SEMI-SUPERVISED VIOLIN FINGERING GENERATION USING VARIATIONAL AUTOENCODERS

Vincent K.M. Cheung **Hsuan-Kai Kao** **Li Su**
Institute of Information Science, Academia Sinica, Taiwan
{cheung,hsuankai,lisu}@iis.sinica.edu.tw

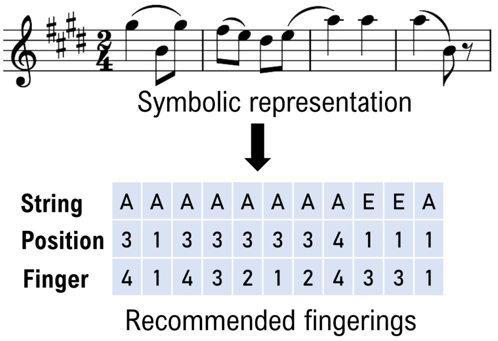
ABSTRACT

There are many ways to play the same note with the fingerboard hand on string instruments such as the violin. Musicians can flexibly adapt their string choice, hand position, and finger placement to maximise expressivity and playability when sounding each note. Violin fingerings therefore serve as important guides in ensuring effective performance, especially for inexperienced players. However, fingering annotations are often missing or only partially available on violin sheet music. Here, we propose a model based on the variational autoencoder that generates violin fingering patterns using only pitch and timing information found on the score. Our model leverages limited existing fingering data with the possibility to learn in a semi-supervised manner. Results indicate that fingering annotations generated by our model successfully imitate the style and preferences of a human performer. We further show its significantly improved performance with semi-supervised learning, and demonstrate our model's ability to match the state-of-the-art in violin fingering pattern generation when trained on only half the amount of labelled data. [1]

1. INTRODUCTION

Musicians produce different pitches on string instruments such as the violin and guitar by pressing on a particular string with their fingerboard hand (typically the left) to temporarily reduce its length. The string oscillates at a higher frequency and a higher pitch is consequently sounded. However, apart from the lowest and highest notes of the instrument, the mapping between pitch and fingering (i.e., where along the fingerboard and with which finger to press) is not unique [1].

For the violin, musicians are faced with the decision of selecting an appropriate string, hand position, and finger placement for every note they play [2]. Such decisions depend on the trade-off between artistic expression and playability [3]. For example, playing a note on the open string is the easiest as no finger placements are required [2]. However, its distinctive, brighter timbre is often undesired as it breaks the consistency in sound quality over a musical context [3, 4]. Musicians instead tend to play these notes on a lower string with vibrato to achieve a warmer and richer tone [5]. Likewise, using the same finger for different pitches consecutively is often avoided as this incurs a constant shift in hand position that could lead to poor intonation or unintended glissandi [1]. Selecting an effective string, position, and fingering combination to sound each note is therefore a non-trivial aspect of violin playing that could shape the outcome of a performance.



String	A	A	A	A	A	A	A	A	E	E	A
Position	3	1	3	3	3	3	3	4	1	1	1
Finger	4	1	4	3	2	1	2	4	3	3	1

Figure 1. Our proposed learning model generates violin fingerings from pitch and timing information found on the score (Example: Elgar's *Salut d'amour*, bars 1-4).

The importance of violin fingerings is evinced as they often appear on the musical score as performance directions or reminders for the musician [6]. They are also used as a pedagogical aid for inexperienced violinists [7]. However, the majority of violin sheet music does not come with fingering annotations. Sampling from the International Music Score Library Project (imslp.org), one of the largest digital repositories of public domain sheet music, 85% of 23,142 scores featuring the violin do not contain any fingering information [2]. In other words, most annotations are still done by hand by the musician, in a process that requires experience and is often time-consuming [8]. Therefore, there is a crucial need for a model that not only generates fingerings, but also requires few labelled data as prior knowledge.

To this end, we propose a violin fingering generation model based on the variational autoencoder [9–11]. Our

[1] Example code and supplementary information are available at https://github.com/vkmcheung/violin-ssvae

[2] Determined by inspecting the first available violin part of every 400th entry in the category *Scores featuring the violin* on 3 May 2021.

Attribution: Vincent K.M.Cheung, Hsuan-Kai Kao, and Li Su, "Semi-supervised violin fingering generation using variational autoencoders", in *Proc. of the 22nd Int. Society for Music Information Retrieval Conf.,* Online, 2021.

model relies only on pitch and timing information found on the musical score as inputs, and can be trained in a semi-supervised manner. This allows our model to capitalise on the predominantly unlabelled existing data in generating fingerings that conform to the style and preferences of violinists.

2. RELATED WORK

Work on fingering generation is not exclusive to the violin and has previously been explored in other musical instruments such as guitar [8, 12, 13] and piano [14–16]. Nevertheless, despite the popularity of the instrument, there exist few models on violin fingering generation. Early approaches have focussed on fingering generation through heuristic rules and dynamic programming. For example, Maezawa et al. [1, 3, 17] introduced three 'consistency rules' for their model to ensure that generated fingerings were consistent in direction and magnitude during pitch and string changes, as well as during a mordent. Fingering generation was thus achieved by minimising the transition cost within a context of two and three notes using a musical score and an audio recording. However, the multimodal nature of the input and rigidity of these models mean that adapting generated fingerings to match individual preferences or styles is not straightforward.

Later works have remedied this problem with learning models. For example, hidden Markov models have been trained on violin textbooks [2] to complement partially annotated fingerings [18]. A recent deep learning model [7] has also combined a pretrained bidirectional long short-term memory (BLSTM) neural network with heuristic rules to generate fingerings with different options. This enabled musicians to select fingerings according to their preferences in e.g., staying in a lower position, or to minimise hand-position shifting. However, the paucity of violin sheet music with labelled fingerings means that there might not always be sufficient training data. By contrast, our semi-supervised approach enables our proposed model to make use of unlabelled data during training to generate high-quality fingering annotations even in the context of limited labelled data.

3. METHODS

Here, we briefly review the background behind semi-supervised variational autoencoders before introducing our proposed model and metrics for performance evaluation.

3.1 Variational autoencoders (VAEs)

VAEs [9, 10] are a popular class of deep generative models that are prized for their ability in estimating complex probability distributions through variational inference [19]. Let X be some observed data generated by latent variable z. We want to learn parameters θ and ϕ that optimise the (log-)likelihood $p_\theta(x|z)$, parametrised by θ, and approximate posterior $q_\phi(z|x)$, parametrised by ϕ. This is achieved by maximising the evidence lower bound (ELBO). If we further assume that $p(z){=}\mathcal{N}(\mathbf{0}, I)$ and $q_\phi(z|x){=}\mathcal{N}(z|\mu_\phi, diag(\sigma^2_\phi))$, then we can use a reparametrisation trick to write samples of z as transformations of a standard Gaussian random variable, i.e.

$$z_i = \mu_i + \sigma_i \epsilon \tag{1}$$

for some $\epsilon \sim \mathcal{N}(\mathbf{0}, I)$. This allows us to compute gradients of the ELBO to optimise θ and ϕ using neural networks, for which $q_\phi(z|x)$ is often referred to as the encoder and $p_\theta(x|z)$ the decoder. In practice, a non-negative hyperparameter β is often added to the ELBO to control the extent to which the approximate posterior $q_\phi(z|x)$ resembles the prior $p(z)$, i.e.

$$\begin{aligned}\log p_\theta(x) \geq \mathbb{E}_{q_\phi(z|x)}&[\log p_\theta(x|z)] \\ &- \beta D_{\mathrm{KL}}(q_\phi(z|x) \parallel p(z)).\end{aligned} \tag{2}$$

We have the original VAE formulation when $\beta = 1$, whilst reconstruction is improved at the expense of a more entangled latent representation when $0 < \beta < 1$ [20, 21].

3.2 Semi-supervised VAEs

The intuition behind semi-supervised VAEs [11] is to capitalise on its generative ability and to extend the VAE latent space to include information from a classifier. Reconstruction errors from the unlabelled data can then be explicitly used to update the classifier during backpropagation.

Formally, let (X, Y) be some observed (partially-) labelled data generated by a continuous latent variable z. Suppose $p(z){=}\mathcal{N}(z|\mathbf{0}, I)$ and $p(y){=}Cat(y|\pi)$, where the latter is a multinomial distribution with distribution π, and that the likelihood $p_\theta(x|y, z)$ is parametrised using a neural network (the decoder). We can again use variational inference to approximate the intractable posterior $p(y, z|x)$ with $q_\phi(y, z|x)$.

Now assuming that $q_\phi(y, z|x){=}q_\phi(y|x)q_\phi(z|x)$, we can construct the approximate posterior using a neural network with two components: a multinomial classifier $q_\phi(y|x){=}Cat(y|\pi(x))$, and a Gaussian encoder with diagonal covariance matrix $q_\phi(z|x) = \mathcal{N}(z|\mu_\phi(x), \sigma^2_\phi(x))$.

As before, finding suitable values for θ and ϕ amounts to optimising the ELBO. For labelled data, that is

$$\begin{aligned}\log p_\theta(x, y) \geq \mathbb{E}_{q_\phi(z|x,y)}&[\log p_\theta(x|y, z)] \\ &- \beta D_{\mathrm{KL}}(q_\phi(z|x) \parallel p(z)) = L(x, y).\end{aligned} \tag{3}$$

For unlabelled data, we can treat the missing label y' as an additional categorical latent variable that generates the observed data and assume that y', z are marginally independent. However, backpropagating through samples from a multinomial distribution is problematic as the operation is not differentiable. Fortunately, we can approximate this sampling operation with the Gumbel-Softmax distribution [22, 23], for which samples can be drawn via the reparametrisation trick

$$y'_i = \frac{\exp((\log(\pi_i) + g)/\tau)}{\sum_{j=1}^{L} \exp((\log(\pi_j) + g)/\tau)}, \tag{4}$$

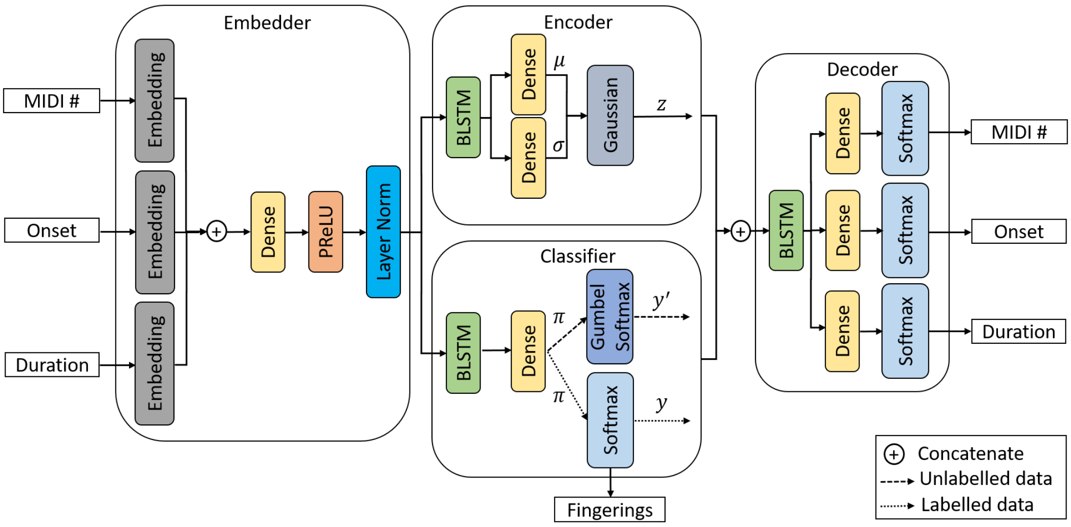


Figure 2. Model architecture. Our VAE-based model supports semi-supervision by treating missing labels from unlabelled data as an additional latent variable for reconstructing pitch and timing information.

where $g \sim Gumbel(0, 1)$, L denotes the number of classes, and τ controls how strongly the distribution approximates the multinomial distribution.

With the two reparametrisation tricks at hand, the ELBO is now maximised for unlabelled data as follows:

$$\begin{aligned}\log p_\theta(x) \geq\ & \mathbb{E}_{q_\phi(y',z|x)}[\log p_\theta(x|y',z)] \\ & - \beta D_{\mathrm{KL}}(q_\phi(z|x) \parallel p(z)) \\ & - \beta D_{\mathrm{KL}}(q_\phi(y'|x) \parallel p_\pi(y')) \\ =\ & U(x).\end{aligned} \tag{5}$$

Lastly, a classification loss is introduced to the classifier $q_\phi(y|x)$ for labelled data. The overall objective of this model is thus to maximise

$$\mathcal{J} = \mathbb{E}_{D_L}\left[L(x,y) + \log q_\phi(y|x)\right] + \mathbb{E}_{D_U}\, U(x), \tag{6}$$

where D_L and D_U denote labelled and unlabelled data, respectively.

3.3 Model architecture

Our proposed model (Figure 2) consists of four modules: embedder, encoder, classifier, and decoder. The embedder accepts a sequence of notes as inputs, where each note is represented by a numeric vector denoting its MIDI number, onset, and duration. The sequence is passed through embedding layers of dimension 16, 8, and 4 for the three respective features, concatenated, and then fed into a dense layer of 64 units with a PReLU activation function before leaving the module through a layer normalisation layer.

Outputs from the embedder are then passed in parallel onto the encoder and classifier. These go through a bidirectional long short-term memory (BLSTM) layer of 64×2 units in the encoder before being mapped onto a Gaussian latent space of 16 dimensions as output via a reparametrisation trick (Equation 1).

The embedder output is likewise passed through a BLSTM layer of 128×2 units in the classifier. This is followed by a dense layer of N_{spf} units, where N_{spf} denotes the number of possible (string, position, finger) arrangements. By considering fingerings as the joint distribution of string, position, and finger, we can model dependencies between these three labels. Otherwise, different optimal (string, position, fingering) combinations might be predicted for each label separately if only their marginal distributions are considered. For labelled data, a softmax activation function is subsequently applied as output of the classifier. This provides a probability density estimate for each fingering combination given the input. For unlabelled data, logits from the dense layer are mapped onto a latent Gumbel-softmax distribution as outputs via a reparametrisation trick as described in Equation 4.

Finally, outputs from the encoder and classifier are concatenated and passed through a 128×2-unit BLSTM layer in the decoder. This is followed by three softmax-activated dense layers of N_{MIDI}, N_{onset}, N_{duration} units as outputs, which denote the number of MIDI, onset, and duration classes, respectively.

3.4 Dataset

We use a recently published dataset of symbolic violin performance for the current study [7]. This dataset is a compilation of 217,690 note-by-note annotations of 14 solo violin excerpts as performed by 10 professional violinists. The excerpts are selected from diverse styles, covering Western classical music from the Baroque, Classical, and Romantic period, as well as Eastern folk melodies. Symbolic information from the score include pitch class and height of each note in addition to its onset and duration within the bar. They are accompanied by the corresponding string selection, hand position, and finger placement used by each musician when performing the piece. Additional descriptors include bar numbers and bowing, but were not used in our model as they are not always present in violin sheet music.

3.5 Implementation

Pitch class and height of each note in the dataset was converted into its corresponding MIDI number as numerical input into the model with $N_{\text{MIDI}} = 47$ to include all possible pitches on the violin (with 0 reserved for missing notes). As timing information in the dataset was based on subdividing the crotchet into 2^{10}=1024 units, we chose to discretise onsets into $N_{\text{onset}} = 2^6+2^5$=96 categories (56 are present in the dataset) and duration into $N_{\text{duration}} = 32$ (26 present) to allow for generalisation beyond excerpts in the dataset.

String selection ($\{G, D, A, E\}$), hand position ($\{1, \ldots, 12\}$), and finger placement ($\{0, \ldots, 4\}$) were combined into a single label consisting of $N_{\text{spf}} = 241$ classes (with 0 reserved for missing fingerings).

The model was trained on different numbers of excerpts (see Section 4), but always tested on one, and we report results following leave-one-out cross-validation. To maintain stylistic consistency and for comparison with previous work [7], we derived training and test data from one violinist (#2 in the dataset). However, it is important to note that violin fingerings are highly individualised and are dependent on performers' background and expert ability.

Training was implemented using batch size = 32 and optimised using Adam [24] with a learning rate of 0.01. Each excerpt was divided into sequences of length 32 for training using a hop size of 16 (i.e., half overlap), and sequences were right-padded with zeros to maintain the same length. We trained two separate models for labelled and unlabelled data simultaneously with shared layers, and oversampled the smaller dataset size to match the input sizes. Five percent of the training data was reserved for validation, and training was early-stopped [25] whenever total validation loss did not improve over 10 epochs, for which the best weights were retained.

Dense and BLSTM layers were initialised using a Glorot Uniform initialiser [26], whilst embedding layers were initialised from uniformly distributed samples. L1/L2 regularisation and L2 regularisation were respectively used for kernel and bias regularisation in the embedding and dense layers of the embedder module. In addition to kernel and bias regularisation, L2 recurrent regularisation was also used in all BLSTM layers. Finally, KL losses were weighted with $\beta = 0.001$ to improve reconstruction quality, and we set the Gumbel Softmax temperature $\tau = 0.75$.

3.6 Evaluation

We consider a variety of objective measures from information retrieval to evaluate model performance. The first is the F1 score, which we calculate using the model's most probable predicted (string, position, finger) combination. Here, we consider the F1 score as a measure for how well our model replicates the fingering style of a performer, since each note can be played with multiple fingerings.

Nevertheless, since our model predicts a probability distribution of fingerings for each note, we can also examine the position to which the true label is ranked. This provides a measure for the quality of predicted fingerings. A high, if not the highest, ranking should be assigned to the performer's chosen fingering. One metric that captures this intuition is the mean reciprocal rank (MRR) [27], given by

$$MRR = \frac{1}{N} \sum_{j=1}^{N} \frac{1}{rank^{(j)}}, \tag{7}$$

where N is the number of notes in the training excerpt and $rank^{(j)}$ denotes the rank of which the true string, position, or finger first appears for note j. Note that because we modelled the joint distribution of these three labels, $rank^{(j)}$ may exceed the number of classes in each label.

Furthermore, given the variation in fingerings used across violinists, it would be interesting to examine the preference or relevance of our model's predicted fingerings to other performers. We can capture this with a metric known as the normalised discounted cumulative gain (nDCG) [28]. First, we derive the relevance score of note j by calculating the proportion of violinists in the dataset (i.e., 10) who selected a given string, position, and finger to play the note. Next, we calculate the discounted cumulative gain (DCG) of j that is given by the sum of revelance scores for the model's top K predicted labels weighted by its log rank, i.e.

$$DCG^{(j)} = \sum_{i=1}^{K} \frac{rel_i^{(j)}}{\log_2(i+1)}, \tag{8}$$

where $rel_i^{(j)}$ denotes the relevance score of the i^{th} most probable predicted label for note j. The idealised DCG (iDCG) can also be computed by taking the DCG where the K labels are ranked from highest to lowest relevance. We can then obtain the normalised discounted cumulative gain (nDCG) at K of note j by dividing $DCG^{(j)}$ by $iDCG^{(j)}$, for which we take the mean across all notes in the testing excerpt.

4. RESULTS AND DISCUSSION

4.1 Style replication

We first consider the fully-supervised case, where our model was trained on 13 excerpts and tested on one. As shown in Figure 3 and Supplementary Table 1, our model generated violin fingerings with an MRR of 0.873 for string selection, 0.715 for hand position, and 0.721 for finger placement. These indicate that the true fingerings as performed by the violinist were predominantly given by the model's most probable predictions. Examining the confusion matrix (Figure 4) more closely, we see that the model had a tendency towards predictions in first and third position. This can also be seen when the model predicted open strings played by the performer as to be played with the second or fourth finger 32% of the time. Interestingly, the converse was not true: the model seemed to have learnt that open strings should be stylistically avoided, as second and fourth finger placements by the performer were only respectively predicted as open strings by the model 3.5% and 6.5% of the time. However, in rare cases, our model

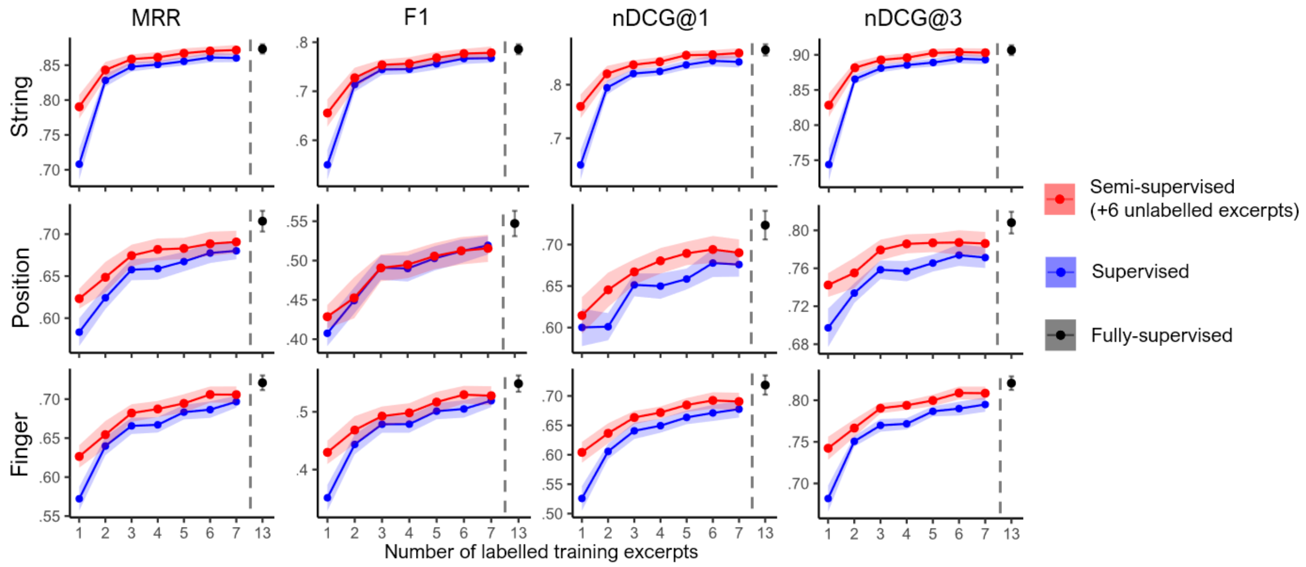


Figure 3. Examining effects of semi-supervision for different labelled training excerpt sizes. Significantly improved performance is observed when our model was trained on both labelled and unlabelled data. The fully supervised case is also shown for comparison. Filled circles and shaded regions denote mean and standard error, respectively.

failed to capture fingerings played in the 11 or 12th position. Upon inspection, we found that these notes were especially high (E7), and our model provided a fingering for the same pitch class but at an octave lower.

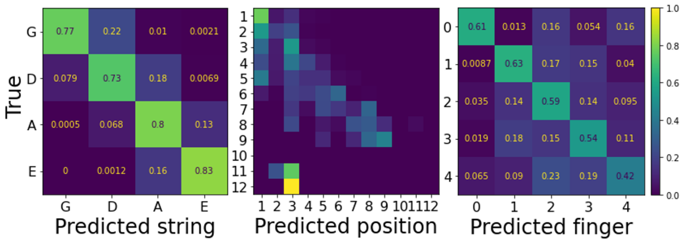


Figure 4. Confusion matrix for our fully supervised model (normalised such that each row sums to 1). Notice the tendency towards predictions in first and third position.

Regarding F1 scores, our model seemed to perform substantially better for string compared to position or finger. This is surprising since violin fingerings were modelled as the joint probability of (string, position, finger) combinations. One possible explanation could be due to the model's tendency towards predictions in first and third position. An incorrectly predicted position would have led to an incorrect finger placement even if the string was correctly predicted. Nevertheless, compared to previous work [7], our model showed noticeable improvement in F1 scores for string, position, and finger, as well as comparable performance in MRR when tested on Elgar's *Salut d'amour* [3] (see Table 1 and Figure 1).

	String		Position		Finger	
	MRR	**F1**	**MRR**	**F1**	**MRR**	**F1**
Our model (semi-supervised)						
1L+6U	.563	.291	.448	.120	.481	.147
4L+6U	.816	.714	.528	.128	.606	.277
7L+6U	.903	.834	.716	.214	.758	.500
Our model (fully-supervised)						
13L	.906	.830	.726	.305	.776	.636
Previous work (Jen et al., 2021 [7]*)*						
13L	.913	.667	.729	.241	(-)	.412

Table 1. Comparing generated fingerings to Elgar's *Salut d'amour*. Our model exceeded previous work when fully supervised, and achieved comparable performance when trained on far fewer labelled data under semi-supervision. *L* and *U* respectively denote number of labelled and unlabelled excerpts used for training.

4.2 Capturing preference across violinists

We next investigated to what extent the generated fingerings were actually performed (and thus regarded as preferred) by violinists in the dataset. The high mean nDCG@1 scores (Figure 3 and Supplementary Table 1) for string, position, and finger indicate that our model's most probable fingering predictions matched those performed by the professionals, and interestingly, also resembled the MRR scores. This suggests that the fingering patterns learnt by our model corresponded to those that showed the least variation amongst the violinists (even though it was only trained on one). Higher mean nDCG@3 scores further indicate that the stylistic variation across violinists could be adequately captured within the model's top three fingering predictions. Taken together, our evaluation measures suggest that the fingerings generated by our model matches the style and preferences of human performers.

[3] For comparison with previous work [7], we took the simple mean of F1 scores across each class instead of their class-size-weighted mean as in the rest of this paper.

4.3 Pitch and timing reconstruction

Our model was also able to reconstruct pitch and timing information with near-perfect MRR scores (all >0.945, see Supplementary Table 2). To test the extent reconstruction depended on the classifier, we replaced its output with zeros during testing. Wilcoxon signed-rank tests revealed significant differences in MRR for pitch and duration ($p = .003$ and $p = .024$, respectively, corrected using Holm's method) when information flow from the classifier to the decoder was blocked. This was associated with a 2% drop in pitch reconstruction performance, as well as a marginal 0.1% improvement in duration MRR. The former is consistent with the fact that fingerings have a direct impact on pitch, whilst the latter suggests that physical constraints might shape music as performed by humans.

Nevertheless, structured representations for pitch height and pitch class could still be seen in the encoder latent space when visualised using a uniform manifold approximation and projection (UMAP) [29] for dimension reduction (Figure 5). By contrast, the encoder latent space did not seem to separate the different fingerings (here we only show finger placement) into such clear clusters. That is expected as label information was only fed into the classifier and was thus only implicit in the encoder.

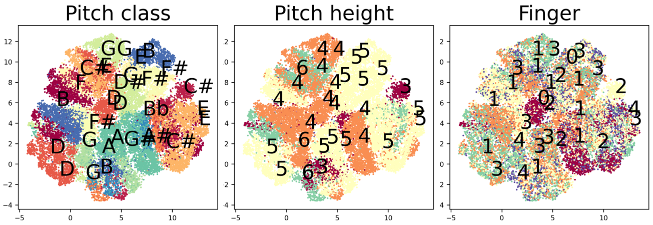


Figure 5. Visualising the encoder latent space with UMAP.

4.4 Semi-supervised learning

To investigate the effects of semi-supervised learning, we trained our model using six randomly selected excerpts as unlabelled data (i.e., without fingerings) and varied the number of labelled excerpts from one to seven. As control, we trained our model on the same labelled excerpts only. If our model could learn from unlabelled data, we would expect better generated fingerings. This is indeed what we found: our model trained on labelled and unlabelled data showed improved performance in all metrics except for the hand position F1 score (Figure 3, Supplementary Table 1). As expected, we also noticed a gradual improvement in performance as the number of labelled excerpts increased.

We further tested for significant effects of semi-supervised learning using random-intercept linear mixed models. The interaction between semi-supervision (labelled only vs. labelled and unlabelled) and number of labelled excerpts, as well as lower order terms were entered as fixed effects. Significant interactions for string MRR, nDCG@1, and nDCG@3 revealed substantial improvements ($\approx 11\%$) under semi-supervision when the ratio between labelled and unlabelled data was $1 \colon 6$. That is helpful, given our sampling (see Section 1) showed that only 15% of violin sheet music contained fingering information. Echoing the above, significant main effects of unlabelled data (with a mean improvement of around 3-6%) were also detected in MRR, nDCG@1, and nDCG@3 for string, position, and finger, as well as F1 scores for string and position (Table 2).

Finally, we note in Table 1 that our model already exceeded previous work [7] in string F1 performance when trained on four labelled plus six unlabelled excerpts, and achieved comparable performance in other metrics with seven plus six unlabelled excerpts. This demonstrates our model's ability to make use of unlabelled data to improve fingering generation performance to match the state of the art model with half the amount of labelled data.

Main effect of semi-supervised learning			
		F(1,169)	p
MRR	String	10.05	.00181 **
	Position	11.47	.000879 ***
	Finger	14.06	.000243 ***
F1	String	6.00	.0153 *
	Position	0.28	.597
	Finger	11.32	.000948 ***
nDCG@1	String	12.45	.000537 ***
	Position	7.15	.00825 **
	Finger	12.19	.000614 ***
nDCG@3	String	10.76	.00126 **
	Position	13.57	.000309 ***
	Finger	16.53	7.34×10^{-5} ***

Table 2. Linear mixed model analyses revealed significant performance improvements in all except one metric when our model was trained under semi-supervision. See Supplementary Table 3 for significance of other factors.

5. CONCLUSION AND FUTURE WORK

In this paper, we presented a semi-supervised model that generates violin fingerings from the musical score. Our approach leverages the generative ability of variational autoencoders and reframes fingering generation as an additional latent variable for learning pitch and timing reconstructions from unlabelled data. We demonstrated that our model better replicated the fingering style of a human performer and generated fingerings that were more preferred amongst violinists when trained on both labelled and unlabelled data. Our method can be readily adapted to fingering generation in other instruments such as piano and guitar, which also suffer from the same lack of labelled data [8]. Another possibility is to extend our model with heuristic rules to tailor generated fingerings for different playing styles or groups (e.g., pedagogy for violinists at different skill levels) [2,3,7]. Lastly, that pitch reconstruction depended on fingering information also highlights the importance of physical constraints and playability in music performed by humans. Such aspects are often overlooked, but should be explored in future machine-based music generation models if a more human-like quality is desired.

6. ACKNOWLEDGEMENTS

This work was partially supported by MOST Taiwan under the project *Content Generation and Interaction Systems for Virtual Musicians* (Grant No. 109-2221-E-001-018-MY3).

7. REFERENCES

[1] A. Maezawa, K. Itoyama, T. Takahashi, K. Komatani, T. Ogata, and H. G. Okuno, "Violin fingering estimation based on violin pedagogical fingering model constrained by bowed sequence estimation from audio input," in *International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems*. Springer, 2010, pp. 249–259.

[2] W. Nagata, S. Sako, and T. Kitamura, "Violin fingering estimation according to skill level based on hidden markov model." in *ICMC*, 2014.

[3] A. Maezawa, K. Itoyama, K. Komatani, T. Ogata, and H. G. Okuno, "Automated violin fingering transcription through analysis of an audio recording," *Computer Music Journal*, vol. 36, no. 3, pp. 57–72, 2012.

[4] P. Barbieri and S. Mangsen, "Violin intonation: a historical survey," *Early music*, vol. 19, no. 1, pp. 69–88, 1991.

[5] D. Huron and C. Trevor, "Are stopped strings preferred in sad music?" *Empirical Musicology Review*, vol. 11, no. 2, pp. 261–269, 2017.

[6] M. A. Winget, "Annotations on musical scores by performing musicians: Collaborative models, interactive methods, and music digital library tool development," *Journal of the American Society for Information Science and Technology*, vol. 59, no. 12, pp. 1878–1897, 2008.

[7] Y.-H. Jen, T.-P. Chen, S.-W. Sun, and L. Su, "Positioning left-hand movement in violin performance: A system and user study of fingering pattern generation," in *26th International Conference on Intelligent User Interfaces*, 2021, pp. 208–212.

[8] A. Wiggins and Y. Kim, "Guitar tablature estimation with a convolutional neural network." in *ISMIR*, 2019, pp. 284–291.

[9] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," in *CoRR*, 2014.

[10] D. J. Rezende, S. Mohamed, and D. Wierstra, "Stochastic backpropagation and approximate inference in deep generative models," in *International conference on machine learning*. PMLR, 2014, pp. 1278–1286.

[11] D. P. Kingma, S. Mohamed, D. J. Rezende, and M. Welling, "Semi-supervised learning with deep generative models," 2014.

[12] G. Hori and S. Sagayama, "Minimax viterbi algorithm for hmm-based guitar fingering decision." in *ISMIR*, 2016, pp. 448–453.

[13] S. I. Sayegh, "Fingering for string instruments with the optimum path paradigm," *Computer Music Journal*, vol. 13, no. 3, pp. 76–84, 1989.

[14] A. Al Kasimi, E. Nichols, and C. Raphael, "A simple algorithm for automatic generation of polyphonic piano fingerings": 8th international conference on music information retrieval," 2007.

[15] M. Balliauw, D. Herremans, D. Palhazi Cuervo, and K. Sörensen, "A variable neighborhood search algorithm to generate piano fingerings for polyphonic sheet music," *International Transactions in Operational Research*, vol. 24, no. 3, pp. 509–535, 2017.

[16] "Statistical learning and estimation of piano fingering," *Information Sciences*, vol. 517, pp. 68–85, 2020.

[17] A. Maezawa, K. Itoyama, T. Takahashi, T. Ogata, and H. G. Okuno, "Bowed string sequence estimation of a violin based on adaptive audio signal classification and context-dependent error correction," in *2009 11th IEEE International Symposium on Multimedia*. IEEE, 2009, pp. 9–16.

[18] S. Sako, W. Nagata, and T. Kitamura, "Violin fingering estimation according to the performer's skill level based on conditional random field," in *International Conference on Human-Computer Interaction*. Springer, 2015, pp. 485–494.

[19] C. Doersch, "Tutorial on variational autoencoders," *arXiv preprint arXiv:1606.05908*, 2016.

[20] I. Higgins, L. Matthey, A. Pal, C. Burgess, X. Glorot, M. Botvinick, S. Mohamed, and A. Lerchner, "beta-vae: Learning basic visual concepts with a constrained variational framework," 2016.

[21] C. P. Burgess, I. Higgins, A. Pal, L. Matthey, N. Watters, G. Desjardins, and A. Lerchner, "Understanding disentangling in β-vae," *arXiv preprint arXiv:1804.03599*, 2018.

[22] E. Jang, S. Gu, and B. Poole, "Categorical reparameterization with gumbel-softmax," in *ICLR*.

[23] C. J. Maddison, A. Mnih, and Y. W. Teh, "The concrete distribution: A continuous relaxation of discrete random variables."

[24] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *ICLR*, 2015.

[25] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *The journal of machine learning research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[26] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proceedings of the thirteenth international conference on artificial intelligence and statistics*. JMLR Workshop and Conference Proceedings, 2010, pp. 249–256.

[27] N. Craswell, *Mean Reciprocal Rank*. Boston, MA: Springer US, 2009, pp. 1703–1703.

[28] K. Järvelin and J. Kekäläinen, "Cumulated gain-based evaluation of ir techniques," *ACM Transactions on Information Systems (TOIS)*, vol. 20, no. 4, pp. 422–446, 2002.

[29] L. McInnes, J. Healy, and J. Melville, "Umap: Uniform manifold approximation and projection for dimension reduction," *arXiv preprint arXiv:1802.03426*, 2018.