the linearly-spaced and higher frequency resolution of the magnitude spectrum.

While the majority of the recent approaches try to learn this inverse transformation end-to-end, this is especially hard for a polyphonic music signal. To precisely reproduce a sustained note, an end-to-end model needs to learn: 1. different patterns for every combination of phase shift and period of a periodic waveform, and 2. how to activate them in the right sequence [6]. Accomplishing both tasks is challenging for speech and arguably even more so for music, which contains generally longer pitched sounds with wider pitch range, possibly multiple concurrent fundamental frequencies (polyphony), and whose absolute precision is essential.

Instead of reconstructing the signal in the time domain, we propose to use as output space an intermediate time-frequency representation consisting of three channels: the magnitude spectrum and the two components of the phase gradient: (M, ϕ'_i, ϕ'_m). The phase gradient is later integrated to estimate the phase spectrum ϕ, and finally audio is computed via the inverse STFT.

A model trained on our proposed output representation does *not* need to learn: 1. the shift variations of periodic waveforms as those are explicitly modeled by the inverse STFT, and 2. how to sequence phase, which is handled via the phase integration algorithm. Differently from the phase spectrum, the phase derivative along time is shift-invariant, thus it is a more suitable target for a shift-invariant architecture, such as the convolutional neural network.

The approaches that have been suggested in the recent years for neural audio synthesis in the time-domain have to use auto-regression to achieve horizontal phase coherence. For example, autoregression is at the core of models like WaveNet [3] and WaveRNN [4], however the fact that it is applied at the rate of audio samples make these model prohibitively expensive for the generation of high-resolution audio signals.

Audio domain shift-invariant convolutional neural vocoders can generate audio samples with much higher efficiency, but are not suited to reconstruct long pitched components precisely. This holds regardless of the training strategy, and includes for example generative adversarial networks (GAN) based models [1, 2] and diffusion based models [8, 9]. A recent neural vocoder for speech mel spectrogram inversion [5] adds an autoregressive loop that works on chunks of audio. The autoregressive nature of this architecture allows performing temporal integration, the operation needed to reconstruct stable sinusoidal components, while advancing by audio chunks rather than samples improves efficiency. However, the poor reconstruction quality observed when applying this model to music signals suggests that it is difficult to learn signal properties such as the rotation of phase from data with sufficient generalization.

In the neural audio synthesis literature, using instantaneous frequency has been considered explicitly in [6], where it is generated alongside the magnitude spectrum in order to reconstruct the audio of single notes, conditioned on the pitch contour and a timbre embedding.

2.4 Phase integration

Recovering the phase spectrum from the phase gradients requires an integration step. Theoretically, perfect integration should be possible under specific constraints, such as continuous phase gradient spectrum and window functions with infinite support. In practice, however there is no closed-form solution for integrating typical discrete phase gradient spectra [10].

Well-known phase gradient integration algorithms have been developed in the Time-Scale Modification (TSM) literature. The standard Phase-Vocoder (PV) algorithm propagates the phase derivative along the time dimension to modify the duration of sinusoidal components [11]. The PV is able to preserve horizontal phase coherence, but struggles with vertical phase coherence, leading to smeared transients. Improvements to the standard phase-vocoder algorithm [12–14] use the magnitude spectrum to identify sinusoidal and/or impulsive components, and can propagate phase in either direction (time or frequency) depending on the local properties of the signal.

The phase gradient integration algorithm that we develop (Sect. 3.2) is inspired by these recent variations, and leads to subjectively improved reconstruction quality, alongside increased computational efficiency. A formal evaluation of the integration algorithm is out of the scope of this manuscript and left as future work.

3. MODEL

In this section we describe our proposed model for mel spectrogram inversion. The model is composed of a time-wise convolutional neural network (Sect. 3.1) that estimates magnitude and phase gradient from the mel spectrogram, and a phase integration algorithm (Sect. 3.2) that estimates the phase spectrum given the phase gradient. The time-domain reconstructed audio is finally obtained via inverse STFT from the magnitude and phase spectra.

3.1 Network architecture

The neural network is a stack of 8 1-d (time) convolutional layers with 1536 hidden channels, a kernel size of 3 frames and ReLU activations (Fig. 3). The input and output have the same number of time frames, but different numbers of frequency bins, and their center frequencies are different, i.e. log-spaced for the mel spectrogram input and linearly spaced for the magnitude and phase gradient outputs. The frequency bins of the input and the magnitude channel of the output are independently standardized using the mean and standard deviation values computed from the training set. We found that a direct path from the input to the magnitude channel of the output leads to significant improvements in the reconstruction of magnitude and the training speed. This direct path consists only of a frequency warping operation from mel- to linear-scale.

The phase gradients (ϕ'_i, ϕ'_m) are computed using the Auger-Flandrin technique [15]. Although we have not ex-

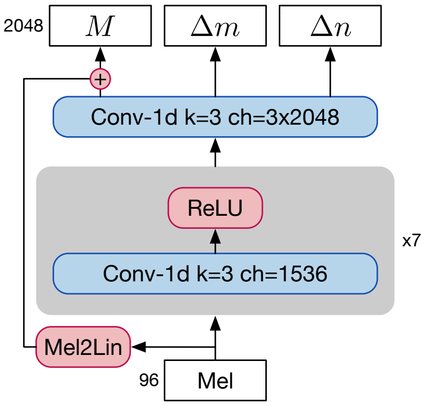

Figure 3: Convolutional network architecture

perimented with other ways to compute the phase gradient, it was argued [16] that using a less precise method, such as finite differences might perform just as well. Instead of using the phase gradients directly, we use the vertical and horizontal bin offsets (Eq. (6)), which are derived from the two components of the phase gradient. In order to remove outliers, the absolute bin offsets are clipped to 4.0 on the frequency dimension and to $N/2R$ on the time dimension.

The model uses linear output activation on all output channels except for the magnitude channel, where we apply a scaled tanh activation $f_\beta(x) = \beta \tanh(x/\beta)$ with $\beta = 5$ to use mostly the linear regime while also preventing overflow [6]. The three channels are then scaled and offset appropriately to match the statistics of the targets.

3.1.1 Losses

The magnitude channel is trained with a Mean Square Error (MSE) loss term, and a further MSE loss term computed on the first 20 linear-frequency cepstral coefficients (LFCC).

$$\begin{aligned}\mathcal{L}_1 &= (\hat{M} - M)^2 & (7)\\ \mathcal{L}_2 &= (\mathrm{DCT}_{:20}(\hat{M}) - \mathrm{DCT}_{:20}(M))^2, & (8)\end{aligned}$$

where $\hat{M}$ indicates the estimated magnitude spectrum, M the target magnitude spectrum, and DCT the normalized Discrete Cosine Transform; in these and the following loss formulas the $[m, n]$ indices and the global average operation have been omitted for simplicity. While $\mathcal{L}_1$ acts on point estimates, $\mathcal{L}_2$ pushes the spectral envelope towards its true value, leading to faster convergence and better reconstruction quality.

The phase gradient channels are trained with another MSE loss, weighted by the power spectrum of the target signal M^2. A matrix $\lambda \in [0, 1]$, computed from the phase gradient (Sect. 3.2), is used to distinguish sinusoidal and impulsive components. The idea is that the phase derivative along the time/frequency dimension contributes to the loss only for sinusoidal/impulsive components:

$$\mathcal{L}_3 = \begin{cases} M^2(\hat{\Delta m} - \Delta m)^2 & \lambda > 0.5 \\ M^2(\hat{\Delta n} - \Delta n)^2 & \lambda \leq 0.5, \end{cases} \tag{9}$$

where $(\hat{\Delta m}, \hat{\Delta n})$ are the estimated bin offsets.

Finally, because λ is a function of $\nabla \phi$ and is used for integration, we add a loss term:

$$\mathcal{L}_4 = M^2(\hat{\lambda} - \lambda)^2, \tag{10}$$

where $\hat{\lambda}$ is computed using the phase gradient estimates and λ using the target values. The final loss is a weighted sum of all the loss terms:

$$\mathcal{L} = \sum_l \alpha_l \mathcal{L}_l, \tag{11}$$

where all weights are set to 1 except $\alpha_2 = 0.1$ to balance the contribution of all terms during training.

Differently from time-domain methods for mel spectrogram inversion, we found reconstruction losses to yield satisfying results and did not add any adversarial loss. Evaluating the advantages of including adversarial losses is left for future research.

3.2 Phase integration

The algorithm we use for integrating phase from phase gradients relies on the classification of spectral bins into either sinusoidal, transient or noise components.

The classification uses the phase gradient and relies on the following rationale [7]: around sinusoidal/impulsive components the reassigned frequency/time is approximately constant along frequency/time

$$\lambda[m, n] = e^{-(\frac{d}{dm}\dot{m}[m,n]/\frac{d}{dn}\dot{n}[m,n])^2}, \tag{12}$$

where the derivatives d/dm and d/dn are computed with centered finite differences.

After computing λ, the phase gradients are propagated horizontally (Eq. (3)) if $\lambda > \lambda^{\mathrm{S}}$, vertically (Eq. (4)) if $\lambda < \lambda^{\mathrm{I}}$, and set to a random value otherwise. λ^{I} and λ^{S} are threshold values for impulsive and sinusoidal components, and are used to identify the spectral bins over which respectively vertical or horizontal phase coherence should be enforced. We empirically set $\lambda^{\mathrm{I}} = 0.4$ and $\lambda^{\mathrm{S}} = 0.5$, as these values performed well on early trials.

4. EXPERIMENTS

In this section we discuss how we evaluate the pitch stability of the proposed model, comparing to strong baseline vocoder models from the speech synthesis literature.

4.1 Experimental Setup

We compare the reconstruction of the proposed `phase-gradient` model against state-of-the-art approaches: `melgan` [1][2], `hifigan` [2][3], `cargan` [5][4], and `diffwave` [8][5]. All models have been trained on the same data, containing 13 hours of ambient music loops

[2] https://github.com/descriptinc/melgan-neurips
[3] https://github.com/kan-bayashi/ParallelWaveGAN.git
[4] https://github.com/descriptinc/cargan
[5] https://github.com/lmnt-com/diffwave

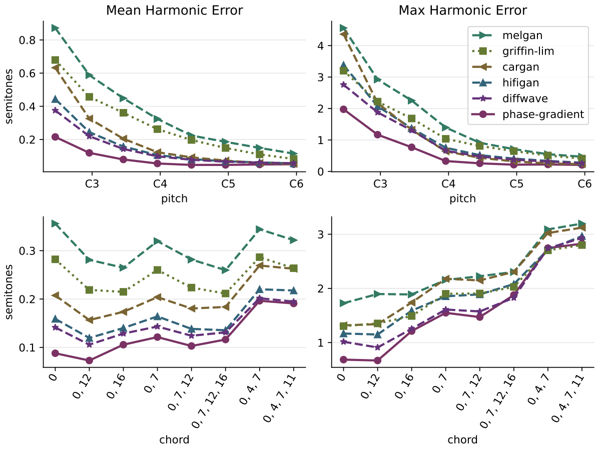

Figure 4: Harmonic error of different mel spectrogram inversion models for synthesized notes in the (C2, C7) range (top row). `phase-gradient` model achieves lower error than the baseline models on the entire range. The values have been smoothed with a moving average with size/stride equal to 12/6 semitones to filter out noise. The bottom row shows the error when adding more notes in different combinations, where the error is averaged over the entire pitch range, and the numbers on the x axis indicate the intervals in semitones that are played simultaneously, e.g. "0, 4, 7" is the major triad.

from commercial libraries[6], split into training, validation and test in the ratio of 80/10/10, which we refer to as the Ambient dataset. The audio from these loop libraries is converted to mono at 44.1kHz, 16-bit, the spectrograms are computed with a frame size of 2048 samples and hop size of 256 samples, and finally 96 bands are used for the mel spectrogram.

All neural networks were trained from scratch. The `phase-gradient` network was trained using 2 Volta GPUs in parallel and batches of 32 examples, with the Adam optimizer and learning rate set to $3e{-}5$. The training was stopped after 1024 epochs, when the validation loss converged, which took approximately 2 days.

As a further non machine-learned baseline, we consider a reconstruction algorithm `griffin-lim`, which generates audio from the mel spectrogram by first warping the frequency axis and the values to obtain a magnitude spectrogram, then applying the Griffin-Lim algorithm [17] for 500 iterations.

4.2 Pitch stability

To evaluate the pitch stability of our model, we used FluidSynth[7] to synthesize a dataset of one second long notes and chords in the (C2, C7) range, using a set of four different sounds: a rhodes piano, a church organ, a string ensemble and a nylon guitar.

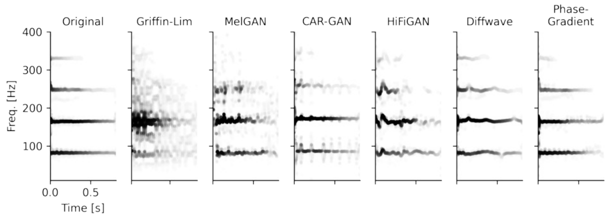

Figure 5: Reconstructions of an E2 nylon guitar note using different mel spectrogram inversion models. The figure shows the reassigned spectrograms [7] which highlight the instability on the fundamental and the harmonics caused by the lack of temporal phase coherence.

Measuring stability with a pitch tracker is only feasible for single notes, and we found that errors computed with a pitch tracker even for monophonic signals did not reflect our perception of reconstruction quality. A possible explanation is that pitch-trackers average out errors in the harmonic frequencies, which is inconvenient in this context because the precise location of the overtones is important for timbre perception [18].

For this reason, we define a harmonic error metric H_{err} as the sum of the frequency errors of the fundamental and the first 4 harmonic frequencies, expressed in the pitch scale:

$$H_{\text{err}}[p, h, n] = 12|\log_2(\frac{\hat{f}_{p,h}[n]}{f_{p,h}[n]})|, \tag{13}$$

where $f_{p,h}[n]$ and $\hat{f}_{p,h}[n]$ are the frequencies of the h-th harmonic of the p-th note, at frame n, for the original and the reconstructed audio respectively; the frequencies are estimated as the closest peaks to the nominal frequency of the partial, using quadratic interpolation, on a magnitude spectrum computed using frame/hop size equal to 4096/256 samples. We look at the mean and maximum values of the harmonic error, as a way to summarize the expected and worst case reconstruction errors.

Results show that our `phase-gradient` model was able to reconstruct the single notes with lower mean and maximum harmonic error over the entire range of notes, especially in the lower pitch range (Fig. 4(a)). A possible explanation for the larger improvement on the low register is that the long waveform period of lower-pitched notes makes them challenging to learn in the time-domain, given the high number of phase variations. A visualization of the different models' reconstruction of the same E2 guitar note is provided in Fig. 5.

We also synthesized different combinations of notes, to test the reconstruction quality on more challenging signals. The combinations include octaves, major 12ths, perfect fifths, an open- and a close-position voicing of the major triad, and a close-position voicing of the major seventh chord. As expected, the harmonic error increases when adding more notes (Fig. 4(b)), as they are harder to recognize in the input mel spectrogram. The `phase-gradient` model is able to yield lower mean and maximum harmonic error on all the considered combi-

[6] we used the following loop packs licensed from Big Fish Audio Ltd.: Ambient Piano, Ambient Skyline 3, Ambient Waves, Eclipse: Ambient Guitars, Ethereal Harp, Zen Ambient Vol. 2

[7] http://www.fluidsynth.org

	FAD (↓)			H_{err} (↓)		MOS (↑)	RTF (↑)	#Params
	Ambient	NSynth	N+C	Notes	Chords	NSynth		
griffin-lim [17]	10.59	6.16	6.79	0.28	0.24	1.42 ± 0.11	7.14	0
melgan [1]	2.07	2.80	2.84	0.36	0.30	1.58 ± 0.15	179.90	4M
cargan [5]	6.47	8.31	8.45	0.21	0.21	1.74 ± 0.12	4.36	25M
hifigan [2]	**0.85**	1.31	**1.81**	0.16	0.17	2.89 ± 0.12	68.12	13M
diffwave [8]	2.62	6.84	1.89	0.14	0.15	3.19 ± 0.12	0.32	7M
phase-gradient	1.26	**1.26**	1.86	**0.09**	**0.14**	**3.73** ± 0.13	3.58	28M
oracle	0.51	0.33	0.00	0.00	0.00	4.34 ± 0.15	—	—

Table 1: Reconstruction results

nations of notes. However, the decrease in the reconstruction quality, particularly on close-position chord voicings, suggests possible connections with the target's frequency resolution.

4.3 Reconstruction quality

To evaluate the overall reconstruction quality of the proposed model, we compute the Frechét Audio Distance (FAD) [19] on the Ambient dataset, the dataset of 1920 one second long notes and chords ("N+C") used in Sect. 4.2, and the NSynth dataset [20]. The results are shown in Table 1 alongside aggregated mean harmonic error results from the experiment discussed in Sect. 4.2.

The FAD metric compares embedding statistics generated on two potentially different sets of audio signals, i.e. evaluation and reference set. On the Ambient and NSynth datasets we compute the FAD between the reconstructed test split (evaluation) and the original training split (reference). On N+C, the reconstructed and original signals from the *entire* dataset are used as evaluation and reference sets. An ideal model (oracle) is added to provide reference values, useful when the evaluation and reference sets are different.

The aggregated mean harmonic error results are computed over two meaningful subsets of the N+C dataset: a "Notes" dataset, representing all single notes, and a "Chords" dataset, containing all note combinations with more than one pitch class (see Fig. 4 bottom row).

We conducted a small listening test to evaluate a set of notes reconstructed with different models. The set includes G2 and G3 notes randomly selected from NSynth, for each instrument family. The 5-scale Mean Opinion Score (MOS) values and the 95% confidence intervals are shown in Table 1.

The results show that the phase-gradient model is competitive with other state-of-the-art models, despite having simpler neural network and training procedure. The fact that hifigan model is able to score lower FAD than the proposed model on the Ambient and N+C datasets suggests it has higher reconstruction accuracy on different sonic characteristics.

Specifically, we noticed that hifigan was able to reconstruct impulsive components such as transients and percussive onsets with higher energy and often more accurately than the phase-gradient model, while struggling with the stability of pitched notes and chords. The diffwave model exhibited high frequency "hissing" noise, but was otherwise surprisingly stable on harmonic components. This quality likely stems from the wide ~7s receptive field, obtained using the entire reverse process at generation time instead of a fast sampling schedule [8]. As reported by its authors in [5], we confirm that the reconstructions made by the cargan model contained "boundary artifacts that appear as repeated clicks", compromising their usability. Finally, the reconstructions obtained with the melgan model were characterized by heavy phase artifacts such as metallic sounds, and very unstable pitch.

Generation time is also shown in Table 1 in terms of real-time factor (RTF), defined as the number of seconds of audio that can be generated per second, evaluated on a single NVidia Volta GPU. phase-gradient's generation is faster than real time, and close to cargan. While phase-gradient's neural network is as fast as hifigan and melgan, 99% of the generation time is spent during the auto-regressive phase integration stage, suggesting a clear direction for optimization.

5. CONCLUSIONS

In this work we have proposed a new mel spectrogram inversion model designed for music that achieves improved reconstruction of sustained notes and chords, compared to state-of-the-art models from the speech synthesis literature. This improvement is obtained using a frequency-domain target representation that is time shift invariant for harmonic signal components. The proposed model is able to reconstruct single notes and chords respectively 60% and 10% more precisely than existing models, when evaluated with a novel harmonic error metric, while still being competitive on generic loop reconstruction. Potential directions for improvement include using pitch-shift augmentation, investigating log-frequency target representations, and training a separate time-domain model to supply the percussive components.

6. ACKNOWLEDGEMENTS

We would like to thank the following colleagues for their valuable help and feedback: David Varas Gonzalez, Tim O'Brien, Avery Wang, Meghna Ranjit, and André Bergner.

7. REFERENCES

[1] K. Kumar, R. Kumar, T. de Boissiere, L. Gestin, W. Z. Teoh, J. Sotelo, A. de Brébisson, Y. Bengio, and A. C. Courville, "Melgan: Generative adversarial networks for conditional waveform synthesis," *Advances in neural information processing systems*, vol. 32, 2019.

[2] J. Kong, J. Kim, and J. Bae, "Hifi-gan: Generative adversarial networks for efficient and high fidelity speech synthesis," *Advances in Neural Information Processing Systems*, vol. 33, pp. 17 022–17 033, 2020.

[3] A. v. d. Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu, "Wavenet: A generative model for raw audio," in *ISCA Speech Synthesis Workshop (SSW)*, 2016.

[4] N. Kalchbrenner, E. Elsen, K. Simonyan, S. Noury, N. Casagrande, E. Lockhart, F. Stimberg, A. v. d. Oord, S. Dieleman, and K. Kavukcuoglu, "Efficient neural audio synthesis," in *International Conference on Machine Learning (ICML)*, 2018.

[5] M. Morrison, R. Kumar, K. Kumar, P. Seetharaman, A. Courville, and Y. Bengio, "Chunked autoregressive gan for conditional waveform synthesis," in *International Conference on Learning Representations (ICLR)*, 2022.

[6] J. Engel, K. K. Agrawal, S. Chen, I. Gulrajani, C. Donahue, and A. Roberts, "Gansynth: Adversarial neural audio synthesis," in *International Conference on Learning Representations (ICLR)*, 2019.

[7] K. R. Fitz and S. A. Fulop, "A unified theory of time-frequency reassignment," *arXiv preprint arXiv:0903.3080*, 2009.

[8] Z. Kong, W. Ping, J. Huang, K. Zhao, and B. Catanzaro, "Diffwave: A versatile diffusion model for audio synthesis," in *International Conference on Learning Representations (ICLR)*, 2021.

[9] N. Kandpal, O. Nieto, and Z. Jin, "Music enhancement via image translation and vocoding," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022, pp. 3124–3128.

[10] Z. Průša and P. L. Søndergaard, "Real-time spectrogram inversion using phase gradient heap integration," in *International Conference on Digital Audio Effects (DAFx)*, 2016, pp. 17–21.

[11] M. Dolson, "The phase vocoder: A tutorial," *Computer Music Journal*, vol. 10, no. 4, pp. 14–27, 1986.

[12] Z. Průša and N. Holighaus, "Phase vocoder done right," in *IEEE European Signal Processing Conference (EUSIPCO)*, 2017, pp. 976–980.

[13] J. Laroche and M. Dolson, "Improved phase vocoder time-scale modification of audio," *IEEE Transactions on Speech and Audio Processing*, vol. 7, no. 3, pp. 323–332, 1999.

[14] E.-P. Damskägg and V. Välimäki, "Audio time stretching using fuzzy classification of spectral bins," *Applied Sciences*, vol. 7, no. 12, p. 1293, 2017.

[15] F. Auger and P. Flandrin, "Improving the readability of time-frequency and time-scale representations by the reassignment method," *IEEE Transactions on signal processing*, vol. 43, no. 5, pp. 1068–1089, 1995.

[16] S. A. Fulop and K. Fitz, "Algorithms for computing the time-corrected instantaneous frequency (reassigned) spectrogram, with applications," *Journal of the Acoustical Society of America*, vol. 119, no. 1, pp. 360–371, 2006.

[17] N. Perraudin, P. Balazs, and P. L. Søndergaard, "A fast griffin-lim algorithm," in *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, 2013, pp. 1–4.

[18] H. Fletcher, E. D. Blackham, and R. A. Stratton, "Quality of piano tones," *Journal of the Acoustical Society of America*, vol. 34, pp. 749–761, 1962.

[19] K. Kilgour, M. Zuluaga, D. Roblek, and M. Sharifi, "Fréchet audio distance: A reference-free metric for evaluating music enhancement algorithms," in *ISCA Interspeech*, 2019, pp. 2350–2354.

[20] J. Engel, C. Resnick, A. Roberts, S. Dieleman, M. Norouzi, D. Eck, and K. Simonyan, "Neural audio synthesis of musical notes with wavenet autoencoders," in *International Conference on Machine Learning (ICML)*, 2017.

LATENT FEATURE AUGMENTATION FOR CHORUS DETECTION

Xingjian Du[1] **Huidong Liang**[1] **Yuan Wan**[1] **Yuheng Lin**[1] **Ke Chen**[2] **Bilei Zhu**[1] **Zejun Ma**[1]
[1] ByteDance AI Lab, Shanghai, China
[2] University of California San Diego, San Diego, United States
duxingjian.real@bytedance.com

ABSTRACT

In this paper, we introduce LA-Chorus, a chorus detection model based on **L**atent feature **A**ugmentation and ResNet-FPN architecture. We make three contributions. Firstly, we propose a method for implicitly augmenting chorus data in the latent space during the training stage. Compared to augmentations on audio surfaces such as time stretching and pitch shifting, latent augmentations indicate changes at a higher level in original audio, thereby increasing the diversity and sufficiency in training. Second, we apply Feature Pyramid Network (FPN) to generate additional embeddings from low dimension to high dimension, consequently achieving a multi-scale training paradigm. Lastly, we release Di-Chorus, a new diversified dataset of 13 genres and 14 languages for the community of music structure analysis. In conjunction with other public datasets, we conduct comprehensive experiments to evaluate the performance of our proposed method compared to other state-of-the-art models, where LA-Chorus outperforms other SOTAs by a considerable margin, meanwhile the proposed latent audio augmentation shows dominant advantages over traditional augmentation methods.

1. INTRODUCTION

Chorus detection, aiming for identifying the most “catchy” or “memorable” part of a song, is one of the fundamental tasks in music structure analysis (MSA) [1]. Chorus detection essentially helps better understand music compositions with computational modelling methods and has various applications, such as automatic chorus preview functions in music software that allow users to efficiently select songs from a large library according to their preference [2].

Currently, chorus detection models are based on deep neural network (DNN) architectures with a supervised MSA method, where annotations of different segments are used as target variables during the training stage [3–5]. The common approach is to regard chorus detection as a binary classification task, where each frame (or several frames) is assigned with a class label according to the corresponding segment annotation, and the model is trained to classify these labels [6].

To better perform this classification task, we identify two critical questions: (1) how to locate chorus with high precision as the resolution of feature maps decreases when model goes deeper, and (2) how to learn sufficient variations of chorus characteristics. The first issue requires incorporating chorus positional information into the latent representations learned by models, resembling the task of object detection in computer vision that aims to find the boundaries of target objects [7]. Nevertheless, as the network goes deeper, latent representations begin to lose positional meanings because of the reduced-sized feature maps (i.e. the resolution decreases) [8]. To address the problem of shrinking resolution in feature maps, previous chorus detection methods first used neural network architectures as backbones to generate audio embeddings, and then applied positional modifications on the networks. For example, [3] introduced a multi-task model that jointly detects chorus segments and their boundaries using convolutional neural network (CNN) to increase positioning precision, and [5] proposed a multi-scale CNN model that up-samples/down-samples the original audio features to better capture both global and local information. In this paper, we incorporate Feature Pyramid Networks (FPN) [8], a popular framework for object detection from computer vision, into a standard ResNet [9] as our model's backbone. It is designed specifically to tackle the problem of feature maps' decreasing resolutions by appending a network of reversed size order for each feature map with lateral connection, which will be discussed in Section 3.

The second issue, as learning sufficient chorus characteristics, can be addressed by using a diversified training dataset to feed the model such that it will generalize well at testing time. Nevertheless, despite the promising progress of emerging music annotations such as Isophonics [10], SALAMI [11], and Harmonics [12], the scarcity of labeled data (as they are costly to retrieve) and the deficiency of data diversity have always been challenging for music information retrieval (MIR) and other machine learning fields. To combat this problem, some traditional augmentation techniques have been proposed on the original inputs, such as rotating or flipping the images in computer vision [13], or time stretching [14] and pitch shifting [15] in audio signal processing. Recently, implicit augmentation methods, which focus on the augmentation in latent space, have shown remarkable performance over the pre-

Attribution: X. Du, H. Liang, Y. Wan, Y. Lin, K. Chen, B. Zhu, Z. Ma, “Latent Feature Augmentation for Chorus Detection”, in *Proc. of the 23rd Int. Society for Music Information Retrieval Conf.,* Bengaluru, India, 2022.

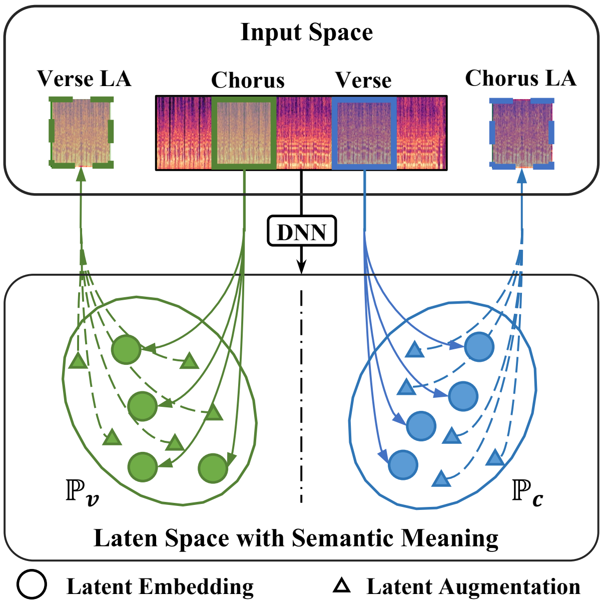

Figure 1. Illustration for implicit audio augmentation in MSA with two annotation types (verse and chorus) from a constant Q transformation (CQT) spectrogram excerpt of "Smooth Criminal" in *Isophonics* [10]. The spectrogram is first encoded into latent embeddings by frame, where chorus embedding and verse embedding follow latent distributions $\mathbb{P}_c$ and $\mathbb{P}_v$ respectively. Then latent augmentations are sampled around embeddings of verse/chorus segments, which correspond to augmented verse/chorus segments in the input space (represented by dashed lines, meaning they are NOT shown explicitly in the input space).

vious "shallow" methods in computer vision [16, 17]. The motivation is that latent representations may carry semantic meanings of original images (e.g. human age, gender, facial expressions) [18]. Such discoveries are also consistent with some works in audio signal processing, where latent audio representations have been found to capture distinctive audio features [19]. For example, [20] investigated the latent spaces of timbre and pitch of various instrument sounds encoded by a GM-VAE model; [21] introduced a Cycle-GAN based model for musical timbre transfer; and [22] disentangled pitch and rhythm representations to produce music analogies. According to these previous works, the latent features of audio correspond to semantic meaning of music and acoustic, such as timbre, rhythm pattern, etc. Therefore, augmentations in the latent space would correspond to changes of semantic features in the input audio segments, which leads to more variations than that of the augmentations in the shallow space. To our best knowledge, there is currently no application of latent augmentations in audio signal processing. In this paper, we illustrate this intuition by an MSA example in Figure 1.

Thus, we propose LA-Chorus, a supervised chorus detection model based on ResNet-FPN architecture that leverages implicit audio augmentations on latent features, which can better locate chorus positions and meanwhile enrich variations in training samples with semantic meanings. Moreover, since songs for most of the public datasets are not easy to retrieve, we further release a diversified collection of songs on YouTube for our MSA community, namely Di-Chorus, which contains 237 songs from 13 genres in 14 languages with annotations by experts. The rest of the paper is structured as follows: the next section introduces related works in chorus detection and latent augmentations, after which we discuss model structure and inference method. The experiment section presents LA-Chorus's performance on public datasets as well as Di-Chorus compared against other state-of-the-art models, followed by an ablation study showing the effectiveness of latent augmentations. Finally, the last section concludes our findings and contributions.

2. RELATED WORK

2.1 Chorus Detection

The origin of chorus detection tightly relates to thumbnailing, which aims to find a short preview (thumbnail) as a meaningful representation of a song [23]. Common approaches for thumbnailing includes evaluating the repeated sections of the audio waveform based on chroma transformation [24], selecting segments with the most repetition [25], and detecting significant change points with self-similarity matrix [26]. On the other hand, MSA assumes that songs contain different types of segments (e.g. chorus, verse, bridge, etc.) with certain structures [27]. Based on the assumption, many chorus detection algorithms in MSA took an unsupervised fashion in the early stage: [28] used heuristics to predict segment labels based on a restricted template for song structures; [29, 30] both applied Hidden Markov Model to derive different song sections; and [31] performed spectral clustering on the co-occurrence matrix generated from k-nearest neighbors.

With the advancement of deep learning in computer vision and natural language processing, DNN gradually makes its presence in MIR, among which ResNet [9], a CNN-based model with residual connections, becomes one of the most popular DNN architectures in recent MIR literature [4, 32]. At the same time, the emergence of labeled databases such as SALAMI [11] and Harmonix [12] make supervised learning gradually attractive for chorus detection, leading to the current paradigm of supervised chorus detection based on deep learning: [33] introduced a hybrid generative model with LSTM to directly predict segment labels; [3] proposed a multi-task method that jointly detects chorus segments and their boundaries; and [5] further proposed a multi-scale network with self-attention convolution to extract latent features of song segments, generating the current state-of-the-art results for chorus detection. In LA-Chorus, we will incorporate Feature Pyramid Network (FPN) [8], a top-down network that dedicates to the object positioning task, into ResNet as our model's backbone. The proposed framework is able to generate latent features with rich positional information and semantic meanings for later augmentation process, which will be discussed in detail in Section 3.