

HIERARCHICAL GENERATIVE MODELING OF MELODIC VOCAL CONTOURS IN HINDUSTANI CLASSICAL MUSIC

Nithya Shikarpur^{1,2} Krishna Maneesha Dendukuri¹ Yusong Wu^{1,2}
Antoine Caillon⁴ Cheng-Zhi Anna Huang^{1,2,3,4}

¹ Mila, Quebec Artificial Intelligence Institute, ² Université de Montréal,
³ Canada CIFAR AI Chair, ⁴ Google DeepMind

snnithya@mit.edu, krishnamaneeshad@gmail.com, wu.yusong@mila.quebec
{acaillon, annahuang}@google.com

ABSTRACT

Hindustani music is a performance-driven oral tradition that exhibits the rendition of rich melodic patterns. In this paper, we focus on generative modeling of singers’ vocal melodies extracted from audio recordings, as the voice is musically prominent within the tradition. Prior generative work in Hindustani music models melodies as coarse discrete symbols which fails to capture the rich expressive melodic intricacies of singing. Thus, we propose to use a finely quantized pitch contour, as an intermediate representation for hierarchical audio modeling. We propose GaMaDHaNi, a modular two-level hierarchy, consisting of a generative model on pitch contours, and a pitch contour to audio synthesis model. We compare our approach to non-hierarchical audio models and hierarchical models that use a self-supervised intermediate representation, through a listening test and qualitative analysis. We also evaluate audio model’s ability to faithfully represent the pitch contour input using Pearson correlation coefficient. By using pitch contours as an intermediate representation, we show that our model may be better equipped to listen and respond to musicians in a human-AI collaborative setting by highlighting two potential interaction use cases (1) primed generation, and (2) coarse pitch conditioning.

1. INTRODUCTION

Hindustani music is a performance-driven music tradition that has a high level of melodic intricacy [1]. Despite the recent advances in generative modeling for music [2, 3], this genre remains difficult to model for several reasons including (1) a lack of a readily available and widely accepted abstract representation reflecting the genre faithfully (like Western symbolic notation), (2) as a niche musical form, the scarcity of available datasets restricts the ability to model the raw waveform directly.

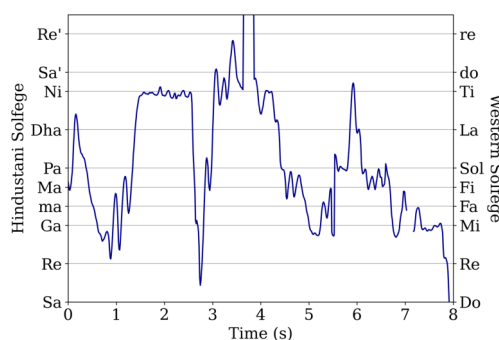


Figure 1. Extracted pitch from Hindustani vocal audio highlighting the melodic intricacies involved. Solfège notation is highlighted as a horizontal grid.

Symbolic notation is a well-defined discrete representation of music including lead sheet, MIDI, piano roll, text, and markup language. Musical notation used in Hindustani pedagogy uses a similar discrete representation by highlighting the prominent notes which fails to faithfully capture the fine melodic intricacies connecting these notes as seen in Fig. 1. Previous work on generative modeling for Hindustani music has side-stepped the lack of well-defined abstract representations with two methods: (1) using musical notation from textbooks or music theory [4–6], (2) leveraging MIDI extracted from audio [7, 8]. However, both methods ignore the rich melodic ornamentation present in this music. Computational analyses for the genre have addressed the difficulty in data representation by using the fundamental frequency contour, hereby referred to as ‘pitch’, as an intermediate representation for several melodic tasks including music style classification [9], motif discovery and matching [10–12] and *raga* recognition [13–15]. With evidence that pitch faithfully represents the melody for computational tasks, we are motivated to incorporate it in the context of generative modeling.

In this work, we present GaMaDHaNi¹ (Generative Modular Design of Hierarchical Networks), a modular hierarchical generative model for Hindustani singing. We employ a two-level hierarchy of data representation in-



© N. Shikarpur, K. M. Dendukuri, Y. Wu, A. Caillon and C. Z. A. Huang. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** N. Shikarpur, K. M. Dendukuri, Y. Wu, A. Caillon and C. Z. A. Huang, “Hierarchical Generative Modeling of Melodic Vocal Contours in Hindustani Classical Music”, in *Proc. of the 25th Int. Society for Music Information Retrieval Conf.*, San Francisco, United States, 2024.

¹Listen to audio samples and access code here: <https://snnithya.github.io/gamadhani-samples/>

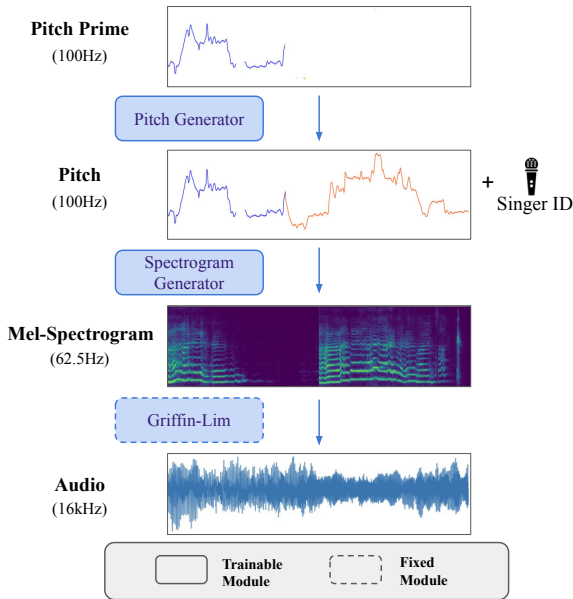


Figure 2. The overall hierarchical generation structure of GaMaDHaNi comprising of the Pitch Generator, the Spectrogram Generator and a vocoder. During inference, given an optional short melodic input, i.e. ‘prime’, each of the generators produce a pitch continuation and a spectrogram conditioned on the resulting pitch respectively.

cluding pitch and spectrogram. The Pitch Generator and Spectrogram Generator are trained to generate these respectively, with the generated spectrogram converted to audio using a vocoder. Fig. 2 highlights the model’s high-level structure. We choose a finely quantized pitch contour as an intermediate representation due to its close relation to melodic content, strongly established in prior literature [9–15]. We model pitch under two paradigms: as discrete tokens using an autoregressive transformer and as continuous values using a diffusion model. In addition, with a relatively small dataset of 120 hours, we find that the pitch intermediate representation is effective at learning melodically diverse ideas (Sec. 4.4). As possible use cases for interaction, (1) we explore using the model to continue a given melodic prompt, termed ‘prime’, as seen in Fig. 2, and (2) we extend the hierarchy upwards to include a coarse pitch target, thereby enabling user-driven steering of the generation process.

We note that our current generation pipeline lacks incorporation of several key elements crucial to Hindustani music, specifically tonic frequency, and raga and tala, i.e. melodic and rhythmic frameworks. This work establishes a preliminary foundation for exploring the potential of generating music within this form while maintaining its characteristic melodic intricacies.

A summary of our core contributions include:

- We propose GaMaDHaNi, the first model capable of generating Hindustani vocal contours while maintaining the rich melodic complexity in the music.
- We present a hierarchical approach to modeling a waveform using an intermediate pitch representation

that works on a small dataset (120 hours).

- Through listening tests and qualitative observations, we show that our hierarchical approach performs better than baselines.

2. RELATED WORK

2.1 Music Representations in Indian Art Music

Past work on melody-based computational tasks for Indian Art Music include music style classification [9], motif discovery and matching [10–12], and raga recognition [14–16]. Previous work shows that fine quantization outperforms coarse quantization in pitch contours for tasks including raga recognition [16, 17] and motif matching [11]. Thus motivated by their ability to capture melodic information we use finely quantized pitch as an intermediate representation. Additionally, for Carnatic music, previous work on compact representations for Gamakas (type of note ornamentation) [18], and non-uniform pitch quantization schemes that can preserve raga-characteristics [19, 20] present forms of representation that are more condensed than the pitch contour while being adequately detailed which could be an interesting inclusion for future work.

2.2 Generative Modeling for Hindustani Music

Hindustani music is an improvised form of music where melodic movements are guided by a melodic framework (raga) [1]. Past work on the generation of this music is of two types: rule-based and data-driven models. AI-Raga [4] is a rule-based AI system developed to generate musical notation of compositions and improvisations that adhere to raga grammar based on an elaborate set of rules termed ‘generative theory of music’ [21]. Another work develops a Finite State Machine (FSM) to generate improvisations based on raga-specific melodic movements situated in theory [5]. An initial attempt at data-driven models learned from the musical notation of *alaps*, i.e. slow improvisation, in textbooks using bigrams in an FSM [6]. RMMM [7] explores the use of LSTM [22] and transformer-based [23] architectures to generate MIDI extracted from a corpus of Hindustani music. Other work also proposes generating MIDI with GANs [8, 24]. All models discussed in this section approach modeling data as solfege notation. While doing so, one gives up on the transitory melodic regions between notes of the melody, which is inherent to Hindustani music. AI-Raga [4] partially addresses this by using domain-informed tuning systems, and a simulation of transitory glides between notes. We propose to address this problem by incorporating a fine pitch data representation. Additionally, in contrast to previous work, we propose to generate audio waveform rather than symbolic data.

2.3 Hierarchical Audio Generation

Within the domain of music generation, hierarchical learning offers two distinct advantages: enhanced learning abilities on data-constrained tasks and multi-level controllability. MIDI-DDSP [25] takes advantage of the hierarchy in

the process of creating realistic audio of instrument performance given a sequence of MIDI data including notes, high-level performance attributes and low-level synthesis attributes. Our approach leverages a different hierarchy based on pitch as opposed to MIDI notes, and we generate pitch from scratch without relying on any symbolic input. Moreover, we choose to directly generate audio spectrograms instead of DDSP synthesis parameters since the latter is designed mainly for instrumental sound.

Another approach to hierarchical models for audio includes the generation of pre-trained compressed representations of audio, i.e. neural audio codecs [26, 27], framed as a language modeling task as seen in MusicLM [28] and MusicGen [3]. We study the effectiveness of this approach as a baseline in our experiments in Sec. 4.3, by comparing Encodec [26] and pitch as intermediate representations.

The use of fundamental frequency contours as an intermediate representation has been widely adopted in the context of Text To Speech synthesis (TTS) and Singing Voice Synthesis (SVS). Both fields follow a hierarchy including an input-conditioned acoustic model which mainly generates a subset of pitch, duration, and spectral features followed by a vocoder. The input could be text in the case of TTS [29–31] and musical score for SVS [32, 33]. CDAR [34] is a TTS model that seeks to control the prosody of generated speech by allowing users to edit parts of the spoken pitch contour while maintaining the realism of the prosody. We thus choose to adopt pitch as an intermediate representation with a strong precedence for its use and controllability in speech and singing applications.

3. METHOD

In this work, we seek a generative model for Hindustani vocal music by learning the joint distribution of amplitude mel-spectrograms s and pitch f following

$$p(s, f) = p_\phi(s|f)p_\theta(f), \quad (1)$$

where p_ϕ and p_θ are parameterized with neural networks called *Spectrogram* and *Pitch* Generators respectively. The generated spectrogram is converted to audio using a vocoder. Pitch conditioning f to p_ϕ is taken from our dataset for training and sampled from p_θ for inference.

3.1 Pitch Generator

We study the modeling of vocal pitch as the primary component in our hierarchical generation pipeline. Vocal pitch f are represented as integer-valued sequences sampled at 100Hz, with 90% of the values ranging from 86Hz to 899Hz, quantized with a fine resolution of 10 cents. To model such sequences, we investigate two distinct methods. The first employs an autoregressive, language-like model to predict the discrete pitch sequence, whereas the second leverages recent advancements in diffusion-based modeling for iterative generation of the entire sequence.

3.1.1 Discrete autoregressive model

We use a vanilla decoder-only transformer, to autoregressively predict the next token of a pitch sequence. In this

task, the pitch values f are considered to be discrete tokens in a vocabulary V , each mapped to an embedding vector of size d through an embedding matrix $E \in R^{|V| \times d}$. The model is trained with cross-entropy loss.

3.1.2 Continuous diffusion model

We use a simple yet effective diffusion variant, Iterative α -Deblending (IADB) [35] as the training objective of our model that generates finely quantized pitch f . IADB defines a simplified diffusion process that is a linear interpolation between noise $x_0 \sim X_0 = \mathcal{N}(0, 1)$ and data $x_1 \sim X_1 = X_{data}$:

$$x_\alpha = (1 - \alpha)x_0 + \alpha x_1. \quad (2)$$

We leverage a deterministic iterative deblending process proposed in [35] to sample a data point $x_1 \sim X_1$ from noise $x_0 \sim X_0$. With the total number of iterations in the process as T , and given a time step $t \in \{0, 1, 2, \dots, T\}$, we define the blending parameter $\alpha_t = \frac{t}{T}$ and an α -blended point x_{α_t} . Thus, the iterative deblending is defined as:

$$x_{\alpha_{t+1}} = (1 - \alpha_{t+1})\bar{x}_0 + \alpha_{t+1}\bar{x}_1, \quad (3)$$

where $(\bar{x}_0, \bar{x}_1) = E_{(X_0 \times X_1)|x_{\alpha_t}, \alpha_t}$ is the expected value of the posterior samples given x_{α_t}, α_t . Heitz et. al. [35] show that using expected posteriors \bar{x}_0, \bar{x}_1 in the deblending process (Eq. 3) instead of x_0, x_1 converges to the same point, while making the sampling process deterministic.

Taking the derivative of x_{α_t} with respect to the blending parameter α_t , the training objective becomes,

$$D_\theta(x_{\alpha_t}|\alpha_t) \approx \frac{dx_{\alpha_t}}{d\alpha_t} = (\bar{x}_1 - \bar{x}_0), \quad (4)$$

Taking a trained model D_θ , we perform an iterative sampling procedure to generate outputs:

$$x_{\alpha_{t+1}} = x_{\alpha_t} + (\alpha_{t+1} - \alpha_t)D_\theta(x_{\alpha_t}, \alpha_t), \quad (5)$$

3.2 Spectrogram Generator

On the next level of the hierarchy, we train a model to generate a spectrogram conditioned on pitch, which is then converted to an audio signal using a vocoder. This method uses IADB as described in Sec. 3.1.2, while additionally conditioned on singer and pitch. Each singer ID is embedded as a discrete vector, and the processed pitch is time-downsampled to match the spectrogram’s time axis. Both conditioning signals are concatenated as additional channels to the mel-spectrogram input. Thus given a conditioning signal c , the training objective $D_\phi(x_{\alpha_t}|\alpha_t, c)$ is similar to Eq. 4 but is additionally conditioned on c .

The singer and pitch values are conditioned using classifier-free guidance (CFG) [36]. Given a conditioning strength w , CFG is implemented such that $\overline{D_\phi}(x_{\alpha_t}|c)$ is used during the iterative sampling, defined as,

$$\overline{D_\phi}(x_{\alpha_t}|\alpha_t, c) = (1-w)D_\phi(x_{\alpha_t}|\alpha_t) + wD_\phi(x_{\alpha_t}|\alpha_t, c) \quad (6)$$

4. EXPERIMENTS

In this paper, we consider the Spectrogram Generator as a tool to convert melodic ideas from the Pitch Generator into perceivable audio. As a result, we evaluate both the Generators with a focus on quality of pitch generation and the spectrogram’s fidelity in representing that pitch.

Through our experiments, we aim to motivate our choices for (1) a hierarchical approach to generation, (2) the use of pitch as an intermediate representation, through listening tests. We also qualitatively evaluate the overall melodic quality of generations. Additionally, we assess the Spectrogram Generator by testing pitch adherence: the ability of the model to reliably reproduce the pitch conditioning through quantitative and qualitative analyses. We leave evaluation of other aspects of the Spectrogram Generator such as audio quality, singer adherence to future work. Readers are encouraged to listen to relevant supplementary audio samples on our project website while going through this and the following sections.

4.1 Dataset

We use a combination of the Saraga and Hindustani Raga Recognition datasets [37, 38]. Audio files in the combined dataset contain audio of vocal performances including the tanpura, i.e. a drone, along with the melodic and rhythmic accompaniment across 56 unique singers. It spans about 120 hours across 362 audio files, where the files range from 88 seconds (s) to 1.2 hours with a median duration of 20 minutes. The dataset is randomly split into training and validation sets at a 90:10 ratio. Furthermore, each audio file is split into 60 s segments resulting in 7174 and 719 segments in the training and validation sets respectively. Due to different inductive biases in the models used, they all have different receptive fields and are thus trained on sequences with lengths varying from 8.2 s-12 s, randomly sampled from the 60 s segments during training.

The vocals are isolated using 2-stem source separation with HT Demucs [39] and further, the pitch is extracted using CREPE [40] and is sampled at 100 Hz. We algorithmically reduce the number of pitch detection errors using a loudness-based pitch filtering approach; using a sliding window to calculate area under the loudness curve, we retain only corresponding pitch values exceeding an empirically set threshold. We normalize the pitch to a logarithmic scale such that an arbitrarily chosen frequency, 440Hz is 0 on this scale, and quantize it into 10-cent bins. Additionally, during training, the pitch is transposed by a random multiple of 10 cents within a range of $[-400, 400]$ cents.

Artifacts in the dataset Our source separation model, HT Demucs [39], allows some leakage from other instruments including mainly the *sarangi* (stringed melodic accompaniment) and the *tabla* (rhythmic accompaniment) as artifacts in the vocal stem due to the out of distribution nature of Hindustani music data for the model. These ‘leaked’ sounds are generated in our models too (both our proposed model and the baselines established). Additionally, instances of speech are found in some generated samples as it is present in our dataset. The Carnatic FTA-Net

[41], presents a domain-informed model trained to extract pitch contours from Carnatic vocal audio. Owing to the similarities between Carnatic and Hindustani music, an interesting direction for future work would be to adopt their methodologies in our data processing pipeline.

4.2 Model Architectures

Below we present model specific architectures and data preprocessing for the Pitch Generators (Autoregressive and Diffusion) and the Spectrogram Generator.

Pitch Generator (Discrete Autoregressive) This model was trained on 12s (1200 token) sequences. The quantized pitch f is converted into a sequence of discrete embedding vectors e , using an embedding space $E \in \mathbb{R}^{|V| \times d}$ where effective vocabulary size is $|V| = 796$ and embedding dimension is $d = 512$. The model is a decoder-only transformer [23] with 8 layers, with each layer having an output dimension of 512. AliBi positional method [42] is used to encode the position of tokens in the sequence. A cosine learning schedule with linear warm-up is used. Samples are generated with a temperature of 0.99 and using top k sampling with $k=40$.

Pitch Generator (Continuous Diffusion) This model was trained on 10.24s (1024 elements) sequences. The quantized pitch contour is limited to a range of 400 integers. This distribution is converted into a continuous Gaussian using the quantile function which maps a variable’s probability distribution to another probability distribution. This model is implemented as a U-Net with three down-sampling and upsampling layers each with a stride of 4, 2 and 2 respectively. Each layer is made of four 1-D convolution layers with weight normalization [43] and Mish non-linearity [44]. The bottleneck involves 4 attention layers with 8 heads each.

Spectrogram Generator This model is trained on 8.2s (512 elements) of mel-spectrogram sequences. The relevant pitch conditioning is linearly interpolated and down-sampled to match the sequence length of the spectral data. The spectral data is produced with 192 mels and a hop size of 256 (0.016 s) given 16 kHz audio and is converted to a continuous Gaussian distribution using the quantile transform function as well. Apart from additional channels for singer and pitch conditioning, the architecture is the same as that used by the Pitch Generator (Continuous Diffusion) (Sec 4.2). For simplicity, spectrograms are converted to audio using the Griffin-Lim algorithm [45]. Future work could harness the power of recent developments in neural vocoders including HiFi-GAN [46].

4.2.1 Conditioning signals

In addition to pitch, the Spectrogram Generator utilizes singer conditioning to help maintain the consistency of the voice in generated audio as seen in the supplementary audio samples. Each singer is assigned a unique ID and mapped to an embedding vector of size $d_{singer} = 128$. Conditioning was implemented with CFG as discussed in Sec. 3.2 with a strength of $w = 3$ for pitch and singer conditioning. This value was determined based on empirical

studies as an optimal balance between fidelity to pitch and minimizing artifacts due to incorrect pitch extraction.

4.3 Baseline Models

Through our baseline models, we aim to motivate two major architectural choices: (1) hierarchy in the model and (2) an intermediate pitch representation. These models thus include a non-hierarchical baseline, a hierarchical baseline with a self-supervised intermediate representation (hierarchical Encodec baseline), and the ground truth.

Non-hierarchical Baseline In this baseline, we highlight a naive approach of modeling audio directly with no hierarchy. We train a diffusion model with the IADB objective directly on processed audio mel-spectrograms. The model architecture is similar to other diffusion models used in this paper (Sec. 4.2) and was trained on the same dataset as our model with sequences of length 8.2s.

Hierarchical Encodec Baseline We train a hierarchical autoregressive baseline on a self-supervised intermediate representation, Encodec [26]. Through this model, we aim to compare the effect of self-supervised and pitch intermediate representations. To this end, we train MSPrior [47, 48], a decoder-only transformer adapted for real-time use, on Encodec tokens [26] extracted using the 24 kHz Encodec model with a target bandwidth of 3 kbps (4 channels per token). This model was trained on only the Hindustani Raga Recognition Dataset (which constitutes about $\frac{5}{6}$ th of our dataset) with a sequence length of 900 (12 s). We use a temperature of 0.99 for sampling.

Ground Truth To set the gold standard of melodic quality, we use ground truth pitch for comparison. As the listening test focuses on evaluating the Pitch Generator, we standardize audio quality across all models (except the hierarchical Encodec baseline which already generates waveform) by synthesizing the ground truth pitch with our Spectrogram Generator. We use five singers (3 low and 2 high voice range) with reasonable representation in the dataset as singer conditioning. Depending on the range of the generated pitch, we randomly select from the appropriate set of singers to generate audio for the contour.

4.4 Human Evaluation on Melodic Quality

To evaluate the musical quality and characteristics of generated samples, we conduct a listening study and offer qualitative observations supported by audio examples in our supplementary material.

Listening study We compare five systems: non-hierarchical baseline, hierarchical Encodec baseline, autoregressive and diffusion variants of our method, and ground truth. Participants were presented with 8.2 s audio samples, from two random systems and asked to rate which one is more musically interesting, on a 5-point Likert scale. We recruited 15 participants who are trained in Hindustani or Carnatic music. Although Carnatic music is stylistically different from Hindustani music, the two share the context of raga and tala giving participants enough context to evaluate samples for this study. Participants' primary instruments were the voice or other melodic instruments includ-

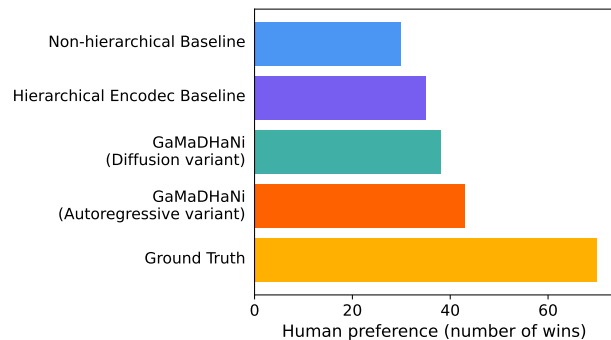


Figure 3. Results from the listening study, showing how many times each system was preferred.

ing the harmonium, sarangi, sarod, sitar, flute, or violin. We collected 240 ratings, with each system involved in 96 comparisons.

Results Fig. 3 shows the number of wins in each system. We ran a Kruskal-Wallis H test and confirmed that there are statistically significant pairs among the combinations. According to a post-hoc analysis using the Wilcoxon signed-rank test with Bonferroni correction (with $p < 0.05/10$), we find that our hierarchical model with an autoregressive Pitch Generator outperforms the non-hierarchical baseline. Given the small sample size, we also compare all systems against each other by aggregating ratings and considering them as independent samples. Using the Independent (Mann-Whitney U) test with Bonferroni correction, we find that both our models, discrete autoregressive and continuous diffusion outperform the non-hierarchical baseline significantly. Through these experiments, we establish that our model outperforms the non-hierarchical baseline.

Diversity in Generation Participants did not prefer our methods significantly more than the hierarchical Encodec baseline. This baseline tends to hold a single note or move through a few stable notes without much dynamism. This understandably was preferred by participants as *vilambit alap* or slower improvisation, a common way to establish a raga in Hindustani music, involves the use of such long and stable notes. With only 8.2 s duration audio samples, the listeners do not have enough time to notice the lack of dynamic movement. In contrast, our proposed methods can render both slow and fast movements, resulting in more variety as seen in generated samples. We hypothesize that this could be due to the different intermediate representations of both models, i.e. due to the importance of intricate melodic movements, a model trained to explicitly generate fine pitch would be able to capture melodic complexity.

Consistency of vocal timbre We note that generations from the hierarchical model, which includes singer conditioning, display more consistency in the timbre of voice; the baseline models sometimes abruptly switch vocal timbre in the middle of generation.

4.5 Pitch Adherence in Spectrogram Generator

Although the Spectrogram Generator loss lacks an explicit term for pitch adherence, we evaluate it by calculating the

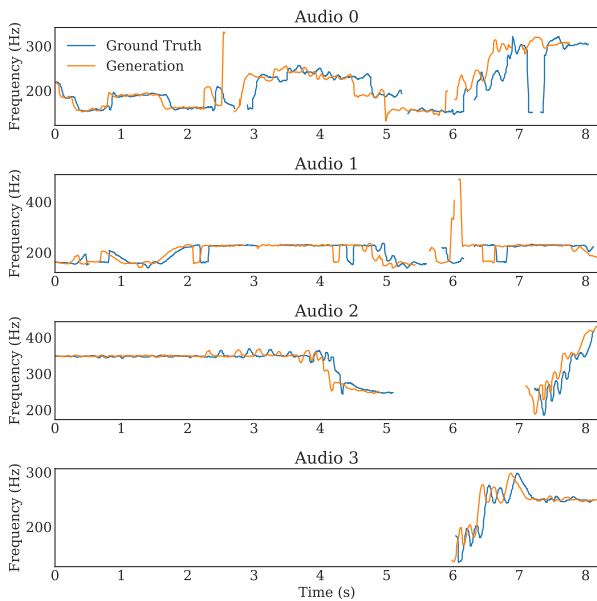


Figure 4. Examples of ground truth pitch (blue) and extracted pitch contour from the generated sample (orange) to highlight pitch adherence with low and high correlation, r (top to bottom). **Low correlation:** Audio 0 ($r = 0.1$) and 1 ($r = 0.11$) are examples of errors in pitch detection. **High correlation:** Audio 2 ($r = 0.94$) and 3 ($r = 0.99$)

Pearson correlation coefficient between the conditioning pitch and the pitch extracted from the generated audio. For this, we choose four singers (two male and two female) to generate audio conditioned on 32 random contours from the validation set resulting in a total of 128 contours to evaluate. We achieve a mean correlation of 0.71 between input and loudness-filtered extracted pitch.

Visual inspection reveals that differences between the input and extracted pitch sequences are pronounced when artifacts due to errors in pitch detection, source separation, or ground truth are present in either sequence. We present instances of samples with high and low correlation in Fig. 4. In addition, we note an inconsistent difference in timing between the ground truth and generated contour in Fig. 4. Future work could investigate pitch-specific training objectives and alternative conditioning representations to improve the precision of the generated audio’s pitch in time. Overall, based on visual analysis, we note that our model faithfully reconstructs the pitch conditioning shape.

5. INTERACTION USE CASES

We show two interactive use cases of GaMaDHaNi: (1) continuing an input melodic sequence or ‘prime’, and (2) guiding generation with coarse solfege-like notation.

5.1 Primed Generation

We investigate using our model for melodic sequence continuation. To this end, we input a four-second pitch sequence from our dataset termed ‘prime’ into our Pitch Generator, and ask the model to continue the sequence. The model can generate realistic-sounding continuations with

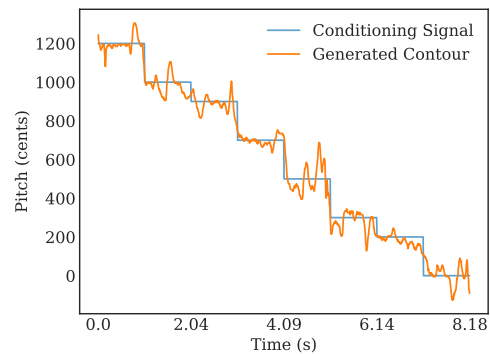


Figure 5. A staircase descending scale (in blue) as a coarse input. This input is then processed as described in Sec. 5.2 and fed into the model. The generated fine-grain contour (in orange) has glides (mindh) and fast jerky movement (gamak) characteristic to Hindustani music.

interesting patterns, as seen in Fig. 2 and in our audio samples. Future work could involve creating an interactive pipeline that would allow our model to directly take input from the user allowing a human-machine collaboration.

5.2 Coarse Pitch Conditioning

To explore further possibilities for interaction, we evaluate the model’s ability to adhere to solfege-like conditioning given to the Pitch Generator. To this end, a ‘coarse pitch’ signal is inferred by calculating a moving average of the pitch with a window size of 1s and a hop size of 0.01s. The Pearson correlation coefficient between the input and generated coarse pitch is 0.97, and between the ground truth and generated pitch is 0.79. Both values are averaged over 64 random samples from the validation set. Thus solfege input, once converted into a similar coarse pitch signal, can be used to guide the model’s generation as seen in Fig. 5, where the model renders a solfege-based descending scale into realistic-sounding audio. Although simple, this is an interesting avenue for interactive generation that we plan to explore in the future.

6. CONCLUSION

We present a modular hierarchical system to generate melodically rich Hindustani vocal audio using a relatively small dataset. Our model has comparable or better performance than established baselines while including an interpretable intermediate pitch representation. We present interesting forms of interaction including primed generations and coarse pitch conditioning that could be developed further to achieve interactive human-machine music making.

There are interesting future directions such as the use of tonic, raga and rhythmic aspects as conditioning for generation. Additionally, the Spectrogram Generator could adopt more advanced vocoders and conditioning signals such as loudness and phoneme features for better results.

7. ETHICS STATEMENT

This work, to our knowledge, is the first model trained to explicitly generate Hindustani vocal music and thus we find it important to emphasize that this work is intended to foster human-AI collaboration, creating a more accessible environment for creative exploration and is by no means intended to replace music teachers or musicians. While we acknowledge the ethical concerns involved in modeling singing voices, we include singer conditioning in our approach with the sole intention of maintaining voice consistency in the generated samples. Additionally, we note that this work utilizes datasets contributed by artists or institutes holding distribution rights to ensure responsible use with informed consent. These datasets were released with appropriate permissions to process audio recordings for research purposes. However, despite our current model's limited scope, future enhancements may pose a risk of mimicking the identities of existing singers, necessitating the establishment of protective guidelines for artists.

8. ACKNOWLEDGMENT

We thank all of our listening study participants for their invaluable contributions and insights. We also appreciate their promptness in completing the tests, which greatly facilitated this work.

9. REFERENCES

- [1] W. Van der Meer, *Hindustani music in the 20th century*. Martinus Nijhoff Publishers, The Hague, 1980.
- [2] A. Agostinelli, T. I. Denk, Z. Borsos, J. Engel, M. Verzetti, A. Caillon, Q. Huang, A. Jansen, A. Roberts, M. Tagliasacchi *et al.*, "Musiclm: Generating music from text," *arXiv preprint arXiv:2301.11325*, 2023.
- [3] J. Copet, F. Kreuk, I. Gat, T. Remez, D. Kant, G. Synnaeve, Y. Adi, and A. Défossez, "Simple and controllable music generation," in *Proc. of the Advances in Neural Information Processing Systems*, 2024.
- [4] V. Vidwans, "Computational music," accessed on 2024-4-11. [Online]. Available: <https://computationalmusic.com/>
- [5] D. Das and M. Choudhury, "Finite state models for generation of hindustani classical music," in *Proc. of the International Symposium on Frontiers of Research in Speech and Music*, 2005, pp. 59–64.
- [6] H. V. Sahasrabudde, "Analysis and synthesis of hindustani classical music," accessed: 2024-07-22. [Online]. Available: https://www.cse.iitb.ac.in/~hvs/paper_1992.html
- [7] S. Gopi and F. William, "Introductory studies on raga multi-track music generation of indian classical music using ai," *The International Conference on AI and Musical Creativity*, 2023.
- [8] S. Adhikary, M. S. M, S. S. K, S. Bhat, and K. P. L, "Automatic music generation of indian classical music based on raga," in *Proc. of the IEEE International Conference for Convergence in Technology (I2CT)*, 2023.
- [9] A. Vidwans, K. K. Ganguli, and P. Rao, "Classification of Indian classical vocal styles from melodic contours," in *Proc. of the CompMusic Workshop*, 2012.
- [10] S. Gulati, J. Serra, V. Ishwar, and X. Serra, "Mining melodic patterns in large audio collections of indian art music," in *Proc. of the International Conference on Signal-Image Technology and Internet-Based Systems*. IEEE, 2014.
- [11] K. K. Ganguli, A. Rastogi, V. Pandit, P. Kantan, and P. Rao, "Efficient melodic query based audio search for hindustani vocal compositions." in *Proc. of the International Society for Music Information Retrieval (ISMIR)*, 2015.
- [12] T. Nuttall, G. Plaja-Roglans, L. Pearson, and X. Serra, "In search of sañcaras: Tradition-informed repeated melodic pattern recognition in carnatic music," in *Proc. of the International Society for Music Information Retrieval (ISMIR)*, 2022.
- [13] P. Chordia and S. Şentürk, "Joint recognition of raag and tonic in north indian music," *Computer Music Journal*, vol. 37, no. 3, pp. 82–98, 2013.
- [14] S. Gulati, J. Serra, V. Ishwar, S. Sentürk, and X. Serra, "Phrase-based rāga recognition using vector space modeling," in *Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2016.
- [15] M. Clayton, P. Rao, N. N. Shikarpur, S. Roychowdhury, and J. Li, "Raga classification from vocal performances using multimodal analysis." in *Proc. of the International Society for Music Information Retrieval (ISMIR)*, 2022.
- [16] P. Chordia and S. Şentürk, "Joint recognition of raag and tonic in north indian music," *Computer Music Journal*, vol. 37, no. 3, pp. 82–98, 2013.
- [17] G. K. Koduria, S. Gulatia, P. Rao, and X. Serra, "Rāga recognition based on pitch distribution methods," *Journal of New Music Research*, 2012.
- [18] S. K. Subramanian, "Modelling gamakas of carnatic music as a synthesizer for sparse prescriptive notation," Ph.D. dissertation, Master's thesis, National University of Singapore, 2013.
- [19] H. Ranjani, A. Srinivasamurthy, D. Paramashivan, and T. V. Sreenivas, "A compact pitch and time representation for melodic contours in indian art music," *The Journal of the Acoustical Society of America*, vol. 145, no. 1, pp. 597–603, 2019.

- [20] V. S. Viraraghavan, A. Pal, H. Murthy, and R. Aravind, "State-based transcription of components of carnatic music," in *Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020.
- [21] V. Vidwans, *The Music of Minds and Machines*. FLAME University, Pune, 2023.
- [22] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
- [23] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.
- [24] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Proc. of the Advances in Neural Information Processing Systems*, 2014.
- [25] Y. Wu, E. Manilow, Y. Deng, R. Swavely, K. Kastner, T. Cooijmans, A. Courville, C.-Z. A. Huang, and J. Engel, "Midi-ddsp: Detailed control of musical performance via hierarchical modeling," in *Proc. of the International Conference on Learning Representations*, 2022.
- [26] A. Défossez, J. Copet, G. Synnaeve, and Y. Adi, "High fidelity neural audio compression," *arXiv preprint arXiv:2210.13438*, 2022.
- [27] N. Zeghidour, A. Luebs, A. Omran, J. Skoglund, and M. Tagliasacchi, "Soundstream: An end-to-end neural audio codec," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 30, pp. 495–507, 2021.
- [28] A. Agostinelli, T. I. Denk, Z. Borsos, J. Engel, M. Verzetti, A. Caillon, Q. Huang, A. Jansen, A. Roberts, M. Tagliasacchi *et al.*, "Musiclm: Generating music from text," *arXiv preprint arXiv:2301.11325*, 2023.
- [29] T. Yoshimura, K. Tokuda, T. Masuko, T. Kobayashi, and T. Kitamura, "Simultaneous modeling of spectrum, pitch and duration in hmm-based speech synthesis," in *Proc. of the European conference on speech communication and technology*, 1999.
- [30] Y. Fan, Y. Qian, F.-L. Xie, and F. K. Soong, "Tts synthesis with bidirectional lstm based recurrent neural networks," in *Proc. of the Annual Conference of the International Speech Communication Association (INTERSPEECH)*, 2014.
- [31] H. Li, Y. Kang, and Z. Wang, "Emphasis: An emotional phoneme-based acoustic model for speech synthesis system," in *Proc. of the Annual Conference of the International Speech Communication Association (INTERSPEECH)*, 2018.
- [32] P. Lu, J. Wu, J. Luan, X. Tan, and L. Zhou, "Xiaoic-eSing: A High-Quality and Integrated Singing Voice Synthesis System," in *Proc. of the Annual Conference of the International Speech Communication Association (INTERSPEECH)*, 2020.
- [33] Y. Yi, Y. Ai, Z. Ling, and L. Dai, "Singing voice synthesis using deep autoregressive neural networks for acoustic modeling," in *Proc. of the Annual Conference of the International Speech Communication Association (INTERSPEECH)*, 2019.
- [34] M. Morrison, Z. Jin, J. Salamon, N. J. Bryan, and G. J. Mysore, "Controllable Neural Prosody Synthesis," in *Proc. of the Annual Conference of the International Speech Communication Association (INTERSPEECH)*, 2020.
- [35] E. Heitz, L. Belcour, and T. Chambon, "Iterative α -(de) blending: A minimalist deterministic diffusion model," in *Proc. of the ACM SIGGRAPH Conference*, 2023, pp. 1–8.
- [36] J. Ho and T. Salimans, "Classifier-free diffusion guidance," in *Proc. NeurIPS Workshop on Deep Generative Models and Downstream Applications*, 2021.
- [37] A. Srinivasamurthy, S. Gulati, R. C. Repetto, and X. Serra, "Saraga: open datasets for research on indian art music," *Empirical Musicology Review*, vol. 16, no. 1, pp. 85–98, 2021.
- [38] S. Gulati, J. Serrà Julià, K. K. Ganguli, S. Sentürk, and X. Serra, "Time-delayed melody surfaces for rāga recognition," in *Proc. of the International Society for Music Information Retrieval (ISMIR)*, 2016.
- [39] S. Rouard, F. Massa, and A. Défossez, "Hybrid transformers for music source separation," in *Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2023.
- [40] J. W. Kim, J. Salamon, P. Li, and J. P. Bello, "Crepe: A convolutional representation for pitch estimation," in *Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018.
- [41] G. Plaja-Roglans, T. Nuttall, L. Pearson, X. Serra, and M. Miron, "Repertoire-specific vocal pitch data generation for improved melodic analysis of carnatic music," *Transactions of the International Society for Music Information Retrieval*, vol. 6, no. 1, pp. 13–26, 2023.
- [42] O. Press, N. A. Smith, and M. Lewis, "Train short, test long: Attention with linear biases enables input length extrapolation," in *Proc. of the International Conference on Learning Representations*, 2021.
- [43] T. Salimans and D. P. Kingma, "Weight normalization: A simple reparameterization to accelerate training of deep neural networks," in *Proc. of the Advances in Neural Information Processing Systems*, 2016.

- [44] D. Misra, “Mish: A self regularized non-monotonic activation function,” in *Proc. of the British Machine Vision Conference*, 2020.
- [45] D. Griffin and J. Lim, “Signal estimation from modified short-time fourier transform,” *IEEE Transactions on acoustics, speech, and signal processing*, vol. 32, no. 2, pp. 236–243, 1984.
- [46] J. Kong, J. Kim, and J. Bae, “Hifi-gan: Generative adversarial networks for efficient and high fidelity speech synthesis,” in *Proc. of the Advances in neural information processing systems*, 2020.
- [47] A. Caillon, “Msprior,” <https://github.com/caillonantoine/msprior>, 2023.
- [48] —, “Hierarchical temporal learning for multi-instrument and orchestral audio synthesis,” Ph.D. dissertation, Sorbonne université, 2023.