PITCHSTAR: SELF-SUPERVISED PITCH STYLE TRANSFER WITH COMPLEMENTARY TEMPORAL AND TIME-AGNOSTIC STYLE REPRESENTATIONS

Leonardo Boulitreau de M. M. Marques and Gaël Richard

LTCI, Télécom Paris, Institut Polytechnique de Paris, France

ABSTRACT

Pitch ornaments such as vibrato and glissando are key to expressive singing, yet transferring them from a reference performance to another remains challenging. Existing systems are often limited to predefined ornaments or rely only on the surrounding pitch context, which might not be reliable due to the dynamic nature of these styles. We introduce PitchSTAR, a flow-matching framework for pitch style transfer. It models a full style reference through note-relative pitch modulation and temporal and time-agnostic branches, which capture phrase-level organization and ornament details and limit trivial copying in a self-supervised training setting. Synthetic experiments verify that these representations are complementary and compositional. On GTSinger, experiments demonstrate that PitchSTAR outperforms all baselines in style similarity, achieving up to a mean 88.9% style consistency across styles while maintaining high pitch accuracy and competitive naturalness, as validated by listening tests.

***Index Terms*—** singing style conversion, pitch style transfer, flow matching, self-supervised learning

1. INTRODUCTION

Singing style conversion (SSC) aims to modify *how* a musical phrase is sung while preserving its content, including the melody, lyrics, and singer identity. This task is inherently ill-posed: the same musical phrase can admit many valid stylistic realizations, and a successful system must generate one that remains consistent with the source content while sounding natural and perceptually convincing [1].

Singing styles are commonly grouped into timbre-based styles, such as mixed voice and breathy, and pitch-based styles, such as vibrato, glissando, and melisma [2]. We refer to the latter as pitch ornaments. Because SSC has only recently emerged as a distinct research problem [3], many current approaches inherit their methods from singing voice conversion [4, 5, 6, 7, 8]. For this reason, the first Singing Voice Style Conversion Challenge [3], realized in 2025, reported that current SSC methods were mostly effective on timbre-based styles, whose defining characteristics tend to be global or slowly varying over time, similarly to singer identity. On the other hand, substantially weaker performance was noted for pitch-based styles [3]. Unlike timbral attributes, pitch ornaments are highly localized and dynamic: their presence, position, duration, and internal trajectory may vary considerably across notes and phrases [9].

With support from the European Union (ERC, HI-Audio - Hybrid and Interpretable Deep Neural Audio Machines, 101052978). Views and opinions expressed are, however, those of the author(s) only and do not necessarily reflect those of the European Union or the European Research Council. Neither the European Union nor the granting authority can be held responsible for them. This work was granted access to the HPC resources of IDRIS under the allocation 2026-AD011018031 made by GENCI.

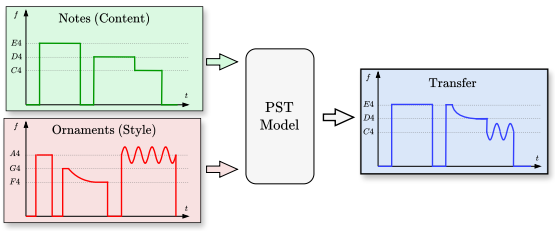


Fig. 1: The Pitch Style Transfer (PST) task: a pitch is generated preserving the target melody and transferring the reference style.

We thus isolate this problem as *Pitch Style Transfer* (PST), illustrated in Figure 1. Given target musical notes as content and an ornamented pitch contour as reference style, PST generates a new pitch contour that follows the target melody while reproducing the reference ornamentation. Treating pitch-style conversion as a separate task provides two advantages. First, it directly targets the dynamic component on which general SSC systems perform poorly. Second, the resulting stylized pitch contour can be used by any pitch-conditioned singing voice synthesizer or vocoder, allowing pitch style to be modified without redesigning the acoustic model.

Earlier PST approaches generally rely on manually designed representations or predefined ornament classes [10]. Vibe-SVC [11, 12] models separate vibrato from the slowly varying pitch using a discrete wavelet transform, enabling conversion between vibrato and non-vibrato singing. Other controllable systems learn a fixed set of singing ornaments from explicit annotations [4]. Although effective for the ornaments represented during training, these methods restrict style to a predefined inventory and thus generalize poorly.

A current state-of-the-art general-purpose stylistic pitch generator is StylePitcher [13]. It formulates pitch generation as self-supervised conditional infilling: given an $\boldsymbol{f}_0$ contour and a binary mask, it learns to reconstruct masked pitch frames with the note sequence, a voiced/unvoiced indicator, and the unmasked surrounding pitch context, from which it must learn to extract the style. On inference, a reference phrase and a target phrase are concatenated, the target pitch is masked, and the reference pitch serves as the context from which the target contour is generated.

StylePitcher provides a strong and flexible baseline for PST because it learns unconstrained ornament patterns without labels. Nevertheless, its infilling formulation introduces two core limitations. First, it outputs absolute pitch while simultaneously being conditioned on the note sequence, requiring the model to reconstruct note levels altogether with th e desired ornamentation. Second, it infers style exclusively from pitch context surrounding the region. However, pitch ornaments are sparse and strongly time-dependent, so the context may contain no ornament or only ornaments occurring at positions with no correspondence with the masked region.

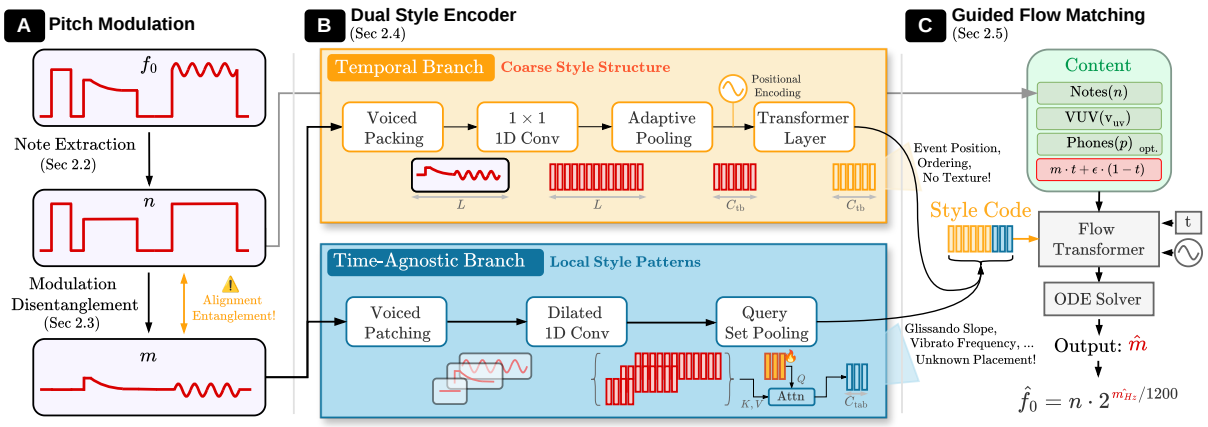


Fig. 2: Training scheme of PitchSTAR. The note-relative pitch modulation is encoded by complementary temporal and time-agnostic branches, which respectively capture the coarse and local characteristics of the reference ornaments while suppressing content leakage.

To address these limitations, we introduce **PitchSTAR**, a self-supervised flow-matching model for PST. Our contributions are threefold: 1) PitchSTAR models the note-relative modulation of the f_0 contour, explicitly separating the melody from its expressive deviations; 2) Instead of conditioning directly on the limited surrounding pitch context, we introduce a complementary dual-branch style encoder that combines temporally structured and time-agnostic representations, which enable full-context conditioning while limiting the transfer of content-related information; and 3) We conduct extensive objective and perceptual evaluations, demonstrating that PitchSTAR improves PST over current state-of-the-art baselines.

2. METHOD

2.1. Overview

The goal of PitchSTAR (Figure 2) is to learn a transport from noise to a singing pitch distribution, guided by content (note sequence $\boldsymbol{n}$ and voicing information $\boldsymbol{v_{uv}}$), and a pitch modulation style $\boldsymbol{m}$. It must be expressive such that the transferred style is perceived as close to the reference while remaining realistic and adhering to the source content. PitchSTAR includes 4 major steps: note extraction (Section 2.2), modulation disentanglement (Section 2.3), dual-branch encoding (Section 2.4), and guided flow matching (Section 2.5).

2.2. Note extraction

The first step of PitchSTAR is to obtain a note transcription of the audio sample $\boldsymbol{a}$. To that aim, a pre-trained model g_ϕ (described in section 3.2) is used to obtain the note boundaries in the form of Onset/Offsets indices $(i_{\text{on}}^{(k)}, i_{\text{off}}^{(k)})$ for each note index k. As suggested in [14], we use the mel-spectrogram A_{mel} and phonetic posteriorgrams A_{ppg} as input representations to the model:

$$\{(i_{\text{on}}^{(k)}, i_{\text{off}}^{(k)})\}_{k=1}^{K} = g_\phi(A_{\text{mel}}, A_{\text{ppg}}), \tag{1}$$

where K is the number of notes. Then, let $\boldsymbol{f}_0 \in \mathbb{R}^I$ denote the pitch curve of the audio signal $\boldsymbol{a}$, where I is the number of analysis frames. To obtain the note frequencies, we process each segment of $\boldsymbol{f}_0$ delimited by each predicted onset/offset pair $(i_{\text{on}}^{(k)}, i_{\text{off}}^{(k)})$. This is done by applying weighted median filtering (WMF) using a Hann window as weights to suppress all intra-segment pitch fluctuations while reducing boundary outliers. The resulting scalar frequency $\boldsymbol{\nu}$ for each note $k \in \{1, \dots, K\}$ is computed as:

$$\boldsymbol{\nu}[k] = \text{WMF}\left(\{\boldsymbol{f}_0[i]\}_{i=i_{\text{on}}^{(k)}}^{i_{\text{off}}^{(k)}}; \boldsymbol{w}_{\text{hann}}\right). \tag{2}$$

A frame-level note sequence is obtained by repeating each $\boldsymbol{\nu}[k]$ over its corresponding segment (defined by its onset and offset):

$$\boldsymbol{n}[i] = \begin{cases} \boldsymbol{\nu}[k] & \text{if } i \in [i_{\text{on}}^{(k)}, i_{\text{off}}^{(k)}] \\ 0 & \text{otherwise.} \end{cases} \tag{3}$$

The voicing of $\boldsymbol{n}$ and $\boldsymbol{f}_0$ is aligned by assigning voiced frames without a detected note to the nearest valid note. Detected notes in unvoiced frames are set to zero, i.e. $\boldsymbol{n} \leftarrow \boldsymbol{n} \cdot \boldsymbol{v}_{uv}$, $\boldsymbol{v}_{uv} = (\boldsymbol{f}_0 > 0)$.
,

2.3. Modulation disentanglement

Style, i.e pitch ornaments, are disentangled from content $\boldsymbol{n}$ through the use of pitch modulation instead of $\boldsymbol{f}_0$. Specifically, we normalize $\boldsymbol{f}_0$ by $\boldsymbol{n}$ and, on voiced frames, compute the deviation in cents:

$$\boldsymbol{m}[i] = 1200 \log_2\left(\frac{\boldsymbol{f}_0[i]}{\boldsymbol{n}[i]}\right). \tag{4}$$

To improve numerical stability, we compress the modulation into a signed logarithmic domain, since $\boldsymbol{m}[i]$ can take positive and negative values. This preserves its sign while reducing dynamic range.

2.4. Dual-branch style encoder

In self-supervision, $\boldsymbol{m}$ is used as both target and reference. Thus, a trivial reference encoder could learn to just copy it from note-modulation alignment: even though $\boldsymbol{n}$ is disentangled from $\boldsymbol{m}$ frequency-wise, they are still perfectly time-aligned (i.e same $\boldsymbol{v}_{\text{uv}}$). We therefore encode the reference through two complementary bottlenecks: A temporal branch, which retains coarse context, and a time-agnostic branch, which retains detailed ornament characteristics. Together, they suppress reference-content leakage while preserving both phrase-level context and fine-grained characteristics.

The *temporal branch* first performs voiced packing: it removes unvoiced frames, then concatenates the remaining modulation. The

packed sequence is processed by a low-capacity (1×1) convolution and adaptively average-pooled into C_{tb} tokens. This suppresses fine variations while preserving the coarse ordering of pitch events along normalized voiced time. Positional embeddings and a shallow Transformer then contextualize phrase-level structure, such as ornament type or placement, without retaining its exact alignment.

Strong temporal pooling, however, may remove local characteristics that are essential for generating an ornament. The *time-agnostic branch* therefore divides the reference into a set of voiced regions (notes) and encodes each using dilated 1-D convolutions. This prevents unvoiced consonants, V/UV boundaries, and transitions between voiced regions from being represented as style. These local features are subsequently aggregated through attention against a fixed set of C_{tab} learned queries without positional encoding. This query-set pooling is thus invariant to the ordering of the local features and therefore discards their absolute placement, while the convolutional receptive fields preserve short-range patterns such as vibrato periodicity and glissando direction.

2.5. Guided flow matching

Following [15], we define the optimal transport probability path $\boldsymbol{m}_t = (1-t)\boldsymbol{\epsilon} + t\boldsymbol{m}$, $\epsilon \sim \mathcal{N}(0, I)$, whose conditional velocity is constant: $\boldsymbol{u}_t(\boldsymbol{m}_t|\boldsymbol{m}) = \boldsymbol{m} - \boldsymbol{\epsilon}$. A transformer-based vector field $\boldsymbol{u}_t^\theta$ predicts this velocity from the noisy modulation $\boldsymbol{m}_t$, the timestep t, the look-up embedded source content $(\boldsymbol{n}, \boldsymbol{v}_{\text{uv}})$, and the dual-branch style code $\boldsymbol{s} = r_\theta(\boldsymbol{m}) = [C_{tb}; C_{tab}]$. $\boldsymbol{m}_t$ is used as the queries, while $\boldsymbol{s}$ conditions the prediction through cross-attention. Since the target is corrupted along the flow and the reference is accessible only through the proposed bottlenecks, the model cannot directly copy its frame-level contour and must instead learn to recover the missing style. The model is optimized on voiced frames through a masked conditional flow-matching objective:

$$\mathcal{L}_{\text{FM}} = \mathbb{E}_{t,\boldsymbol{\epsilon},\boldsymbol{m}} \left[\frac{\left\| \boldsymbol{v}_{uv} \odot \left(\boldsymbol{u}_t^\theta(\boldsymbol{m}_t, t, \boldsymbol{n}, \boldsymbol{v}_{\text{uv}}, \boldsymbol{s}) - (\boldsymbol{m} - \boldsymbol{\epsilon}) \right) \right\|_2^2}{\sum_i \boldsymbol{v}_{uv}[i]} \right], \tag{5}$$

To enable classifier-free guidance, the style code is randomly replaced on training by a learned null condition with probability p_{cfg}. On inference, the learned vector field is integrated from $t \in [0, 1]$. The generated $\hat{\boldsymbol{m}}$ is mapped to Hz and combined with the content $\boldsymbol{n}$ through inversion of Eq. 4, yielding the transferred pitch contour.

3. EXPERIMENTS AND RESULTS

We experiment with PitchSTAR on both synthetic and real singing data. Section 3.1 introduces a synthetic experiment to analyze the effect of each branch on learning global and local style attributes and their complementarity. Section 3.2 introduces the dataset used, its processing pipeline, and all considered model configurations. Section 3.3 presents the evaluation scenarios and results.

3.1. Synthetic Illustrative Example

To investigate the importance of each branch, a synthetic pitch distribution is built with known local and global ornament parameters. We construct note sequences $\boldsymbol{n}$ with random note tones, durations, and intervals. Ornaments are sinusoidal vibratos of a half-semitone:

$$\boldsymbol{f}_0 = \boldsymbol{n} + A_{0.5\text{st}} \sin\left(2\pi f_{\text{vib}} i\right) \cdot v_{\text{place}}[i] \cdot w_{\text{n}}[i], \tag{6}$$

in which two parameters are defined: a global vibrato placement $v_{\text{place}}[i] \in \mathbb{R}^I$, and local vibrato frequency $f_{vib} \in \mathbb{R}$. v_{place} is a Kaiser window ($\alpha = 3$), centered at $0.5 \cdot \mathcal{N}(0.2, 0.1) + 0.5 \cdot \mathcal{N}(0.8, 0.1)\%$ of the sample length. f_{vib} is drawn once from $0.5 \cdot \mathcal{N}(3, 0.2) + 0.5 \cdot \mathcal{N}(6, 0.2)$Hz. For smooth vibrato boundaries, Kaiser windows w_n are also applied to each note. We train PitchSTAR on this dataset with 20k training samples for 20k steps.

We evaluate two scenarios. First, we perform a *branch ablation* by removing each branch during training. On inference, we condition the model only on samples with early placement ($\approx 20\%$) and high frequency ($\approx$ 6Hz). This experiment determines which ornament parameters are transferable from each single-branch training: one is expected to transfer global attributes, whereas the other should retain local characteristics. To obtain f_{vib} of the transfers, we compute the short-time Fourier transform of $\hat{\boldsymbol{m}}$ and time-average it. To estimate v_{place}, we compute the analytic signal using the Hilbert transform and extract its amplitude envelope. Figure 3 (top) shows the results. As shown by the orange curves, when using only the temporal branch, the model preserves the early ornament placement (dashed line), but produces a broader, bimodal vibrato frequency distribution (solid line). Conversely, only the time-agnostic branch reproduces the reference frequency mode around 6 Hz, but is ambiguous between early and late placement. Combining both branches resolves these uncertainties, yielding a concentrated high-frequency mode together with the intended early temporal envelope.

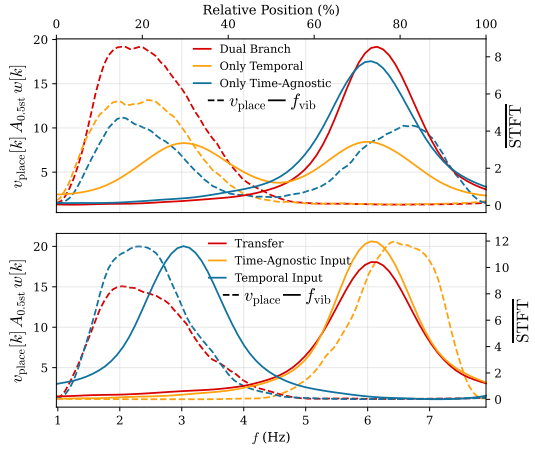


Fig. 3: (Top) Distributions of synthetic ornaments. (Bottom) Complementarity of branches.

Secondly, we perform *cross-reference synthesis*, in which the temporal encoder input is an early-placement, low-frequency reference, and the time-agnostic input is a late-placement and high-frequency reference. If the style codes provide complementary control, the generated contour should combine the early placement of the former with the high vibrato frequency of the latter. The results shown in Figure 3 (bottom) show that the codes are compositional: the transferred output (red curves) combines the approximately 6 Hz frequency distribution of one input with the early-position distribution of another, instead of reproducing either reference as a whole. These results show that the two bottlenecks provide complementary control over local statistics and phrase-level temporal organization.

3.2. Setup

Data. We use the GTSinger dataset [16]. It contains 29k high-quality singing phrases from 1366 songs, most between 5s and 15s, totaling 80h. These are sung by 20 singers across 9 languages, in

various singing styles, both timbre-based (Breathy, Falsetto, Mixed Voice, Pharyngeal) and pitch-based (Glissando and Vibrato). We use song-disjoint splits, with 250 validation and 500 test phrases.

Pre-Processing. $\boldsymbol{f}_0$ curves are extracted with the open-source state-of-the-art pre-trained RMPVE [17] model, with 16 kHz, 1024 FFT bins, a 160-sample hop, and downsampled to 50Hz to shorten sequence lengths. To extract note onsets and offsets, an open-source pre-trained phoneme-informed transcription model is used [14]. In the dual-encoder, we found $C_{tb} = 8$ and $C_{tab} = 4$ yielded the best results. Dilated Conv1Ds use a kernel size of 5 and dilations of (1, 2). We use 5 transformer layers with 8 heads and a dimension of 512, totalizing 29M parameters. We train it on 2 L40S GPUs, with an effective batch size of 64, for 100k steps, with AdamW, lr=10^{-4}, a 5k-step linear warm-up, cosine scheduling, and $p_{\text{cfg}} = 0.5$. We use an Euler solver with 16 timesteps, and a cfg scale of 1.10.

Models. The following models are evaluated: **StylePitcher** [13] (our re-implementation): predicts a masked segment of the pitch curve, given content notes and surrounding pitch context as style; **StylePitcher w/ Mod**: StylePitcher operating our proposed modulation curves instead of raw pitch; **VEVO 2 [18]**: A token-based current state-of-the-art SSC model trained on large datasets and used in several SSC benchmarks [3]. **PitchSTAR**: The proposed framework; and **PitchSTAR+Phones**: We evaluate whether using content phonemes, available on GTSinger, mapped to articulation categories (e.g., plosive, fricative, nasal), improves the generated pitch. All audios are generated with a 44.1kHz pre-trained version of the NSF-HiFiGAN [19] vocoder by simply substituting the original pitch curve with the stylized one, except for VEVO 2 which is not pitch controllable, and generates audio directly from tokens.[1]

3.3. Evaluation

As objective metrics, we analyze pitch difference using Raw Pitch Accuracy (RPA) within both the 50- and 25-cent limits. Each test sample's $\boldsymbol{f}_0$ is split into content $\boldsymbol{n}$ and style (modulation $\boldsymbol{m}$, or $\boldsymbol{f}_0$ for StylePitcher). These are fed to the models for reconstruction. We also analyse neural consistency with a pitch style classifier trained on modulations to predict the style labels available on GTSinger. A detection F1-Score of 89% for Vibrato, 93% for Glissando, and 89% for No Ornamentation is obtained on the test set. We select the 10 most confidently classified samples per class of the training set and use them as references for the entire test set, totaling 5000 samples. We check if the same label is predicted for the reference and transfer.

Model	Reconstruction		Neural Consistency		
	RPA_{50}(↑)	RPA_{25}(↑)	V-F1(↑)	G-F1(↑)	N-F1(↑)
StylePitcher [13]	46.85	23.88	24.98	46.62	32.38
w/ Mod	66.73	43.09	70.16	55.29	55.81
VEVO 2 [18]	49.17	29.52	50.22	40.39	70.43
PitchSTAR	**80.34**	**63.24**	**97.07**	**86.29**	<u>83.34</u>
+Phones	<u>79.99</u>	<u>62.44</u>	<u>92.04</u>	<u>82.89</u>	**84.41**

Table 1: Objective results. V-F1, G-F1, and N-F1 stand for the per-class Vibrato, Glissando, and No Ornamentation F1 scores. Bold and underlined values represent the best and second-best values.

Table 1 presents the results. Using pitch modulation instead of raw pitch improves StylePitcher's reconstruction and transfer, particularly for vibrato. PitchSTAR outperforms all pitch-infilling baselines, achieving the highest neural consistency among all styles, especially for glissando. Despite its large-scale training, we noticed VEVO 2 frequently alters melodic timing, resulting in poor reconstruction compared with the modulation-based methods, all of which preserve exact source voicing and constrain outputs to deviations around content notes. We found phoneme conditioning yielded no audible perceptual improvements or significant pitch reconstruction.

A perceptual test is conducted to evaluate the audios in two aspects: Naturalness and Style Similarity to a reference, from 0 to 100. For each model and ornament, we selected one transferred sample from the style consistency experiment that yielded the best correct confidence score of the pitch style classifiers, summing to a total of 8 samples per test. We used the unstyled content pitches as a low anchor for style similarity, and a ground-truth real pitch curve in the target style as a high anchor for naturalness. 15 participants were selected, with an average of 8 years of musical training.

Model	Style Similarity		Naturalness	
	V-MOS (↑)	G-MOS (↑)	V-MOS (↑)	G-MOS (↑)
GT	-	-	$90.54_{\pm 2.89}$	$80.07_{\pm 8.03}$
Source	$33.76_{\pm 7.89}$	$42.62_{\pm 8.35}$	-	-
StylePitcher [13]	$37.10_{\pm 6.90}$	$48.17_{\pm 6.13}$	$42.82_{\pm 10.76}$	$43.25_{\pm 9.08}$
w/ Mod	<u>60.41</u>$_{\pm 6.79}$	<u>56.10</u>$_{\pm 8.23}$	<u>46.14</u>$_{\pm 8.74}$	<u>47.29</u>$_{\pm 7.56}$
VEVO 2 [18]	$48.43_{\pm 6.46}$	$48.88_{\pm 8.55}$	$40.07_{\pm 9.63}$	$42.71_{\pm 9.01}$
PitchSTAR	**67.40**$_{\pm 6.15}$	**66.86**$_{\pm 6.98}$	**49.79**$_{\pm 8.24}$	**48.36**$_{\pm 6.57}$

Table 2: Subjective results. Style Similarity and Naturalness scores are shown with 95% confidence intervals. "V" stands for vibrato and "G" for Glissando.

Table 2 shows that PitchSTAR achieves the highest mean scores in all scenarios. Relative to StylePitcher w/ Mod, it improves style similarity by 6.99 points for vibrato and 10.76 points for glissando, while the corresponding naturalness gains are smaller, at 3.65 and 1.07 points. Pairwise Wilcoxon signed-rank tests with Bonferroni correction confirm significance over the competing systems, except for glissando naturalness against StylePitcher w/ Mod. The consistent gains of StylePitcher w/ Mod over StylePitcher support the use of modulation. VEVO 2 receives the lowest naturalness ratings. Importantly, none of the systems closes the naturalness gap to ground truth, indicating that improving perceptual realism remains an open challenge in scenarios of strong stylistic control.

4. CONCLUSION AND LIMITATIONS

We introduced PitchSTAR, a self-supervised framework that transfers pitch ornaments using note-relative modulation and complementary temporal and time-agnostic style representations. Synthetic experiments demonstrate that these representations capture distinct, compositional ornament attributes, while objective and perceptual evaluations on GTSinger show improved pitch preservation, style consistency, and perceived quality over existing state-of-the-art approaches. On the other hand, the smaller naturalness gains and remaining gap to ground truth indicate that transferred contours still lack realism. Moreover, our evaluation is limited to vibrato and glissando. Future work will investigate joint pitch–acoustic modeling to improve perceptual naturalness.

[1] Audio demos, code, and weights available at https://pitchstar.netlify.app/

5. REFERENCES

[1] Ziqian Wang, Xianjun Xia, Chuanzeng Huang, and Lei Xie, "S^2voice: Style-aware autoregressive modeling with enhanced conditioning for singing style conversion," *arXiv preprint arXiv:2601.13629*, 2026.

[2] Yuya Yamamoto, Juhan Nam, and Hiroko Terasawa, "Analysis and detection of singing techniques in repertoires of j-pop solo singers," in *Ismir Hybrid Conference*, 2022.

[3] Lester Phillip Violeta, Xueyao Zhang, Jiatong Shi, Yusuke Yasuda, Wen-Chin Huang, Zhizheng Wu, and Tomoki Toda, "The singing voice conversion challenge 2025: From singer identity conversion to singing style conversion," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2026, pp. 17707–17711.

[4] Zhetao Hu, Yiquan Zhou, Wenyu Wang, Zhiyu Wu, Xin Gao, and Jihua Zhu, "Controllable singing style conversion with boundary-aware information bottleneck," *arXiv preprint arXiv:2604.05526*, 2026.

[5] Matthew Le, Apoorv Vyas, Bowen Shi, Brian Karrer, Leda Sari, Rashel Moritz, Mary Williamson, Vimal Manohar, Yossi Adi, Jay Mahadeokar, and Wei-Ning Hsu, "Voicebox: Text-guided multilingual universal speech generation at scale," in *Advances in Neural Information Processing Systems*, 2023, vol. 36, pp. 14005–14034.

[6] Lester Phillip Violeta, Wen-Chin Huang, and Tomoki Toda, "Serenade: A singing style conversion framework based on audio infilling," in *33rd European Signal Processing Conference (EUSIPCO)*, 2025, pp. 411–415.

[7] Yiquan Zhou, Wenyu Wang, Hongwu Ding, Jiacheng Xu, Jihua Zhu, Xin Gao, and Shihao Li, "Syki-svc: Advancing singing voice conversion with post-processing innovations and an open-source professional testset," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2025, pp. 1–5.

[8] Xueyao Zhang, Junan Zhang, Yuancheng Wang, Chaoren Wang, Yuanzhe Chen, Dongya Jia, Zhuo Chen, and Zhizheng Wu, "Vevo2: A unified and controllable framework for speech and singing voice generation," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 34, pp. 2262–2275, 2026.

[9] Marti Umbert, Jordi Bonada, Masataka Goto, Tomoyasu Nakano, and Johan Sundberg, "Expression control in singing voice synthesis: Features, approaches, evaluation, and challenges," *IEEE Signal Processing Magazine*, vol. 32, no. 6, pp. 55–73, 2015.

[10] Jordi Bonada and Merlijn Blaauw, "Hybrid neural-parametric f0 model for singing synthesis," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 7244–7248.

[11] Joon-Seung Choi, Dong-Min Byun, Hyung-Seok Oh, and Seong-Whan Lee, "VibE-SVC: Vibrato Extraction with High-frequency F0 Contour for Singing Voice Conversion," in *Interspeech*, 2025, pp. 1233–1237.

[12] Joon-Seung Choi, Dong-Min Byun, and Seong-Whan Lee, "Vibrato expression control for singing voice conversion with improving independent control," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 34, pp. 3358–3370, 2026.

[13] Jingyue Huang, Qihui Yang, Fei-Yueh Chen, Julian McAuley, Randal Leistikow, Perry R. Cook, and Yongyi Zang, "Stylepitcher: Generating style-following and expressive pitch curves for versatile singing tasks," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2026, pp. 15737–15741.

[14] Sangeon Yong, Li Su, and Juhan Nam, "A phoneme-informed neural network model for note-level singing transcription," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2023, pp. 1–5.

[15] Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le, "Flow matching for generative modeling," in *The Eleventh International Conference on Learning Representations*, 2023.

[16] Yu Zhang, Changhao Pan, Wenxiang Guo, Ruiqi Li, Zhiyuan Zhu, Jialei Wang, Wenhao Xu, Jingyu Lu, Zhiqing Hong, Chuxin Wang, LiChao Zhang, Jinzheng He, Ziyue Jiang, Yuxin Chen, Chen Yang, Jiecheng Zhou, Xinyu Cheng, and Zhou Zhao, "Gtsinger: A global multi-technique singing corpus with realistic music scores for all singing tasks," in *Advances in Neural Information Processing Systems*, 2024, vol. 37, pp. 1117–1140.

[17] Haojie Wei, Xueke Cao, Tangpeng Dan, and Yueguo Chen, "RMVPE: A Robust Model for Vocal Pitch Estimation in Polyphonic Music," in *Interspeech*, 2023, pp. 5421–5425.

[18] Xueyao Zhang, Junan Zhang, Yuancheng Wang, Chaoren Wang, Yuanzhe Chen, Dongya Jia, Zhuo Chen, and Zhizheng Wu, "Vevo2: A unified and controllable framework for speech and singing voice generation," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 34, pp. 2262–2275, 2026.

[19] Reo Yoneyama, Yi-Chiao Wu, and Tomoki Toda, "Source-filter hifi-gan: Fast and pitch controllable high-fidelity neural vocoder," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2023, pp. 1–5.