \alpha^2, \ldots, \alpha^d)\mathbf{y} \end{array} \tag{4}$$

where α is a fixed hyperparameter [4].

[3] For example, if $k = 2$ semitones, the probability of $\mathbf{x}$ to be C4 is exactly the probability of $\mathbf{x}^{(k)}$ to be a D4, and the same holds for any pitch independently of the actual pitch of $\mathbf{x}$.

[4] We found $\alpha = 2^{1/36}$ to work well in practice.

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

$$\mathcal{L}_{\text{equiv}}(\mathbf{y}, \mathbf{y}^{(k)}, k) = h_\tau \left(\frac{\phi(\mathbf{y}^{(k)})}{\phi(\mathbf{y})} - \alpha^k \right) \tag{5}$$

where h_τ is the Huber loss function [47], defined by

$$h_\tau(x) = \begin{cases} \frac{x^2}{2} & \text{if } |x| \leq \tau \\ \frac{\tau^2}{2} + \tau(|x| - \tau) & \text{otherwise} \end{cases} \tag{6}$$

Regularization loss. Note that if $\mathbf{y}^{(k)}$ is the k-transposition of $\mathbf{y}$ then $\mathcal{L}_{\text{equiv}}(\mathbf{y}, \mathbf{y}^{(k)}, k)$ 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 $\mathbf{y}$ and $\mathbf{y}^{(k)}$, i.e. the cross-entropy between the k-transposition of $\mathbf{y}$ and $\mathbf{y}^{(k)}$:

$$\mathcal{L}_{\text{SCE}}(\mathbf{y}, \mathbf{y}^{(k)}, k) = \sum_{i=0}^{d-1} y_i \log \left(y^{(k)}_{i+k} \right) \tag{7}$$

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

Invariance loss. $\mathcal{L}_{\text{equiv}}$ and $\mathcal{L}_{\text{SCE}}$ allow our model to learn relative transpositions between different inputs and learn to output probability distributions $\mathbf{y}$ and $\mathbf{y}^{(k)}$ 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 $\mathcal{T}$ of data augmentations that preserve pitch (such as gain or additive white noise). We create augmented views $\tilde{\mathbf{x}} = t(\mathbf{x})$ of our inputs $\mathbf{x}$ by applying random transforms $t \sim \mathcal{T}$.

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

$$\mathcal{L}_{\text{inv}}(\mathbf{y}, \tilde{\mathbf{y}}) = \text{CrossEntropy}(\mathbf{y}, \tilde{\mathbf{y}}) \tag{8}$$

Combining the losses. For a given input sample $\mathbf{x}$ and a given set of augmentations $\mathcal{T}$,

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

Our final objective loss is then:

$$\begin{aligned} \mathcal{L}(\mathbf{y}, \tilde{\mathbf{y}}, \tilde{\mathbf{y}}^{(k)}, k) = \lambda_{\text{inv}} \ & \mathcal{L}_{\text{inv}}(\mathbf{y}, \tilde{\mathbf{y}}) \\ + \lambda_{\text{equiv}} \ & \mathcal{L}_{\text{equiv}}(\tilde{\mathbf{y}}, \tilde{\mathbf{y}}^{(k)}, k) \\ + \lambda_{\text{SCE}} \ & \mathcal{L}_{\text{SCE}}(\tilde{\mathbf{y}}, \tilde{\mathbf{y}}^{(k)}, k) \end{aligned} \tag{9}$$

We illustrate this in Figure 2. To set the weights λ_* we use the gradient-based method proposed by [48–50].

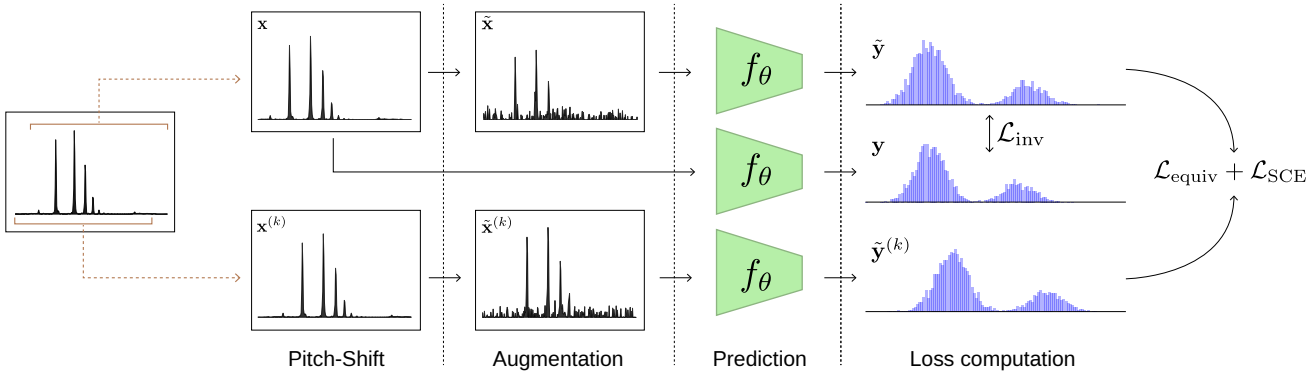

Figure 2. Overview of the PESTO method. The input CQT frame (log-frequencies) is first cropped to produce a pair of pitch-shifted inputs $(\mathbf{x}, \mathbf{x}^{(k)})$. Then we compute $\tilde{\mathbf{x}}$ and $\tilde{\mathbf{x}}^{(k)}$ by randomly applying pitch-preserving transforms to the pair. We finally pass $\mathbf{x}$, $\tilde{\mathbf{x}}$ and $\tilde{\mathbf{x}}^{(k)}$ through the network f_θ and optimize the loss between the predicted probability distributions.

3.2 Audio-frontend

The inputs $\mathbf{x}$ 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 $f_{\min} = 27.5$ 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 $K = 99b$ 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 Transpostion-preserving architecture

The architecture of f_θ 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 is never reduced. 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.

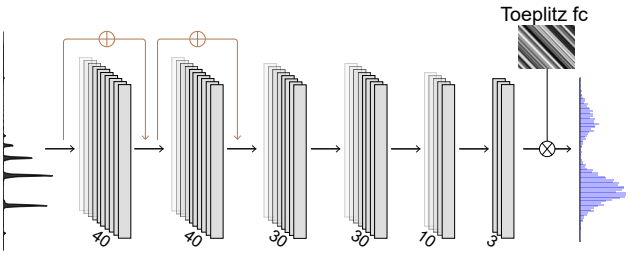

Figure 3. Architecture of our network f_θ. The number of channels varies between the intermediate layers, however the frequency resolution remains unchanged until the final Toeplitz fully-connected layer.

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 **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.

$$A = \begin{pmatrix} a_0 & a_{-1} & a_{-2} & \cdots & a_{-n+2} & a_{-n+1} \\ a_1 & a_0 & a_{-1} & \ddots & \ddots & a_{-n+2} \\ a_2 & a_1 & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ a_{m-1} & \cdots & \cdots & \cdots & \cdots & a_{m-n} \end{pmatrix} \tag{10}$$

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 $\{-k_{max}, \ldots, k_{\max}\}$, then generate two CQT outputs, denoted as x and $x^{(k)}$, where x is obtained by cropping the input CQT at indices $[k_{\max}, K - k_{\max} - 1]$, and $x^{(k)}$ is obtained by cropping the input CQT at indices $[k_{\max} - k, K - k_{\max} + k - 1]$, with K the total number of bins of the original CQT frame and $k_{\max}$ = 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

Model	# params	Trained on	Raw Pitch Accuracy *MIR-1K*	Raw Pitch Accuracy *MDB-stem-synth*
SPICE [19]	2.38M	private data	90.6%	89.1%
DDSP-inv [45]	-	*MIR-1K* / *MDB-stem-synth*	91.8%	88.5%
PESTO (ours)	28.9k	*MIR-1K*	**96.1%**	94.6%
PESTO (ours)	28.9k	*MDB-stem-synth*	93.5%	**95.5%**
CREPE [16]	22.2M	many (supervised)	**97.8%**	**96.7%**

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.

3.5 Absolute pitch inference from y

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

$$\hat{p}(\mathbf{x}) = \frac{1}{b}\left(\arg\max f_\theta(\mathbf{x}) + p_0\right) \tag{11}$$

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_θ. 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 $t_1, t_2 \sim \mathcal{T}$ 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_θ (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.

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

	Raw Pitch Accuracy (*MIR-1K*)			
Model	clean	20 dB	10 dB	0 dB
SPICE [19]	91.4%	91.2%	90.0%	81.6%
PESTO				
$\beta = 0$	**94.8%**	90.7%	79.2%	50.0%
$\beta = 1$	94.5%	94.2%	92.9%	**83.1%**
$\beta \sim \mathcal{U}(0, 1)$	94.7%	94.4%	92.9%	81.7%
$\beta \sim \mathcal{N}(0, 1)$	**94.8%**	**94.5%**	**93.0%**	82.6%
$\beta \sim \mathcal{N}(0, \frac{1}{2})$	**94.8%**	**94.5%**	92.9%	81.0%
CREPE [16]	**97.8%**	**97.3%**	**95.3%**	**84.8%**

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).

	MIR-1K		MDB	
	RPA	RCA	RPA	RCA
PESTO baseline	96.1%	96.4%	94.6%	95.0%
Loss ablations				
w/o $\mathcal{L}_{\text{equiv}}$	5.8%	8.6%	1.3%	6.1%
w/o $\mathcal{L}_{\text{inv}}$	96.1%	96.4%	92.5%	94.5%
w/o $\mathcal{L}_{\text{SCE}}$	96.1%	96.5%	86.9%	93.8%
Miscellaneous				
no augmentations	94.8%	95.4%	94.8%	95.2%
non-Toeplitz fc	5.7%	8.7%	1.2%	6.1%

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

and allows testing various signal-to-noise (here vocal-to-background) ratios (SNRs).

We indicate the results in Table 2. As foreseen, the performance of PESTO when trained on clean vocals (row $\beta = 0$) and applied to vocal-with-background considerably drop: from 94.8% (clean) to 50.0% (SNR = 0 dB) [9].

To improve the robustness to background music, we slightly modify our method to train our model on mixed sources. Instead of using gain and white noise as data augmentations, we create an augmented view of our original vocals signal $\mathbf{x}_{\text{vocals}}$ by mixing it (in the complex-CQT domain) with its corresponding background track $\mathbf{x}_{\text{background}}$:

$$\mathbf{x} = \mathbf{x}_{\text{vocals}} + \beta \mathbf{x}_{\text{background}} \tag{12}$$

Then, thanks to $\mathcal{L}_{\text{inv}}$, the model is trained to ignore the background music for making its predictions.

The background level β is randomly sampled for each CQT frame. The influence of the distribution we sample β from is depicted in Table 2. This method significantly limits the drop in performances observed previously and also makes PESTO outperform SPICE in noisy conditions.

4.4.3 Ablation study

Table 3 depicts the influence of our different design choices. First, we observe that the equivariance loss $\mathcal{L}_{\text{equiv}}$ and the final Toeplitz fully-connected layer (eq.(10)) are absolutely essential for our model not to collapse. Moreover, data augmentations seem to have a negligible influence on out-of-domain RPA (-0.2%) but slightly help when training and evaluating on the same dataset (+1.2%).

On the other hand, it appears that both $\mathcal{L}_{\text{inv}}$ and $\mathcal{L}_{\text{SCE}}$ do not improve in-domain performances but help the model to generalize better. This is especially true for $\mathcal{L}_{\text{SCE}}$, whose addition enables to improve RPA from 86.9% to 94.6% on *MDB-stem-synth*.

Finally, according to the drop of performances in RPA and RCA when removing $\mathcal{L}_{\text{inv}}$, it seems that the invariance loss prevents octave errors on the out-of-domain dataset.

[9] It should be noted that the difference between the 96.1% of Table 1 and the 94.8% of Table 2 is due to the fact that we do not apply any data augmentation (gain or additive white noise) when $\beta = 0$.

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 ANRT 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

[1] D. Talkin, "A robust algorithm for pitch tracking (RAPT)," pp. 495–518, 1995.

[2] P. Boersma, "Acurate short-term analysis of the fundamental frequency and the harmonics-to-noise ratio of a sampled sound," *IFA Proceedings 17*, vol. 17, pp. 97–110, 1993. [Online]. Available: http://www.fon.hum.uva.nl/paul/papers/Proceedings_1993.pdf

[3] M. Mauch and S. Dixon, "PYIN: A fundamental frequency estimator using probabilistic threshold distributions," in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, vol. 1, no. 1, 2014, pp. 659–663.

[4] A. Camacho and J. G. Harris, "A sawtooth waveform inspired pitch estimator for speech and music," *The Journal of the Acoustical Society of America*, vol. 124, no. 3, pp. 1638–1652, 2008.

[5] B. S. Lee and D. P. W. Ellis, "Noise robust pitch tracking by subband autocorrelation classification," in *Proc. Interspeech 2012*, 2012, pp. 707–710.

[6] K. Han and D. Wang, "Neural Network Based Pitch Tracking in Very Noisy Speech," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 22, no. 12, pp. 2158–2168, 2014.

[7] C. Hawthorne, E. Elsen, J. Song, A. Roberts, I. Simon, C. Raffel, J. Engel, S. Oore, and D. Eck, "Onsets and frames: Dual-objective piano transcription," in *Proceedings of the 19th International Society for Music Information Retrieval Conference, ISMIR 2018*. International Society for Music Information Retrieval, oct 2018, pp. 50–57. [Online]. Available: https://archives.ismir.net/ismir2018/paper/000019.pdf

[8] J. W. Kim and J. P. Bello, "Adversarial learning for improved onsets and frames music transcription," in *Proceedings of the 20th International Society for Music Information Retrieval Conference, ISMIR 2019*. International Society for Music Information Retrieval, jun 2019, pp. 670–677. [Online]. Available: https://archives.ismir.net/ismir2019/paper/000081.pdf

[9] Q. Kong, B. Li, X. Song, Y. Wan, and Y. Wang, "High-Resolution Piano Transcription with Pedals by Regressing Onset and Offset Times," *IEEE/ACM Transactions on Audio Speech and Language Processing*, vol. 29, pp. 3707–3717, oct 2021. [Online]. Available: https://arxiv.org/abs/2010.01815v3

[10] G. Song, Z. Wang, F. Han, and S. Ding, "Transfer learning for music genre classification," *IFIP Advances in Information and Communication Technology*, vol. 510, pp. 183–190, 2017.

[11] S. Oramas, F. Barbieri, O. Nieto Caballero, and X. Serra, "Multimodal deep learning for music genre classification," *Transactions of the International Society for Music Information Retrieval. 2018; 1 (1): 4-21.*, 2018.

[12] N. Ndou, R. Ajoodha, and A. Jadhav, "Music Genre Classification: A Review of Deep-Learning and Traditional Machine-Learning Approaches," in *2021 IEEE International IOT, Electronics and Mechatronics Conference (IEMTRONICS)*, 2021, pp. 1–6.

[13] V. Lostanlen and C. E. Cella, "Deep convolutional networks on the pitch spiral for music instrument recognition," in *Proceedings of the 17th International Society for Music Information Retrieval Conference, ISMIR 2016*. International Society for Music Information Retrieval, may 2016, pp. 612–618. [Online]. Available: https://archives.ismir.net/ismir2016/paper/000093.pdf

[14] Y. Han, J. Kim, and K. Lee, "Deep Convolutional Neural Networks for Predominant Instrument Recognition in Polyphonic Music," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 1, pp. 208–221, 2017.

[15] A. Solanki and S. Pandey, "Music instrument recognition using deep convolutional neural networks," *International Journal of Information Technology*, vol. 14, no. 3, pp. 1659–1668, 2022. [Online]. Available: https://doi.org/10.1007/s41870-019-00285-y

[16] J. W. Kim, J. Salamon, P. Li, and J. P. Bello, "Crepe: A Convolutional Representation for Pitch Estimation," in *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, vol. 2018-April, feb 2018, pp. 161–165. [Online]. Available: http://arxiv.org/abs/1802.06182

[17] C. Weiß and G. Peeters, "Deep-Learning Architectures for Multi-Pitch Estimation: Towards Reliable Evaluation," feb 2022. [Online]. Available: http://arxiv.org/abs/2202.09198

[18] R. Geirhos, J. H. Jacobsen, C. Michaelis, R. Zemel, W. Brendel, M. Bethge, and F. A. Wichmann, "Shortcut learning in deep neural networks," *Nature Machine Intelligence*, vol. 2, no. 11, pp. 665–673, apr 2020. [Online]. Available: https://www.nature.com/articles/s42256-020-00257-z

[19] B. Gfeller, C. Frank, D. Roblek, M. Sharifi, M. Tagliasacchi, and M. Velimirovic, "SPICE: Self-Supervised Pitch Estimation," *IEEE/ACM Transactions on Audio Speech and Language Processing*, vol. 28, pp. 1118–1128, oct 2020. [Online]. Available: https://dl.acm.org/doi/pdf/10.1109/TASLP.2020.2982285

[20] E. Quinton, "Equivariant Self-Supervision for Musical Tempo Estimation," *Proceedings of the 23rd International Society for Music Information Retrieval Conference, ISMIR 2022*, sep 2022. [Online]. Available: https://archives.ismir.net/ismir2022/paper/000009.pdf

[21] R. Hadsell, S. Chopra, and Y. LeCun, "Dimensionality reduction by learning an invariant mapping," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2, pp. 1735–1742, 2006.

[22] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *37th International Conference on Machine Learning, ICML 2020*, vol. PartF16814. International Machine Learning Society (IMLS), feb 2020, pp. 1575–1585. [Online]. Available: http://proceedings.mlr.press/v119/chen20j/chen20j.pdf

[23] J. Zbontar, L. Jing, I. Misra, Y. LeCun, and S. Deny, "Barlow Twins: Self-Supervised Learning via Redundancy Reduction," *38th International Conference on Machine Learning, ICML 2021*, mar 2021. [Online]. Available: http://proceedings.mlr.press/v139/zbontar21a/zbontar21a.pdf

[24] A. Bardes, J. Ponce, and Y. LeCun, "VICReg: Variance-Invariance-Covariance Regularization for Self-Supervised Learning," in *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022. [Online]. Available: https://openreview.net/forum?id=xm6YD62D1Ub

[25] T. Wang and P. Isola, "Understanding contrastive representation learning through alignment and uniformity on the hypersphere," in *37th International Conference on Machine Learning, ICML 2020*, vol. PartF16814. International Machine Learning Society (IMLS), may 2020, pp. 9871–9881. [Online]. Available: https://proceedings.mlr.press/v119/wang20k/wang20k.pdf

[26] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Momentum Contrast for Unsupervised Visual Representation Learning," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. IEEE Computer Society, nov 2020, pp. 9726–9735. [Online]. Available: https://openaccess.thecvf.com/content_CVPR_2020/papers/He_Momentum_Contrast_for_Unsupervised_Visual_Representation_Learning_CVPR_2020_paper.pdf

[27] X. Chen, H. Fan, R. Girshick, and K. He, "Improved Baselines with Momentum Contrastive Learning," mar 2020. [Online]. Available: http://arxiv.org/abs/2003.04297

[28] J. B. Grill, F. Strub, F. Altché, C. Tallec, P. H. Richemond, E. Buchatskaya, C. Doersch, B. A. Pires, Z. D. Guo, M. G. Azar, B. Piot, K. Kavukcuoglu, R. Munos, and M. Valko, "Bootstrap your own latent a new approach to self-supervised learning," in *Advances in Neural Information Processing Systems*, vol. 2020-Decem. Neural information processing systems foundation, jun 2020. [Online]. Available: https://papers.nips.cc/paper/2020/file/f3ada80d5c4ee70142b17b8192b2958e-Paper.pdf

[29] X. Chen and K. He, "Exploring simple Siamese representation learning," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. IEEE Computer Society, nov 2021, pp. 15 745–15 753. [Online]. Available: https://openaccess.thecvf.com/content/CVPR2021/papers/Chen_Exploring_Simple_Siamese_Representation_Learning_CVPR_2021_paper.pdf

[30] Y. Dubois, T. Hashimoto, S. Ermon, and P. Liang, "Improving Self-Supervised Learning by Characterizing Idealized Representations," sep 2022. [Online]. Available: https://openreview.net/pdf?id=agQGDz6gPOo

[31] A. Saeed, D. Grangier, and N. Zeghidour, "Contrastive learning of general-purpose audio representations," in *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, vol. 2021-June. Institute of Electrical and Electronics Engineers Inc., oct 2021, pp. 3875–3879. [Online]. Available: https://arxiv.org/abs/2010.10915v1

[32] J. Anton, H. Coppock, P. Shukla, and B. W. Schuller, "Audio Barlow Twins: Self-Supervised Audio Representation Learning," sep 2022. [Online]. Available: http://arxiv.org/abs/2209.14345

[33] D. Niizumi, D. Takeuchi, Y. Ohishi, N. Harada, and K. Kashino, "BYOL for Audio: Exploring Pre-Trained General-Purpose Audio Representations," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, pp. 1–15, apr 2022. [Online]. Available: https://dl.acm.org/doi/pdf/10.1109/TASLP.2022.3221007

[34] J. F. Gemmeke, D. P. Ellis, D. Freedman, A. Jansen, W. Lawrence, R. C. Moore, M. Plakal, and M. Ritter, "Audio Set: An ontology and human-labeled dataset for audio events," *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, pp. 776–780, 2017.

[35] J. Spijkervet and J. A. Burgoyne, "Contrastive Learning of Musical Representations," *Proceedings of the 22nd International Society for Music Information Retrieval Conference, ISMIR 2021*, mar 2021. [Online]. Available: https://archives.ismir.net/ismir2021/paper/000084.pdf

[36] R. Dangovski, L. Jing, C. Loh, S. Han, A. Srivastava, B. Cheung, P. Agrawal, and M. Soljačić, "Equivariant Contrastive Learning," *ICLR 2022 - 10th International Conference on Learning Representations*, oct 2022. [Online]. Available: https://openreview.net/pdf?id=gKLAAfiytI

[37] R. Winter, M. Bertolini, T. Le, F. Noé, and D.-A. Clevert, "Unsupervised Learning of Group Invariant and Equivariant Representations," feb 2022. [Online]. Available: https://openreview.net/pdf?id=47lpv23LDPr

[38] Q. Garrido, L. Najman, and Y. Lecun, "Self-supervised learning of Split Invariant Equivariant representations," feb 2023. [Online]. Available: https://openreview.net/pdf?id=2sIVxJ9Hp0

[39] A. M. Noll, "Cepstrum pitch determination." *The Journal of the Acoustical Society of America*, vol. 41 2, pp. 293–309, 1967.

[40] J. J. Dubnowski, R. W. Schafer, and L. R. Rabiner, "Real-Time Digital Hardware Pitch Detector," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 24, no. 1, pp. 2–8, 1976.

[41] M. J. Ross, H. L. Shaffer, A. Cohen, R. L. Freudberg, and H. Manley, "Average magnitude difference function pitch extractor," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 22, pp. 353–362, 1974.

[42] A. de Cheveigné and H. Kawahara, "YIN, a fundamental frequency estimator for speech and music," *The Journal of the Acoustical Society of America*, vol. 111, no. 4, pp. 1917–1930, 2002.

[43] Y.-J. Luo, K. W. Cheuk, T. Nakano, M. Goto, and D. Herremans, "Unsupervised disentanglement of pitch and timbre for isolated musical instrument sounds," in *International Society for Music Information Retrieval Conference*, 2020.

[44] K. Tanaka, Y. Bando, K. Yoshii, and S. Morishima, "Unsupervised disentanglement of timbral, pitch, and variation features from musical instrument sounds with random perturbation," in *2022 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, 2022, pp. 709–716.

[45] J. Engel, R. Swavely, A. Roberts, L. . Hanoi, . Hantrakul, and C. Hawthorne, "Self-Supervised Pitch Detection by Inverse Audio Synthesis," *Workshop on Self-Supervision in Audio and Speech at the 37th International Conference on Machine Learning (ICML 2020)*, pp. 1–9, 2020. [Online]. Available: https://goo.gl/magenta/ddsp-inv

[46] J. Engel, L. Hantrakul, C. Gu, and A. Roberts, "DDSP: Differentiable Digital Signal Processing," *The Eighth International Conference on Learning Representations, ICLR 2020*, jan 2020. [Online]. Available: https://openreview.net/pdf?id=B1x1ma4tDr

[47] P. J. Huber, "Robust Estimation of a Location Parameter," *The Annals of Mathematical Statistics*, vol. 35, no. 1, pp. 73–101, 1964. [Online]. Available: http://www.jstor.org/stable/2238020

[48] Z. Chen, V. Badrinarayanan, C. Y. Lee, and A. Rabinovich, "GradNorm: Gradient normalization for adaptive loss balancing in deep multitask networks," in *35th International Conference on Machine Learning, ICML 2018*, vol. 2. International Machine Learning Society (IMLS), nov 2018, pp. 1240–1251. [Online]. Available: http://proceedings.mlr.press/v80/chen18a/chen18a.pdf

[49] P. Esser, R. Rombach, and B. Ommer, "Taming transformers for high-resolution image synthesis," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. IEEE Computer Society, dec 2021, pp. 12 868–12 878. [Online]. Available: https://openaccess.thecvf.com/content/CVPR2021/papers/Esser_Taming_Transformers_for_High-Resolution_Image_Synthesis_CVPR_2021_paper.pdf

[50] J. MacGlashan, E. Archer, A. Devlic, T. Seno, C. Sherstan, P. R. Wurman, and P. Stone, "Value Function Decomposition for Iterative Design of Reinforcement Learning Agents," jun 2022. [Online]. Available: https://openreview.net/pdf?id=pNEisJqGuei

[51] R. M. Bittner, J. J. Bosch, D. Rubinstein, G. Meseguer-Brocal, and S. Ewert, "A Lightweight Instrument-Agnostic Model for Polyphonic Note Transcription and Multipitch Estimation," in *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, vol. 2022-May. Institute of Electrical and Electronics Engineers Inc., mar 2022, pp. 781–785. [Online]. Available: https://arxiv.org/abs/2203.09893v2

[52] K. W. Cheuk, H. Anderson, K. Agres, and D. Herremans, "nnAudio: An on-the-Fly GPU Audio to Spectrogram Conversion Toolbox Using 1D Convolutional Neural Networks," *IEEE Access*, vol. 8, pp. 161 981–162 003, 2020.

[53] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer Normalization," jul 2016. [Online]. Available: http://arxiv.org/abs/1607.06450

[54] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2016-Decem, dec 2016, pp. 770–778. [Online]. Available: https://www.cv-foundation.org/openaccess/content_cvpr_2016/papers/He_Deep_Residual_Learning_CVPR_2016_paper.pdf

[55] B. Xu, N. Wang, T. Chen, and M. Li, "Empirical Evaluation of Rectified Activations in Convolutional Network," may 2015. [Online]. Available: http://arxiv.org/abs/1505.00853

[56] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *Journal of*

Machine Learning Research, vol. 15, pp. 1929–1958, 2014.

[57] C.-L. Hsu and J.-S. R. Jang, "On the Improvement of Singing Voice Separation for Monaural Recordings Using the MIR-1K Dataset," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, no. 2, pp. 310–319, 2010.

[58] J. Salamon, R. M. Bittner, J. Bonada, J. J. Bosch, E. Gómez, and J. P. Bello, "An analysis/synthesis framework for automatic f0 annotation of multitrack datasets," *Proceedings of the 18th International Society for Music Information Retrieval Conference, ISMIR 2017*, pp. 71–78, 2017.

[59] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Köpf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "PyTorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems*, vol. 32. Neural information processing systems foundation, dec 2019. [Online]. Available: https://pytorch.org/

[60] D. P. Kingma and J. L. Ba, "Adam: A method for stochastic optimization," in *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, dec 2015. [Online]. Available: http://arxiv.org/abs/1412.6980

[61] G. E. Poliner, D. P. Ellis, A. F. Ehmann, E. Gómez, S. Streich, and B. Ong, "Melody transcription from music audio: Approaches and evaluation," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 15, no. 4, pp. 1247–1256, 2007.