



**HAL**  
open science

# PITCHFLOWER: A FLOW-BASED NEURAL AUDIO CODEC WITH PITCH CONTROLLABILITY

Diego Torres, Axel Roebel, Nicolas Obin

► **To cite this version:**

Diego Torres, Axel Roebel, Nicolas Obin. PITCHFLOWER: A FLOW-BASED NEURAL AUDIO CODEC WITH PITCH CONTROLLABILITY. 2025. <hal-05409018>

**HAL Id: hal-05409018**

**<https://hal.science/hal-05409018v1>**

Preprint submitted on 10 Dec 2025

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



HAL Authorization

# PITCHFLOWER: A FLOW-BASED NEURAL AUDIO CODEC WITH PITCH CONTROLLABILITY

Diego Torres    Axel Roebel    Nicolas Obin

Sciences and Technologies of Music and Sound  
IRCAM, CNRS, Sorbonne Université – Paris, France

## ABSTRACT

We present PitchFlower, a flow-based neural audio codec with explicit pitch controllability. Our approach enforces disentanglement through a simple perturbation: during training, F0 contours are flattened and randomly shifted, while the true F0 is provided as conditioning. A vector-quantization bottleneck prevents pitch recovery, and a flow-based decoder generates high quality audio. Experiments show that PitchFlower achieves more accurate pitch control than WORLD at much higher audio quality, and outperforms SiFiGAN in controllability while maintaining comparable quality. Beyond pitch, this framework provides a simple and extensible path toward disentangling other speech attributes. We release the code at <https://github.com/diegotg2000/PitchFlower/>.

**Index Terms**— neural audio codec, disentanglement, pitch control, flow-matching, speech synthesis

## 1. INTRODUCTION

Speech attribute manipulation has long been an active area of research, with the goal of enabling modifications of interpretable features such as emotion, accent, speaker identity, or pitch. Pitch control, in particular, allows speech and music to be altered in meaningful ways. Its most familiar application is vocal tuning, where the singer’s pitch must match the intended note. In speech, changing intonation (of which pitch is a fundamental part) directly affects meaning and communicative intent. Traditional approaches to pitch control are rooted in the source-filter model of voice production. For example, the vocoder WORLD [1] decomposes speech into F0, spectral envelope, and aperiodicity ratio. Pitch can then be modified by adjusting the extracted F0 and resynthesizing the signal. However, such manipulations ignore the interactions between F0 and the other speech attributes, often resulting in unnatural-sounding audio with noticeable artifacts. More recent methods leverage deep learning [2–8] to achieve pitch control. Some, such as SiFiGAN [3], still embed assumptions from the source-filter model. Others apply them more indirectly, working with WORLD-derived representations as in PeriodGrad [4]. Alternative strategies include conditioning on explicit, interpretable attributes (e.g., FastVGAN [6]) or combining learned and explicit representations, as in Promonet [8] and KaraTuner [7]. These examples highlight how the choice of representation is tightly coupled with the feasibility of pitch control.

In parallel, neural audio codecs (NACs) have emerged as a powerful representation of speech. By leveraging neural networks and large-scale data, NACs achieve higher fidelity at lower bitrates than traditional codecs [9–11]. Crucially, they have also become a backbone for generative tasks such as text-to-speech, positioning NACs as a standard representation for modern speech and audio systems.

This has naturally raised the question of disentanglement within NACs. Most prior work has focused on separating linguistic content from speaker identity, with SpeechTokenizer [12] being a notable example: it leverages self-supervised models to impose a dedicated semantic level within the codec. Pitch disentanglement, however, has received far less attention. FACodec [13] considers four factors simultaneously (content, speaker, prosody, and acoustic details) but its high-level prosody codes do not allow for precise F0 control. PeriodCodec [14], designed for singing voice, is to our knowledge the first NAC with explicit pitch control. However, it inherits training instabilities of other GAN-based codecs and introduces additional losses and hyperparameters.

We introduce *PitchFlower*, a neural audio codec with precise pitch controllability. PitchFlower achieves disentanglement through a simple perturbation-based strategy: we perturb pitch information at the input and task the model to reconstruct the original signal, conditioned on the ground-truth F0.

The main contributions of this work are:

- We propose PitchFlower, the first flow-based neural audio codec with explicit pitch controllability.
- We introduce a perturbation+bottleneck methodology that enforces disentanglement while keeping the model simple to train. This allows PitchFlower to be trained with a single generative loss.
- We provide a systematic comparison of disentanglement strategies (bottleneck, adversarial, semantic distillation), analyzing their trade-offs in terms of controllability, audio quality, and information preservation.
- We demonstrate that PitchFlower achieves stronger controllability than DSP-based baselines and competitive performance with state-of-the-art neural approaches.

## 2. PITCHFLOWER

### 2.1. Architecture

PitchFlower adopts the standard architecture of recent flow-based audio codecs [15, 16], consisting of an autoencoder, a vector-quantization bottleneck, and a flow decoder (Figure 1). The F0 contour is provided as an explicit conditioning signal to the flow decoder. The objective is to disentangle pitch such that latent representations are free of F0 information, allowing pitch to be directly controlled by modifying the conditioning contour.

To enforce disentanglement, we perturb the input during training by flattening its F0 contour. Specifically, each frame’s F0 is replaced with the utterance-level mean plus a random shift sampled

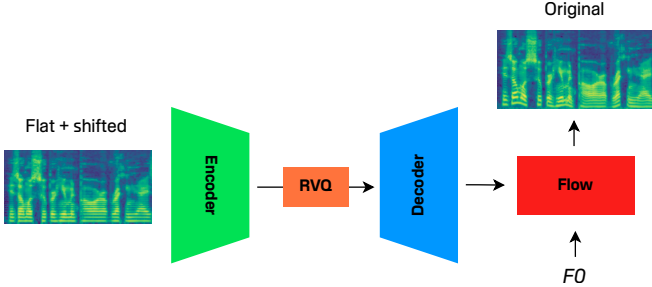


Fig. 1. Architecture and training methodology of PitchFlower.

from  $\mathcal{U}(-\Delta, \Delta)$ . WORLD is used to extract and modify the F0 values for this transformation.

A key element of PitchFlower is the flow decoder. Since perturbation and quantization inevitably remove information beyond pitch, the flow decoder compensates by sampling plausible values from the learned distribution, ensuring realistic and high-quality audio. Formally, given a perturbed mel-spectrogram with autoencoder output  $e$ , a target mel-spectrogram  $x_1$ , a noise sample  $x_0 \sim \mathcal{N}(0, 1)$ , and the flow decoder  $v_\theta$ . The conditional flow-matching loss is then [17]:

$$\mathcal{L}_{\text{CFM}} = \mathbb{E} \left\| v_\theta(x_t, e, f_0, t) - \left( x_1 - (1 - \sigma_{\min})x_0 \right) \right\|^2. \quad (1)$$

Unlike prior work [15, 16], where the flow acts as a post-net refining autoencoder outputs, here everything is trained end-to-end with the flow-matching loss as the only generative loss.

The final loss is then

$$\mathcal{L} = \mathcal{L}_{\text{CFM}} + \lambda_{\text{commit}} \mathcal{L}_{\text{commit}}. \quad (2)$$

## 2.2. Implementation details

We train and evaluate our model on the LibriTTS dataset [18]. The entire dataset is preprocessed offline, as opposed to an on-the-fly approach. The pitch perturbation range is set to  $\Delta = 5$  semitones. The encoder and decoder are built from two ConvNeXt blocks [19] (6 layers each), followed by self-attention [20]. The second block uses a stride of 2 for down/up-sampling. The RVQ module contains 8 codebooks with 512 entries each (dimension 256). The flow decoder consists of 4 blocks with 8 layers each. Hidden dimensions are 512 for the autoencoder and 256 for the flow module. The F0 contour is encoded with a 3-layer MLP (64 units per layer), with separate embeddings for unvoiced frames and missing F0. For training, we adopt conditional flow matching [17] with  $\sigma_{\min} = 10^{-4}$ . Classifier-free guidance [21] is applied by dropping F0 conditioning 10% of the time. At inference, we use 10 flow steps and a classifier-free guidance scale of 3.0. The model is trained for 800k iterations on a single RTX 4070 GPU, using AdamW (lr= $10^{-4}$ ), batch size 32, and 1.5-second audio segments. The commitment loss weight is set to 0.25. Vocos [22] is used as vocoder to produce waveforms from mel-spectrograms.

## 3. EVALUATION

We design our evaluation methodology around the basic requirements for a pitch-controllable model: accurate modification of the F0 with high audio quality. In addition, the model should preserve the linguistic content of the utterance and minimize changes to the

speaker’s voice quality. To capture these aspects, we rely on four objective metrics, evaluated on the dev-clean subset of LibriTTS with pitch shifts ranging from  $-6$  to  $+6$  semitones.

(1) *Word error rate*. We use an ASR model to transcribe the transposed audio and compare it to the original transcription. Specifically, we use the English-only medium-sized<sup>1</sup> version of Whisper [23].

(2) *Speaker similarity*. We measure similarity between the original and transposed audio using speaker embeddings extracted with ECAPA-TDNN [24]<sup>2</sup>, compared via cosine similarity.

(3) *FORMSE*. We compute F0 contours for both the original and transposed audio. The original contour is shifted by the target amount, and the root mean square error (Hz) is computed against the transposed contour. F0 estimation is performed with CREPE [25]<sup>3</sup>.

(4) *UTMOS*. We use UTMOS [26]<sup>4</sup> to automatically estimate the perceptual quality of the transposed audio.

## 4. DISENTANGLEMENT STRATEGIES

We investigate several strategies for disentanglement within the AutoFlower framework (autoencoder + flow). Unlike PitchFlower, these variants do not apply F0 masking at the input, and instead rely on alternative mechanisms to separate pitch information.

(0) *Bottleneck-based disentanglement*. Even without explicit masking, the RVQ module acts as an information bottleneck, encouraging the model to exclude F0 from the codes and rely on the conditioning of signal instead. This effect alone provides a non-trivial degree of controllability, following the principle first exploited in AutoVC [27].

(1) *Adversarial disentanglement*. Inspired by PeriodCodec [14] and NaturalSpeech3 [13], we add a pitch predictor with a gradient reversal layer. The predictor takes the quantized bottleneck as input and classifies log-F0 into one of six bins: five bins spanning 32–1024 Hz (log-scale) plus one for unvoiced frames. The gradient reversal layer encourages the encoder to remove pitch information from the latent variables.

(2) *Semantic distillation*. Following the line of SpeechTokenizer [12], we apply a distillation loss from HuBERT [28] onto the autoencoder representations. Unlike prior work that constrains only the first quantization level, we distill the full RVQ output. The loss is computed as cosine similarity between HuBERT features and the corresponding RVQ states.

Combining these components yields six variants for comparison: (i) bottleneck-only, (ii) bottleneck + HuBERT distillation, (iii) adversarial, (iv) adversarial + HuBERT, (v) PitchFlower, and (vi) PitchFlower + HuBERT.

### 4.1. Results

Results are summarized in Figure 2. They can be analyzed along two dimensions: the effect of semantic distillation with HuBERT, and the comparison between disentanglement strategies.

*Effect of HuBERT*. Adding HuBERT generally degrades intelligibility and speaker similarity. For both PitchFlower and the bottleneck baseline, WER increased when the distillation loss was applied, while the adversarial variant showed mixed results. Speaker

<sup>1</sup><https://huggingface.co/openai/whisper-medium>

<sup>2</sup><https://huggingface.co/speechbrain/spkrec-ecapa-voxceleb>

<sup>3</sup><https://github.com/maxrmorrison/torchcrepe>

<sup>4</sup><https://github.com/sarulab-speech/UTMOS22>

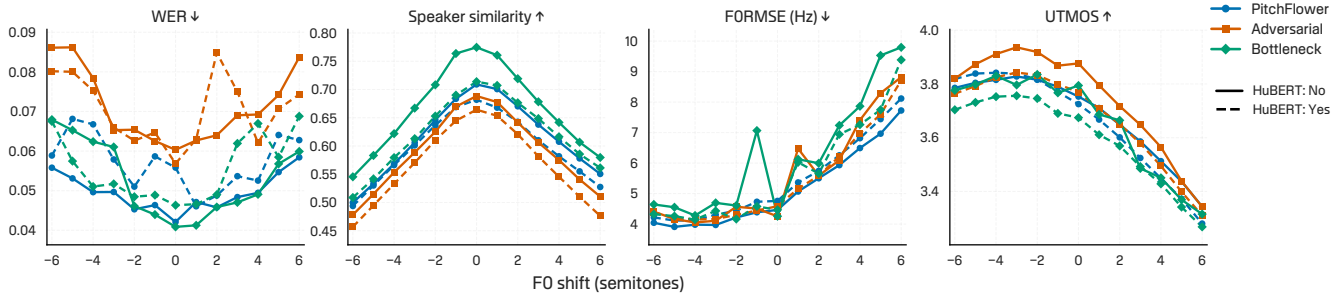


Fig. 2. Objective comparison of different disentanglement strategies.

similarity and UTMOS scores consistently dropped across all methods. The only clear benefit of HuBERT was improved F0 controllability, as seen in lower FORMSE for the bottleneck and adversarial methods. This suggests that HuBERT encourages more pitch-independent representations, but at the cost of other aspects of signal quality. The exception is PitchFlower, where HuBERT was detrimental across metrics.

*Comparison of methods.* Among the three base strategies, the adversarial method achieved the worst WER and speaker similarity. This effect likely arises because the encoder, when forced to hide pitch information from the predictor, may also suppress other speaker-related cues, leading to degraded intelligibility and voice preservation. Interestingly, it yielded the highest UTMOS scores. PitchFlower and the bottleneck baseline were more balanced, showing comparable intelligibility and audio quality. However, PitchFlower exhibited lower speaker similarity, which we attribute to distortions introduced by WORLD during pitch flattening.

*Overall ranking.* In terms of disentanglement, PitchFlower provided the most accurate F0 control, followed by the adversarial method, with the bottleneck baseline last. Adding HuBERT improved the baseline to match the adversarial method, and combining adversarial training with HuBERT further stabilized performance by reducing large peaks in the FORMSE curve.

## 5. COMPARISON WITH RELATED WORKS

We compare PitchFlower against two established methods: WORLD [1], a DSP source-filter vocoder, and SiFiGAN [3], a neural vocoder with pitch controllability. In addition, we evaluate a variant of our model, PitchFlowerUV, where the pitch-masking transformation follows DiffPitcher [5]: instead of the flat+shift operation, all F0 values are replaced with unvoiced frames before being resynthesized with WORLD.

Figure 3 shows that PitchFlower achieves the best pitch controllability, with consistently lower FORMSE across transpositions. This indicates that the flat+shift transformation is more effective at suppressing pitch than unvoicing. In terms of audio quality, the three neural methods perform similarly, while WORLD lags behind. Intelligibility is comparable overall: PitchFlower improves over WORLD and PitchFlowerUV but shows slightly higher WER than SiFiGAN, suggesting that unvoicing interferes more with linguistic content than flat+shift. For speaker similarity, SiFiGAN clearly outperforms all other methods, while both PitchFlower variants show nearly identical curves, reflecting residual artifacts inherited from WORLD. Subjective evaluations reveal a significantly higher audio quality of the deep learning methods compared to WORLD. How-

ever, no significant difference is found between the two versions of PitchFlower and SiFiGAN, as shown in Table 1.

Overall, PitchFlower balances quality and controllability better than existing methods, with the main trade-off being speaker similarity

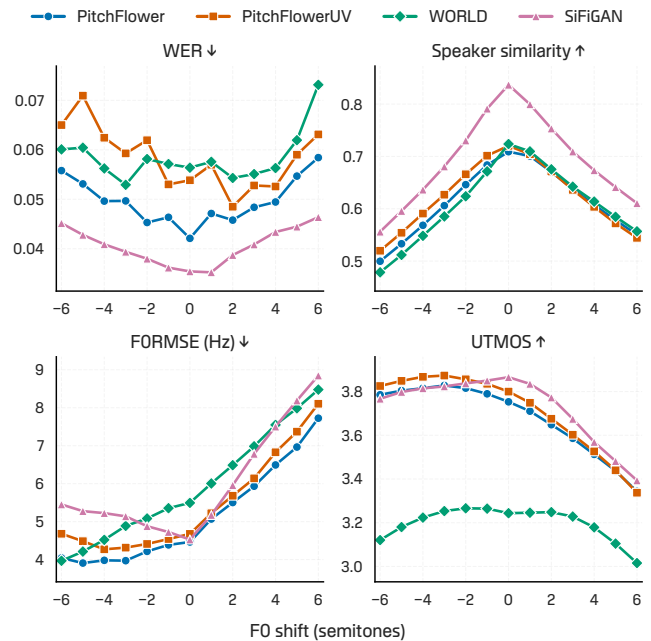


Fig. 3. Objective evaluation results comparing PitchFlower with baselines. An alternative version of our model, PitchFlowerUV, is also considered.

## 6. ABLATIONS

### 6.1. Effect of Bottleneck

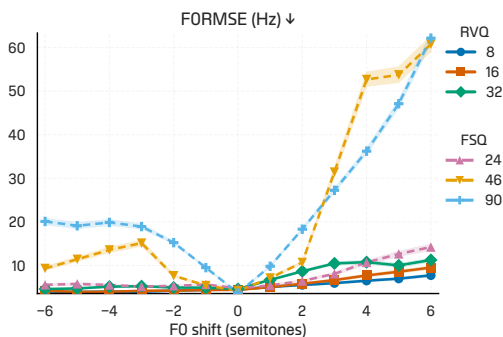
We analyze how the type and size of the bottleneck influence disentanglement in PitchFlower. Starting from the baseline with an RVQ module (8 levels, 512 codes each), we doubled the number of levels and compared against FSQ bottlenecks of similar capacity, using 9 quantization levels per dimension with 24, 46, and 90 dimensions. Figure 4 shows a consistent gap between the two types: for a given capacity, RVQ yields better pitch controllability, suggesting stronger

**Table 1.** Mean Opinion Scores for quality (QMOS) and similarity (SMOS).

Method	QMOS	SMOS
PitchFlower	$3.67 \pm 0.19$	$3.47 \pm 0.22$
PitchFlowerUV	$3.58 \pm 0.20$	$3.57 \pm 0.25$
WORLD	$2.81 \pm 0.25$	$3.21 \pm 0.26$
SiFiGAN	$3.45 \pm 0.19$	$3.46 \pm 0.23$
Real Audio	$4.15 \pm 0.15$	$4.66 \pm 0.14$

disentanglement. Increasing capacity in FSQ rapidly breaks controllability, while RVQ remains stable. As expected, smaller bottlenecks improve disentanglement for both.

Finally, we tested a model without an autoencoder, where the flow directly reconstructs the original mel-spectrogram from its perturbed version. This system, lacking a bottleneck, failed to disentangle pitch, as the flow exploited residual information and WORLD artifacts to recover F0. These results confirm that perturbation and bottleneck play complementary roles: perturbation removes pitch cues explicitly, while the bottleneck prevents their recovery.



**Fig. 4.** Pitch control for different sizes and types of bottleneck

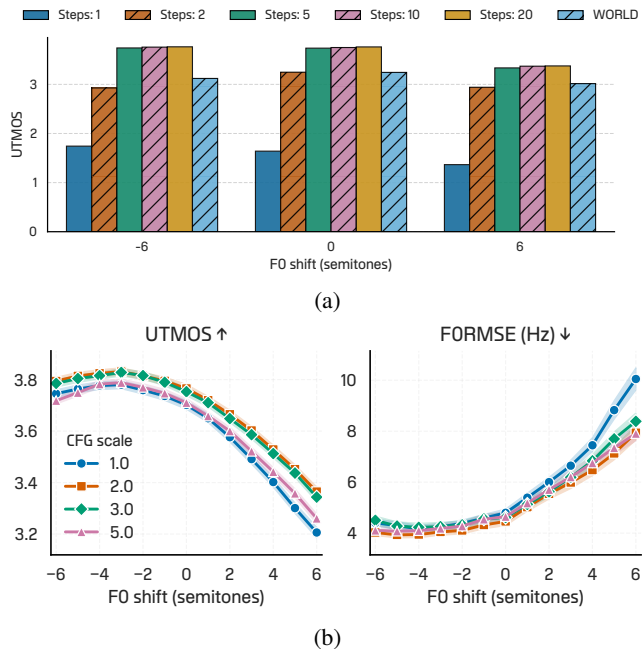
## 6.2. Flow inference parameters

Since flow models require multiple evaluations to generate samples, we examine the effect of the number of steps. As shown in Figure 5, audio quality (UTMOS) saturates at 5 steps, with no further improvements beyond this point. Notably, comparable quality to WORLD is already achieved with only 2 steps.

We also study the classifier-free guidance scale. Figure 5 shows that increasing this parameter improves both audio quality and pitch controllability up to a limit. Values between 2.0 and 3.0 give the best performance, while 5.0 degrades audio quality with only marginal gains in controllability.

## 7. LIMITATIONS

The main limitation of our approach lies in the supported F0 range. Without inductive biases or explicit assumptions, the system can only generate values observed during training. Consequently, transposition quality depends on both the shift factor and the original F0. Shifts of up to  $-1.5$  octaves are feasible when starting from high F0, but performance degrades at low frequencies, saturating near 60 Hz and failing to follow the contour. On LibriTTS, the effective range is roughly 60-700 Hz; extending beyond this requires training on data with a wider F0 distribution.



**Fig. 5.** (a) UTMOS score when changing the number of flow steps. (b) UTMOS and FORMSE curves for different values of the CFG scale.

## 8. CONCLUSIONS

We introduced PitchFlower, a pitch-controllable neural audio codec that achieves disentanglement through perturbation and information masking. Experiments show that PitchFlower surpasses DSP-based baselines and performs on par with state-of-the-art neural approaches. Compared to WORLD, it delivers substantially higher audio quality while maintaining more accurate pitch control. Relative to SiFiGAN, PitchFlower offers stronger controllability and similar audio quality, though with slightly lower speaker similarity, likely due to artifacts introduced by WORLD during the perturbation step.

Our study of alternative disentanglement strategies highlights their different trade-offs. Bottleneck-only approaches achieve high speaker similarity and intelligibility but weak disentanglement. Adversarial methods improve disentanglement but tend to suppress other information, reducing intelligibility and similarity. Semantic distillation with HuBERT further enhances disentanglement, yet at the cost of overall quality. Among these strategies, PitchFlower strikes the most favorable balance between pitch controllability and audio quality.

Looking forward, the framework we propose is not limited to pitch. The same principles could be applied to disentangle other speech attributes such as emotion or timbre. We attribute PitchFlower’s effectiveness to three factors: (i) a perturbation that removes pitch without degrading other information, (ii) a bottleneck that prevents recovery of the masked signal, and (iii) a flow-based decoder capable of reconstructing realistic audio even from perturbed inputs. Together, these elements make PitchFlower a simple yet powerful step toward more controllable neural audio codecs.

## 9. ACKNOWLEDGMENTS

This work was partly performed using HPC resources from GENCI-IDRIS (Grant 2025-AD011016144), and funded by the ANR project EVA (ANR-23-CE23-0018).

## 10. REFERENCES

- [1] Masanori Morise, Fumiya Yokomori, and Kenji Ozawa, “World: A vocoder-based high-quality speech synthesis system for real-time applications,” *IEICE Trans. Inf. Syst.*, vol. E99.D, no. 7, pp. 1877–1884, 2016.
- [2] Frederik Bous and Axel Roebel, “A bottleneck auto-encoder for f0 transformations on speech and singing voice,” *Information*, vol. 13, no. 3, 2022.
- [3] Reo Yoneyama, Yi-Chiao Wu, and Tomoki Toda, “Source-filter hifi-gan: Fast and pitch controllable high-fidelity neural vocoder,” in *ICASSP*, 2023, pp. 1–5.
- [4] Yukiya Hono, Kei Hashimoto, Yoshihiko Nankaku, and Keiichi Tokuda, “Periodgrad: Towards pitch-controllable neural vocoder based on a diffusion probabilistic model,” in *ICASSP*, 2024, pp. 12782–12786.
- [5] Jiarui Hai and Mounya Elhilali, “Diff-pitcher: Diffusion-based singing voice pitch correction,” in *WASPAA*, 2023, pp. 1–5.
- [6] Mathilde Abrassart, Nicolas Obin, and Axel Roebel, “Fastvgan: Lightweight voice conversion with explicit control of f0 and duration parameters,” in *SSW*, 2025.
- [7] Xiaobin Zhuang, Huiran Yu, Weifeng Zhao, Tao Jiang, and Peng Hu, “Karatuner: Towards end-to-end natural pitch correction for singing voice in karaoke,” in *Interspeech*, 2022, pp. 4262–4266.
- [8] Max Morrison, Cameron Churchwell, Nathan Pruyn, and Bryan Pardo, “Fine-grained and interpretable neural speech editing,” in *Interspeech*, 2024, pp. 187–191.
- [9] Neil Zeghidour, Alejandro Luebs, Ahmed Omran, Jan Skoglund, and Marco Tagliasacchi, “Soundstream: An end-to-end neural audio codec,” *IEEE/ACM TASLP*, vol. 30, pp. 495–507, 2022.
- [10] Alexandre Défossez, Jade Copet, Gabriel Synnaeve, and Yossi Adi, “High fidelity neural audio compression,” *TMLR*, 2023.
- [11] Rithesh Kumar, Prem Seetharaman, Alejandro Luebs, Ishaan Kumar, and Kundan Kumar, “High-fidelity audio compression with improved rvqgan,” *NeurIPS*, vol. 36, pp. 27980–27993, 2023.
- [12] Xin Zhang, Dong Zhang, Shimin Li, Yaqian Zhou, and Xipeng Qiu, “Spechtokenizer: Unified speech tokenizer for speech language models,” in *ICLR*, 2024.
- [13] Zeqian Ju, Yuan Cheng Wang, Kai Shen, Xu Tan, Detai Xin, Dongchao Yang, Eric Liu, Yichong Leng, Kaitao Song, Siliang Tang, et al., “Naturalspeech 3: Zero-shot speech synthesis with factorized codec and diffusion models,” in *ICML*, 2024, pp. 22605–22623.
- [14] Masato Takagi, Miku Nishihara, Yukiya Hono, Kei Hashimoto, Yoshihiko Nankaku, and Keiichi Tokuda, “Periodcodec: A pitch-controllable neural audio codec using periodic signals for singing voice synthesis,” in *Interspeech*, 2025, pp. 4913–4917.
- [15] Simon Welker, Matthew Le, Ricky TQ Chen, Wei-Ning Hsu, Timo Gerkmann, Alexander Richard, and Yi-Chiao Wu, “Flowdec: A flow-based full-band general audio codec with high perceptual quality,” in *ICLR*, 2025.
- [16] Nicola Pia, Martin Strauss, Markus Multrus, and Bernd Edler, “Flowmac: Conditional flow matching for audio coding at low bit rates,” in *ICASSP*, 2025, pp. 1–5.
- [17] Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le, “Flow matching for generative modeling,” in *ICLR*, 2023.
- [18] Heiga Zen, Viet Dang, Rob Clark, Yu Zhang, Ron J. Weiss, Ye Jia, Zhifeng Chen, and Yonghui Wu, “Libritts: A corpus derived from librispeech for text-to-speech,” in *Interspeech*, 2019, pp. 1526–1530.
- [19] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie, “A convnet for the 2020s,” in *CVPR*, 2022, pp. 11976–11986.
- [20] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan Gomez, Łukasz Kaiser, and Illia Polosukhin, “Attention is all you need,” *NeurIPS*, vol. 30, 2017.
- [21] Qinqing Zheng, Matt Le, Neta Shaul, Yaron Lipman, Aditya Grover, and Ricky TQ Chen, “Guided flows for generative modeling and decision making,” *CoRR*, 2023.
- [22] Hubert Siuzdak, “Vocos: Closing the gap between time-domain and fourier-based neural vocoders for high-quality audio synthesis,” in *ICLR*, 2024.
- [23] Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever, “Robust speech recognition via large-scale weak supervision,” in *ICML*, 2023, pp. 28492–28518.
- [24] Brecht Desplanques, Jenthe Thienpondt, and Kris Demuyne, “Ecapa-tdnn: Emphasized channel attention, propagation and aggregation in tdnn based speaker verification,” in *Interspeech*, 2020, pp. 3830–3834.
- [25] Jong Wook Kim, Justin Salamon, Peter Li, and Juan Pablo Bello, “Crepe: A convolutional representation for pitch estimation,” in *ICASSP*, 2018, pp. 161–165.
- [26] Takaaki Saeki, Detai Xin, Wataru Nakata, Tomoki Koriyama, Shinnosuke Takamichi, and Hiroshi Saruwatari, “Utmos: Utokyo-sarulab system for voicemos challenge 2022,” in *Interspeech*, 2022, pp. 4521–4525.
- [27] Kaizhi Qian, Yang Zhang, Shiyu Chang, Xuesong Yang, and Mark Hasegawa-Johnson, “Autovc: Zero-shot voice style transfer with only autoencoder loss,” in *ICML*, 2019, pp. 5210–5219.
- [28] Wei-Ning Hsu, Benjamin Bolte, Yao-Hung Hubert Tsai, Kushal Lakhotia, Ruslan Salakhutdinov, and Abdelrahman Mohamed, “Hubert: Self-supervised speech representation learning by masked prediction of hidden units,” *IEEE/ACM TASLP*, vol. 29, pp. 3451–3460, 2021.