

**Figure 1:** Overall architecture of the proposed system. The orange box includes the auxiliary tasks only for training.

performance features, and  $y = \{y_n\}_{n=1}^N$  be a sequence of the conditional score features. Our network has two *chordwise* latent variables  $z^{(\text{pln})} = \{z_c^{(\text{pln})}\}_{c=1}^C \in \mathbb{R}^{C \times d^{(\text{pln})}}$  and  $z^{(\text{str})} = \{z_c^{(\text{str})}\}_{c=1}^C \in \mathbb{R}^{C \times d^{(\text{str})}}$  that represent explicit planning and structural attribute, where  $d^{(\text{pln})}$  and  $d^{(\text{str})}$  are the sizes of  $z^{(\text{pln})}$  and  $z^{(\text{str})}$ , respectively. Our network generates notewise performance parameters  $x$  from these latent variables and given score features  $y$ . The overall architecture of our proposed system is illustrated in Figure 1.

**Generation.** A probabilistic generator parameterized by  $\theta$  produces the note-level performance parameters  $x$  from the two latent variables  $z^{(\text{pln})}$  and  $z^{(\text{str})}$  with the given condition  $y$ . We note that the latent variables are in chord-level. This decreases a computational cost and also enables intuitive modeling of polyphonic piano performance where each time step represents a stack of notes and the simultaneous notes share common characteristics [8]:

$$p_\theta(x, y, z^{(\text{pln})}, z^{(\text{str})}) = p_\theta(x | z^{(\text{pln})}, z^{(\text{str})}, y) \quad (2)$$

$$p_\theta(z^{(\text{pln})}) \prod_{c=1}^C p_\theta(z_c^{(\text{str})} | z_{<c}^{(\text{str})}, y_{\leq c}^{(\text{chd})})$$

where  $y^{(\text{chd})} = \text{N2C}(e_y)$  is the chordwise embedding, and  $e_y$  is the notewise embedding for  $y$ . We assume that the prior of  $z_c^{(\text{pln})}$  is a standard normal distribution. In contrast,  $z_c^{(\text{str})}$  is sampled from a sequential prior [24, 36, 37], conditioned on both previous latent variables and chordwise score features:  $z_c^{(\text{str})} \sim \mathcal{N}(\mu^{(\text{prior})}, \text{diag}(\sigma^{(\text{prior})^2}))$ , where  $[\mu^{(\text{prior})}, \sigma^{(\text{prior})}] = f^{(\text{prior})}(z_{<c}^{(\text{str})}, y_{\leq c}^{(\text{chd})})$ , and  $f^{(\text{prior})}$  is a unidirectional recurrent neural network. The latent representations and  $y^{(\text{chd})}$  pass through the decoder as shown in Figure 1. During training, the model predicts the intermediate chordwise output that is computed as  $\text{N2C}(x)$ . This is to enhance reconstruction power of our system, propagating accurate information of chord-level attributes to the final decoder. The intermediate activation is then extended to the notewise through the C2N operation. The note-level parameters are generated autoregressively based on this activation and the notewise score feature. We use teacher forcing during training [38].

**Inference.** A probabilistic encoder parameterized by  $\phi$  approximates the posterior distributions of the latent representations  $z^{(\text{pln})}$  and  $z^{(\text{str})}$  from the performance input  $x$

and conditional score input  $y$ :

$$q_\phi(z^{(\text{pln})}, z^{(\text{str})} | x, y) = q_\phi(z^{(\text{pln})} | x^{(\text{chd})}) \prod_{c=1}^C q_\phi(z_c^{(\text{str})} | x_{\leq c}^{(\text{chd})}, y_{\leq c}^{(\text{chd})}) \quad (3)$$

where  $x^{(\text{chd})} = \text{N2C}(e_x)$  is the chordwise embedding, and  $e_x$  is the notewise embedding for  $x$ . The posterior distributions of  $z_c^{(\text{pln})}$  and  $z_c^{(\text{str})}$  are approximated by distribution parameters encoded by  $f^{(\text{pln})}(x^{(\text{chd})})$  and  $f^{(\text{str})}(x^{(\text{chd})}, y^{(\text{chd})})$ , where  $f^{(\text{pln})}$  and  $f^{(\text{str})}$  are bidirectional and unidirectional recurrent neural networks, respectively. We note that  $z^{(\text{pln})}$  is independent of the score features  $y$ . This allows a flexible transfer of the explicit planning among other musical pieces. On the other hand,  $z^{(\text{str})}$  is constrained by  $y$  since the structural attributes are dependent on the note structure.

**Training.** We train the models  $p_\theta$  and  $q_\phi$  by approximating marginal distributions of the performance features  $x$  conditioned on the score features  $y$ . This requires to maximize negative evidence lower bound (ELBO) that includes regularization force by Kullback–Leibler divergence [34]:

$$\begin{aligned} \mathcal{L}_{\text{VAE}} = & \mathbb{E}_{q_\phi(z^{(\text{pln})}, z^{(\text{str})} | x, y)} [\log p_\theta(x | z^{(\text{pln})}, z^{(\text{str})}, y)] \\ & + \mathbb{E}_{q_\phi(z^{(\text{pln})}, z^{(\text{str})} | x, y)} [\log p_\theta(k | z^{(\text{pln})}, z^{(\text{str})}, y)] \\ & - \text{KL}(q_\phi(z^{(\text{pln})} | x) || p_\theta(z^{(\text{pln})})) \\ & - \sum_{c=1}^C \text{KL}(q_\phi(z_c^{(\text{str})} | x_{\leq c}^{(\text{chd})}, y_{\leq c}^{(\text{chd})}) || p_\theta(z_c^{(\text{str})} | z_{<c}^{(\text{str})}, y_{\leq c}^{(\text{chd})})) \end{aligned} \quad (4)$$

where  $k = \text{N2C}(x)$  is the chordwise performance features.

## 2.4 Regularizing the Latent Variables

We enhance disentanglement of the latent representations  $z^{(\text{pln})}$  and  $z^{(\text{str})}$  using four regularization tasks [24].

**Prediction Tasks.** We extract new supervisory signals for additional prediction tasks from the input data [24]. We define a signal of explicit planning  $I^{(\text{pln})}$  as a set of smoothed contours of the expressive parameters. It is extracted as a polynomial function predicted from the chordwise performance parameters  $k$ . We also derive a signal of structural attribute as  $I^{(\text{str})} = \text{sign}(k - I^{(\text{pln})})$  which represents normalized directions of the performance parameters. We train two discriminators  $D^{(\text{pln})}$  and  $D^{(\text{str})}$  that directly receive  $z^{(\text{pln})}$  and  $z^{(\text{str})}$ , respectively.  $D^{(\text{pln})}$  is composed of  $A$  sub-discriminators where each discriminator  $D_a^{(\text{pln})}$  predicts a signal  $I_a^{(\text{pln})}$  for each expressive attribute  $a$  from  $z_a^{(\text{pln})} \in \mathbb{R}^{C \times (d^{(\text{pln})}/A)}$ , where  $z_a^{(\text{pln})}$  is a constituent part of  $z^{(\text{pln})}$ , and  $A$  is the number of expressive attributes. This setting is for a clear disentanglement among the expressive attributes. On the other hand,  $D^{(\text{str})}$  predicts the signal  $I^{(\text{str})}$  at once for all expressive attributes that belong to the same musical structure. All discriminators are jointly trained with the generative model, and the costs  $\mathcal{L}_{\text{pln}}$  and  $\mathcal{L}_{\text{str}}$  are minimized

as  $\mathcal{L}_{\text{pln}} = \frac{1}{A} \sum_a \text{MSE}(D_a^{(\text{pln})}(z_a^{(\text{pln})}), I_a^{(\text{pln})})$  and  $\mathcal{L}_{\text{str}} = \text{MSE}(D^{(\text{str})}(z^{(\text{str})}), I^{(\text{str})})$ , respectively.

**Factorizing Latent Variables.** We further constrain a generator to guarantee that  $z^{(\text{pln})}$  delivers correct information regardless of  $z^{(\text{str})}$  [39]. During training, we sample a new output  $\tilde{x}$  using  $z^{(\text{pln})} \sim q_\phi(z^{(\text{pln})}|x)$  and  $\tilde{z}^{(\text{str})} \sim p_\theta(z^{(\text{str})})$ . Then, we re-infer  $\tilde{z}^{(\text{pln})} \sim q_\phi(\tilde{z}^{(\text{pln})}|\tilde{x})$  to estimate the supervisory signal  $I^{(\text{pln})}$ . This prediction loss is backpropagated only through the generator:

$$\mathcal{L}_{\text{fac}} = \frac{1}{A} \sum_a \text{MSE}(D_a^{(\text{pln})}(\tilde{z}_a^{(\text{pln})}), I_a^{(\text{pln})}) \quad (5)$$

**Aligning Latent Variables with Factors.** Finally, we enable the "sliding-fader" control of the expressive attributes [28]. To this end, we employ the regularization loss proposed by Pati *et al.* [27] that aligns specific dimensions of  $z^{(\text{pln})}$  with the target expressive attributes. This method assumes that a latent representation can be disentangled through its monotonic relationship with a target attribute. Let  $d_i$  and  $d_j$  be a target dimension  $d$  of  $i$ th and  $j$ th latent representations, respectively, where  $d \in z_a^{(\text{pln})}$ ,  $i, j \in [1, M]$ , and  $M$  is the size of a mini-batch. A distance matrix  $\mathcal{D}_d$  is computed between  $d_i$  and  $d_j$  within a mini-batch, where  $\mathcal{D}_d = d_i - d_j$ . A similar distance matrix  $\mathcal{D}_a$  is computed for the two target attribute values  $a_i$  and  $a_j$ . We minimize a MSE between  $\mathcal{D}_d$  and  $\mathcal{D}_a$  as follows:

$$\mathcal{L}_{\text{reg}} = \text{MSE}(\tanh(\mathcal{D}_d), \text{sign}(\mathcal{D}_a)) \quad (6)$$

## 2.5 Overall Objective

The overall objective of our proposed network aims to generate realistic performance features with properly disentangled representations for the intended factors:

$$\mathcal{L} = \mathcal{L}_{\text{VAE}} + \lambda_{\text{pln}} \mathcal{L}_{\text{pln}} + \lambda_{\text{str}} \mathcal{L}_{\text{str}} + \lambda_{\text{fac}} \mathcal{L}_{\text{fac}} + \lambda_{\text{reg}} \mathcal{L}_{\text{reg}} \quad (7)$$

where  $\lambda_{\text{pln}}$ ,  $\lambda_{\text{str}}$ ,  $\lambda_{\text{fac}}$ , and  $\lambda_{\text{reg}}$  are hyperparameters for balancing the importance of the loss terms.

## 3. EXPERIMENTAL SETUPS

### 3.1 Dataset and Implementation

We use Yamaha e-Competition Dataset [8] and Vienna 4x22 Piano Corpus [40]. From these datasets, we collect 356 performances of 34 pieces by Frédéric Chopin, which have been representative research subjects for analyzing the Western musical expression [6, 22, 41, 42]. We use 30 pieces (108,738 batches) for training and the rest for testing. To verify the generality of model performances, we also collect the external dataset from ASAP dataset [43]. We use 116 performances for 23 pieces by 10 composers who represent various eras of Western music. For subjective evaluation, we collect 42 songs of non-Classical songs from online source<sup>2</sup> which are less constrained to written expression than most Classical excerpts.

<sup>2</sup> <http://www.ambrosepianotabs.com/page/library>

We basically follow Jeong *et al.* [8] to compute the input features from the aligned pairs of performance and score data. We set MIDI velocities and Beat Per Minute (BPM) of all notes in the score data to be 64 and 120, respectively. We also remove any grace notes for simplicity and manually correct any errors. The performance features are further normalized into a range from -1 to 1 for training. We use an ADAM optimizer [44] with an initial learning rate of 1e-5, which is reduced by 5% every epoch during backpropagation. We empirically set  $\lambda_{\text{pln}}$ ,  $\lambda_{\text{str}}$ ,  $\lambda_{\text{fac}}$ , and  $\lambda_{\text{reg}}$  to be 1000, 100, 1, 10, respectively. We set a degree of the polynomial function computing  $I^{(\text{pln})}$  as 4 through an ablation study described in the supplementary material.

### 3.2 Comparative Methods

To the best of our knowledge, there is no existing method that does not intentionally follow the written guidelines in the musical score. Therefore, we use variants of our proposed network as comparing methods that differ in model architecture: **Notewise** denotes the proposed model without the hierarchical learning. **CVAE** denotes a variant of Notewise where  $z^{(\text{pln})}$  is substituted with the supervisory signal  $I^{(\text{pln})}$ . We also conduct an ablation study that investigates necessity of the four loss terms.

## 4. EVALUATION

We evaluate the proposed network in terms of both objective and subjective criteria.

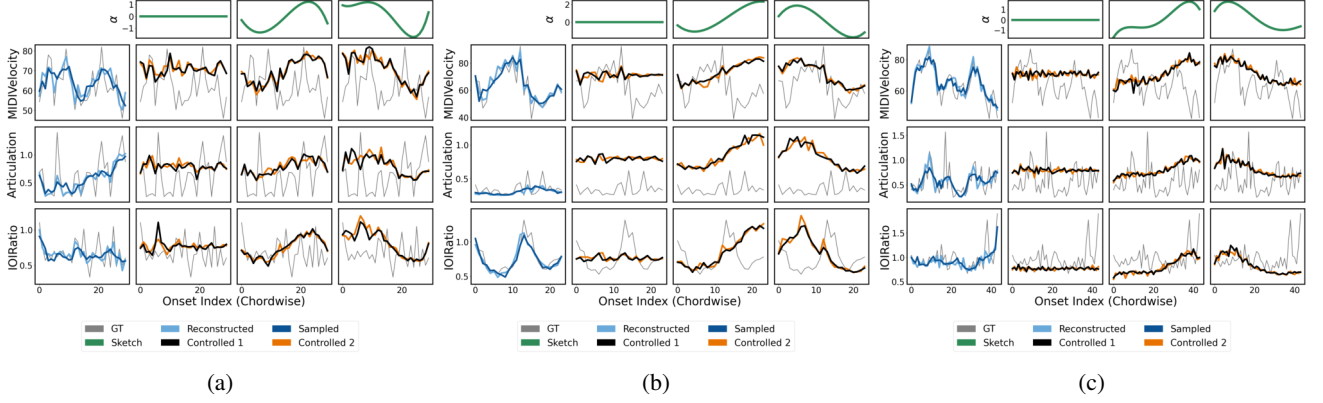
### 4.1 Generation Quality

We compute Pearson's correlation coefficients between the reconstructed or generated samples and human piano performances [6, 9, 11, 19]. We first measure the reconstruction quality of the test samples (" $R_{\text{recon}}$ "). Then, we evaluate the samples generated from  $\tilde{z}^{(\text{str})} \sim p_\theta(z^{(\text{str})})$  and either of: 1)  $z^{(\text{pln})} \sim q_\phi(z^{(\text{pln})}|x)$  (" $R_{x|\text{pln}}$ ") and 2)  $z_0^{(\text{pln})} \sim q_\phi(z_0^{(\text{pln})}|x_0)$  (" $R_{x|\text{pln}_0}$ "), where  $x_0$  is a zero matrix.

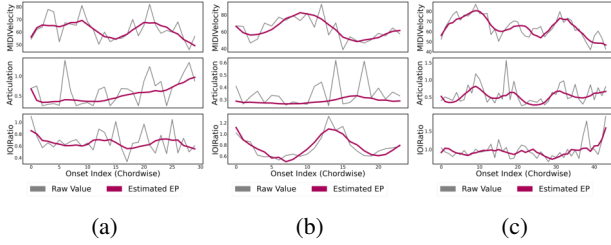
The results are shown in Table 1. Notewise shows the best scores in both datasets, and our method outperforms CVAE in  $R_{\text{recon}}$ . It indicates that our proposed architecture where a latent representation is used instead of a direct condition is generally good at reconstructing the human data. When using the randomly sampled  $\tilde{z}^{(\text{str})}$ , our method and the model without  $\mathcal{L}_{\text{reg}}$  show stable scores compared to other baseline models. The model without  $\mathcal{L}_{\text{reg}}$  also shows the highest scores in  $R_{x|\text{pln}}$  for both datasets. It indicates that  $\mathcal{L}_{\text{reg}}$  may contribute the least to generation power among other loss terms. CVAE and the model only with  $\mathcal{L}^{(\text{pln})}$  also show high scores in  $R_{x|\text{pln}_0}$ . This may be due to the posterior collapse that makes the decoder depends mostly on the score condition [45], which is demonstrated in the supplementary material.

### 4.2 Disentangling Latent Representations

We verify whether the latent representations are well-disentangled by appropriate information [24]. To this



**Figure 2:** Qualitative samples for the proposed system. Light-blue, blue and gray lines denote the reconstructed results, sampled results from the inferred  $z^{(\text{pln})}$ , and their ground truths, respectively; black and orange lines denote the controlled results that are generated from different random  $\tilde{z}^{(\text{str})}$ ; and green lines denote the "sketch" values, or  $\alpha$ , that are inserted to  $z^{(\text{pln})}$ . The samples demonstrate three excerpts that are: (a) Haydn's Keyboard Sonata, Hob. XVI:39, 3rd movement, mm. 53-56; (b) Schubert's Impromptu, Op. 90, No. 4, mm. 149-152; and (c) Balakirev's Islamey, Op. 18, mm. 29-32.



**Figure 3:** Qualitative results for estimating the explicit planning from raw piano performances. Pink and gray lines denote the estimated contours and raw performance parameters, respectively. The results in (a), (b), and (c) are from the same excerpts for (a), (b), and (c) in Figure 2, respectively.

Dataset	Internal			External		
Metric	$R_{\text{recon}}$	$R_{x \text{pln}}$	$R_{x \text{pln}_0}$	$R_{\text{recon}}$	$R_{x \text{pln}}$	$R_{x \text{pln}_0}$
Notewise	<b>0.870</b>	0.392	0.203	<b>0.875</b>	0.479	0.177
CVAE	0.730	0.338	0.223	0.741	0.399	0.216
$\mathcal{L}_{\text{pln}}$	0.627	0.357	0.229	0.687	0.414	<b>0.220</b>
$\mathcal{L}_{\text{pln}} + \mathcal{L}_{\text{str}}$	0.770	0.325	0.181	0.837	0.398	0.195
w/o $\mathcal{L}_{\text{fac}}$	0.774	0.289	0.176	0.838	0.354	0.173
w/o $\mathcal{L}_{\text{reg}}$	0.737	<b>0.437</b>	0.224	0.793	<b>0.502</b>	0.216
Ours	0.737	0.427	<b>0.231</b>	0.789	0.498	0.203

**Table 1:** Evaluation results for the generation quality. The higher score is the better.

end, each model infers the latent representations  $z^{(\text{pln})}$  and  $z^{(\text{str})}$  from the test sets. Each model also randomly samples  $\tilde{z}^{(\text{str})}$  and infers  $z_0^{(\text{pln})} \sim q_\phi(z^{(\text{pln})}|x_0)$ . We use  $z_0^{(\text{pln})}$  to measure the structural attribute, since  $z_0^{(\text{pln})}$  represents a flat expression where the structural attribute can be solely exposed. Each model generates new outputs as  $x^{(\text{pln})} \sim p_\theta(x^{(\text{pln})}|z^{(\text{pln})}, \tilde{z}^{(\text{str})}, y)$  and  $x^{(\text{str})} \sim p_\theta(x^{(\text{str})}|z_0^{(\text{pln})}, z^{(\text{str})}, y)$ . Then, we compute a new signal  $\tilde{I}^{(\text{pln})}$  from  $x^{(\text{pln})}$  using the polynomial regression. The MSE values are calculated as  $\text{MSE}_p = \text{MSE}(\tilde{I}^{(\text{pln})}, I^{(\text{pln})})$  and  $\text{MSE}_s = \text{MSE}(x^{(\text{str})}, k - I^{(\text{pln})})$ .

Dataset	Internal		External	
Metric	$\text{MSE}_p$	$\text{MSE}_s$	$\text{MSE}_p$	$\text{MSE}_s$
Notewise	0.003	0.006	0.022	0.028
CVAE	0.034	0.045	0.085	0.092
$\mathcal{L}_{\text{pln}}$	0.028	0.036	0.074	0.077
$\mathcal{L}_{\text{pln}} + \mathcal{L}_{\text{str}}$	0.012	0.015	0.022	0.027
w/o $\mathcal{L}_{\text{fac}}$	0.018	0.023	0.021	0.025
w/o $\mathcal{L}_{\text{reg}}$	0.002	0.004	0.014	0.022
Ours	<b>0.001</b>	<b>0.002</b>	<b>0.012</b>	<b>0.020</b>

**Table 2:** Evaluation results for the disentanglement of the latent representations.

Dataset	Internal			External		
Metric	C	R	L	C	R	L
Notewise	0.782	0.916	0.632	0.775	0.914	0.656
CVAE	0.798	0.812	0.620	0.773	0.802	0.649
$\mathcal{L}_{\text{pln}}$	0.693	0.852	0.323	0.694	0.834	0.324
$\mathcal{L}_{\text{pln}} + \mathcal{L}_{\text{str}}$	0.633	0.882	0.253	0.639	0.865	0.277
w/o $\mathcal{L}_{\text{fac}}$	0.831	0.846	0.789	0.832	0.831	0.847
w/o $\mathcal{L}_{\text{reg}}$	0.804	<b>0.955</b>	0.653	0.808	<b>0.946</b>	0.657
Ours	<b>0.942</b>	0.953	<b>0.976</b>	<b>0.944</b>	0.945	<b>0.977</b>

**Table 3:** Evaluation results for the controllability of the expressive attributes. C, R, and L denotes consistency, restrictiveness, and linearity, respectively. Each score is the average score for the expressive attributes.

Table 2 shows that our method achieves the best scores in all metrics for both datasets. This confirms that our proposed system can learn the latent representations that reflect the intended attributes. Notewise and the model without  $\mathcal{L}_{\text{reg}}$  also show the robust scores compared to other baseline models. It indicates that using the notewise modeling alone is still relevant for achieving appropriate representations. It also implies that  $\mathcal{L}_{\text{reg}}$  may not contribute to the disentanglement as much as other loss terms.

### 4.3 Controllability of Expressive Attributes

We sample a new input  $\tilde{x}$  where entries of each feature are constant across time. Then, each model infers

Metric	Winning Rate (Human-likeness)		
	T	UT	Overall
Notewise	0.317( $\pm 0.223$ )	0.541( $\pm 0.316$ )	0.493( $\pm 0.309$ )
CVAE	<b>0.467(<math>\pm 0.356</math>)</b>	0.477( $\pm 0.342$ )	0.475( $\pm 0.338$ )
Ours	0.417( $\pm 0.256$ )	<b>0.555(<math>\pm 0.256</math>)</b>	<b>0.525(<math>\pm 0.258</math>)</b>

**Table 4:** Evaluation results for the winning rate in terms of human-likeness. T, UT, and Overall denote musically trained, untrained, and all groups, respectively.

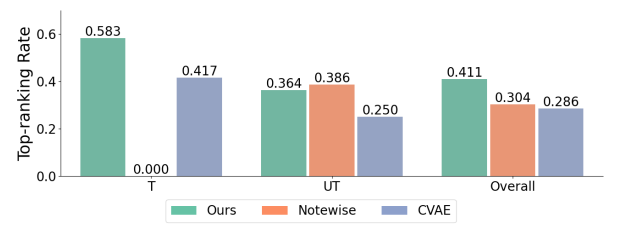
$\bar{z}^{(\text{pln})} \sim q_{\phi}(\bar{z}^{(\text{pln})}|\bar{x})$ . We control each attribute by varying dimension values of  $\bar{z}^{(\text{pln})}$  following Tan *et al.* [28] and examine the new samples generated from  $\bar{z}^{(\text{pln})}$ . We leverage the existing metrics to measure the controllability of each model [28]: *Consistency* ("C") measures consistency across samples in terms of their controlled attributes; *restrictiveness* ("R") measures how much the uncontrolled attributes maintain their flatness over time; and *linearity* ("L") measures how much the controlled attributes are correlated with the corresponding latent dimensions. We average over the three expressive attributes—dynamics, articulation, and tempo—into one score for each metric.

Table 3 demonstrates that our system shows the best scores in consistency and linearity in both internal and external datasets. This indicates that our proposed method can robustly control the latent representation  $z^{(\text{pln})}$  in intended way. The model without  $\mathcal{L}_{\text{reg}}$  outperforms our method in restrictiveness. It indicates that the uncontrolled attributes by this model are the least interfered by the controlled attribute. However, its scores on consistency and linearity are lower than ours. It confirms that  $\mathcal{L}_{\text{reg}}$  promotes linear control of the target attributes.

#### 4.4 Subjective Evaluation

We conduct a listening test to compare the proposed model architecture to Notewise and CVAE. We qualitatively evaluate the base quality of the samples that have flat expressions, so that quality judgments are independent of any preference of arbitrary explicit planning. We generate each sample using  $z_0^{(\text{pln})}$ . A listening test is composed of 30 trials where each participant chooses a more "human-like" sample out of the generated sample and its plain MIDI [9]. Both samples have the same length which is a maximum of 15 seconds, rendered with TiMidity++<sup>3</sup> without any pedal effect. *Human-likeness* denotes how similar the sample is to an actual piano performance that commonly appears in popular music. A total of 28 participants are involved, and 6 participants are professionally trained in music.

The results are demonstrated in Table 4. We measure a *winning rate*, a rate of winning over the plain MIDI, and a *top-ranking rate*, a rate of being the highest rank among the three models in terms of winning rate. These metrics are further explained in the supplementary material. The results show that musically *trained* ("T") and *untrained* ("UT") groups show the different tendency of each other: in the trained group, CVAE shows the best winning rate, and our method gets the best top-ranking rate; in the



**Figure 4:** Evaluation results for the top-ranking rate. T, UT, and Overall denote musically trained, untrained, and all groups, respectively.

untrained group, our method shows the highest winning rate, whereas Notewise is top-ranked most frequently. We note that our system reveals smaller variances than those of CVAE and Notewise of the musically trained and untrained groups in the winning rate, respectively. Moreover, our system receives the highest overall scores for both metrics. It indicates that our system can be stably perceived more human-like than the plain MIDI compared to other baseline models.

#### 4.5 Qualitative Examples

Our system can render new piano performances from the scratch given a musical score. It can directly generate expressive parameters from the randomly sampled  $\bar{z}^{(\text{pln})} \sim p_{\theta}(z^{(\text{pln})})$  and  $\bar{z}^{(\text{str})} \sim p_{\theta}(z^{(\text{str})})$ . We note that  $\bar{z}^{(\text{pln})}$  does not have temporal dependency: each  $\bar{z}_c^{(\text{pln})}$  is sampled independently of  $\bar{z}_{c-1}^{(\text{pln})}$ . Hence, we need to insert specific values  $\{\alpha^{(c)}\}_{c=1}^C$ , which we call as "smooth sketches", into the target dimensions of  $z^{(\text{pln})}$  if any temporal dependency of explicit planning is necessary. Figure 2 shows that the controlled parameters are greatly correlated with  $\alpha$ , while their local characteristics follow those of the ground truth. In addition, the black and orange lines together demonstrate granular variety in the parameters induced by different  $\bar{z}^{(\text{str})}$  for the same musical structure. Moreover, Figure 3 shows that our system can estimate explicit planning from arbitrary human performances, indicating that our system can derive relevant information on explicit planning from the unseen data.

### 5. CONCLUSION

We propose a system that can render expressive piano performance with flexible control of musical expression. We attempt to achieve representations for the explicit planning and structural attribute through self-supervised learning objectives. We also leverage the two-step modeling of two hierarchical units for an intuitive generation. Experimental results confirm that our system shows stable generation quality, disentangles the target representations, and controls all expressive attributes independently of each other. Future work can be improving our system using a larger dataset for various genres and composers. We can also further compare our system with recent piano-rendering models [8] to investigate any connections between a performer's explicit planning and a composer's intent.

<sup>3</sup> <https://sourceforge.net/projects/timidity/>

## 6. ACKNOWLEDGMENTS

We deeply appreciate Dasaem Jeong, Taegyun Kwon, and Juhan Nam for giving technical support to initiate this research. We also especially appreciate Hyeong-Seok Choi for providing critical feedback on the model architecture and evaluation. We greatly appreciate You Jin Choi and all of my colleagues who gave great help with respect to the listening test.

## 7. REFERENCES

- [1] G. Widmer and W. Goebel, "Computational models of expressive music performance: The state of the art," *Journal of New Music Research*, vol. 33, no. 3, pp. 203–216, 2004.
- [2] C. E. Cancino-Chacón, M. Grachten, W. Goebel, and G. Widmer, "Computational models of expressive music performance: A comprehensive and critical review," *Frontiers in Digital Humanities*, vol. 5, no. 25, pp. 1–23, 2018.
- [3] G. Widmer, S. Flossmann, and M. Grachten, "YQX plays Chopin," *AI Magazine*, vol. 30, no. 3, pp. 35–48, 2009.
- [4] T. H. Kim, S. Fukayama, T. Nishimoto, and S. Sagayama, "Statistical approach to automatic expressive rendition of polyphonic piano music," in *Guide to Computing for Expressive Music Performance*. Springer, 2013, pp. 145–179.
- [5] C. E. Cancino-Chacón and M. Grachten, "An evaluation of score descriptors combined with non-linear models of expressive dynamics in music," in *Proceedings of the International Conference on Discovery Science*, 2015.
- [6] C. E. Cancino-Chacón, T. Gadermaier, G. Widmer, and M. Grachten, "An evaluation of linear and non-linear models of expressive dynamics in classical piano and symphonic music," *Machine Learning*, vol. 106, no. 6, pp. 887–909, 2017.
- [7] A. Maezawa, "Deep piano performance rendering with conditional VAE," in *Late-Breaking Demo, the 19th International Society for Music Information Retrieval Conference*, 2018.
- [8] D. Jeong, T. Kwon, Y. Kim, K. Lee, and J. Nam, "VirtuosoNet: A hierarchical RNN-based system for modeling expressive piano performance," in *Proceedings of the 20th International Society for Music Information Retrieval Conference*, 2019.
- [9] D. Jeong, T. Kwon, Y. Kim, and J. Nam, "Graph neural network for music score data and modeling expressive piano performance," in *Proceedings of the 36th International Conference on Machine Learning*, 2019.
- [10] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 8, 2013.
- [11] A. Maezawa, K. Yamamoto, and T. Fujishima, "Rendering music performance with interpretation variations using conditional variational RNN," in *Proceedings of the 20th International Society for Music Information Retrieval Conference*, 2019.
- [12] H. H. Tan, Y.-J. Luo, and D. Herremans, "Generative modeling for controllable audio synthesis of expressive piano performance," in *Proceedings of the 37th International Conference on Machine Learning*, 2020.
- [13] R. Bresin and A. Friberg, "Emotional coloring of computer-controlled music performances," *Computer Music Journal*, vol. 24, no. 4, 2000.
- [14] S. R. Livingstone, R. Muhlberger, A. R. Brown, and W. F. Thompson, "Changing musical emotion: A computational rule system for modifying score and performance," *Computer Music Journal*, vol. 34, no. 1, 2010.
- [15] M. Bernays and C. Traube, "Investigating pianists' individuality in the performance of five timbral nuances through patterns of articulation, touch, dynamics, and pedaling," *Frontiers in Psychology*, vol. 5, no. 157, pp. 1–19, 2014.
- [16] S. Oore, I. Simon, S. Dieleman, D. Eck, and K. Simonyan, "This time with feeling: Learning expressive musical performance," *Neural Computing and Applications*, vol. 32, pp. 955–967, 2020.
- [17] A. Bhatara, A. K. Tirovolas, L. M. Duan, B. Levy, and D. J. Levitin, "Perception of emotional expression in musical performance," *Journal of Experimental Psychology: Human Perception and Performance*, vol. 37, no. 3, pp. 921–934, 2011.
- [18] A. Friberg, R. Bresin, and J. Sundberg, "Overview of the KTH rule system for musical performance," *Advances in Cognitive Psychology*, vol. 2, no. 2-3, pp. 145–161, 2006.
- [19] S. Flossmann, M. Grachten, and G. Widmer, "Expressive performance rendering with probabilistic models," in *Guide to Computing for Expressive Music Performance*. Springer, 2013, pp. 75–98.
- [20] R. H. Woody, "The relationship between explicit planning and expressive performance of dynamic variations in an aural modeling task," *Journal of Research in Music Education*, vol. 47, no. 4, pp. 331–342, 1999.
- [21] A. Lerch, C. Arthur, A. Pati, and S. Gururani, "Music performance analysis: A survey," in *Proceedings of the 20st International Society for Music Information Retrieval Conference*, 2019.



- [22] B. H. Repp, “A microcosm of musical expression: II. quantitative analysis of pianists’ dynamics in the initial measures of Chopin’s Etude in E major,” *The Journal of the Acoustical Society of America*, vol. 105, no. 3, pp. 1972–1988, 1999.
- [23] H. Honing, “From time to time: The representation of timing and tempo,” *Computer Music Journal*, vol. 25, no. 3, 2001.
- [24] Y. Zhu, M. R. Min, A. Kadav, and H. P. Graf, “S3VAE: Self-supervised sequential VAE for representation disentanglement and data generation,” in *Proceedings of Computer Vision and Pattern Recognition*, 2020.
- [25] F. Locatello, S. Bauer, M. Lucic, G. Rätsch, S. Gelly, B. Schölkopf, and O. Bachem, “Challenging common assumptions in the unsupervised learning of disentangled representations,” in *Proceedings of the 36th International Conference on Machine Learning*, 2019.
- [26] D. Hendrycks, M. Mazeika, S. Kadavath, and D. Song, “Using self-supervised learning can improve model robustness and uncertainty,” in *Proceedings of the 33rd Conference on Neural Information Processing Systems*, 2019.
- [27] A. Pati and A. Lerch, “Attribute-based regularization of latent spaces for variational auto-encoders,” in *Neural Computing and Applications*, 2020.
- [28] H. H. Tan and D. Herremans, “Music FaderNets: Controllable music generation based on high-level features via low-level feature modeling,” in *Proceedings of the 21st International Society for Music Information Retrieval Conference*, 2020.
- [29] D. Jeong, T. Kwon, Y. Kim, and J. Nam, “Score and performance features for rendering expressive music performances,” in *Proceedings of the Music Encoding Conference*, 2019.
- [30] A. Roberts, J. Engel, and D. Eck, “Hierarchical variational autoencoders for music,” in *Proceedings of the 31st Conference on Neural Information Processing Systems*, 2017.
- [31] H.-W. Dong, W.-Y. Hsiao, L.-C. Yang, and Y.-H. Yang, “MuseGAN: Multi-track sequential generative adversarial networks for symbolic music generation and accompaniment,” in *Proceedings of the 32nd AAAI Conference on Artificial Intelligence*, 2018.
- [32] S.-L. Wu and Y.-H. Yang, “MuseMorphose: Full-song and fine-grained music style transfer with just one Transformer VAE,” *arXiv preprint arXiv:2105.04090*, 2021.
- [33] Z. Wang, Y. Zhang, Y. Zhang, J. Jiang, R. Yang, J. Z. (Jake), and G. Xia, “Pianotree VAE: Structured representation learning for polyphonic music,” in *Proceedings of the 37th International Conference on Machine Learning*, 2020.
- [34] D. P. Kingma and M. Welling, “Auto-encoding variational bayes,” *arXiv preprint arXiv:1312.6114*, 2013.
- [35] K. Sohn, X. Yan, and H. Lee, “Learning structured output representation using deep conditional generative models,” in *Proceedings of the 28th International Conference on Neural Information Processing Systems*, 2015.
- [36] Y. Li and S. Mandt, “Disentangled sequential autoencoder,” in *Proceedings of the 35th International Conference on Machine Learning*, 2018.
- [37] J. Chung, K. Kastner, L. Dinh, K. Goel, A. C. Courville, and Y. Bengio, “A recurrent latent variable model for sequential data,” in *Proceedings of the 28th International Conference on Neural Information Processing Systems*, 2015.
- [38] R. J. Williams and D. Zipser, “A learning algorithm for continually running fully recurrent neural networks,” *Neural Computation*, vol. 1, no. 2, 1989.
- [39] Z. Hu, Z. Yang, X. Liang, R. Salakhutdinov, and E. P. Xing, “Toward controlled generation of text,” in *Proceedings of the 34th International Conference on Machine Learning*, 2017.
- [40] W. Goebel, “Melody lead in piano performance: Expressive device or artifact?” *The Journal of the Acoustical Society of America*, vol. 110, no. 1, 2001.
- [41] M. Grachten and G. Widmer, “Linear basis models for prediction and analysis of musical expression,” *Journal of New Music Research*, vol. 41, no. 4, pp. 311–322, 2012.
- [42] Z. Shi, “Computational analysis and modeling of expressive timing in Chopin Mazurkas,” in *Proceedings of the 22nd International Society for Music Information Retrieval Conference*, 2021.
- [43] F. Foscarin, A. McLeod, P. Rigaux, F. Jacquemard, and M. Sakai, “ASAP: A dataset of aligned scores and performances for piano transcription,” in *Proceedings of the 21st International Society for Music Information Retrieval Conference*, 2020.
- [44] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *arXiv preprint arXiv:1412.6980*, 2014.
- [45] 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,” in *Proceedings of the 5th International Conference on Learning Representations*, 2017.

# EXPLOITING DEVICE AND AUDIO DATA TO TAG MUSIC WITH USER-AWARE LISTENING CONTEXTS

Karim M. Ibrahim<sup>1,2</sup>

Elena V. Epure<sup>2</sup>

Geoffroy Peeters<sup>1</sup>

Gaël Richard<sup>1</sup>

<sup>1</sup> LTCI, Télécom Paris, Institut Polytechnique de Paris

<sup>2</sup> Deezer Research

karim.ibrahim@telecom-paris.fr

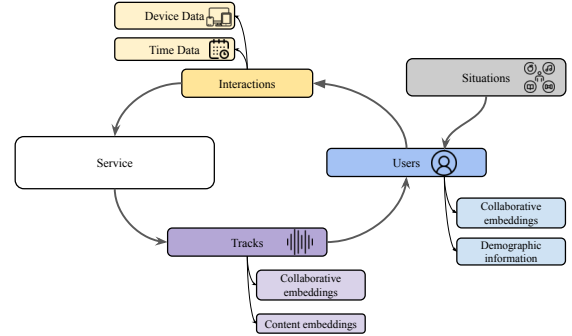
## ABSTRACT

As music has become more available especially on music streaming platforms, people have started to have distinct preferences to fit to their varying listening situations, also known as context. Hence, there has been a growing interest in considering the user's situation when recommending music to users. Previous works have proposed user-aware autotaggers to infer situation-related tags from music content and user's global listening preferences. However, in a practical music retrieval system, the autotagger could be only used by assuming that the context class is explicitly provided by the user. In this work, for designing a fully automatised music retrieval system, we propose to disambiguate the user's listening information from their stream data. Namely, we propose a system which can generate a situational playlist for a user at a certain time 1) by leveraging user-aware music autotaggers, and 2) by automatically inferring the user's situation from stream data (e.g. device, network) and user's general profile information (e.g. age). Experiments show that such a context-aware personalized music retrieval system is feasible, but the performance decreases in the case of new users, new tracks or when the number of context classes increases.

## 1. INTRODUCTION

Since the invention of recorded music, people have been shifting from consuming music as a main activity in a live setting, to using music as a background activity as they go through the day. With the growing availability of music on streaming platforms, people developed distinct preferences for the varying listening situations, also known as context [1]. Consequently, there has been a growing interest in considering the user's situation when automatically recommending music to users.

Previous works have proposed user-aware autotaggers to infer situation-related tags from music content and user's global listening preferences [2]. However, in a practical music retrieval system, the autotagger could be only



**Figure 1.** The available data to online music streaming services.

used by assuming that the context class is explicitly provided by the user. In this work, we perform a study to evaluate the feasibility of inferring the listening situation. The listening situation for our system is an activity, location, or time that is influencing the listener's preferences.

The process of music streaming from the perspective of our proposed approach can be found in Figure 1. We find that the music service is informed of the users, their track history, plus their past and current interactions with the service, i.e. the device and time data sent during an active session. However, the service is unaware of the influencing listening situation. Our goal is to utilize the available information for the service to infer the listening situation and the suitable tracks for the inferred situation. We propose an approach that infers the potential context from the user interactions in near real time, while the tagging of tracks with their potential listening situation happens in the background using autotaggers. Both systems are user-aware.

Our contributions in this paper are: 1) a large dataset of tracks, device data, and user embeddings labeled with their situational use through a rigorous labelling pipeline; 2) an extended evaluation of music autotaggers in predicting personalized situational tags in various scenarios; 3) a simple, yet effective model that ranks the potential listening situations for a given user based on the transmitted data from the device to the service.

## 2. RELATED WORK

Our proposed approach is related to two different problems: music autotagging with contextual tags, and instant



© F. Author, S. Author, and T. Author. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0).  
**Attribution:** F. Author, S. Author, and T. Author, "Exploiting device and audio data to tag music with user-aware listening contexts", in *Proc. of the 23rd Int. Society for Music Information Retrieval Conf.*, Bengaluru, India, 2022.

prediction of the user’s listening situation. Previous work has already showed that listening situation (i.e. context) has a strong influence on the user’s preferences [1, 3–5]. Hence, context has become an important factor for reaching a personalized user experience [6].

On one hand, music content is highly complex and is often challenging to be analyzed and described in human readable terms. This missing link between the content of the music and a set of semantic descriptors is referred to as the “semantic gap” [7]. One common way, which is often used when searching for or organising music, is the intended listening situation [8]. Unlike most tags that depend solely on the music content, certain tags depend also on the user [9, 10]. There has been a recent work on predicting personalized situation-related tags from music content and user embeddings [2], which we adopt here too.

On the other hand, the listening situation, e.g. activity or location, can change frequently, which leads to changes in user preferences. Explicitly inferring the listening situation is a challenging task that has only been studied on a small scale [11]. We aim at addressing this missing link by performing an extensive study on predicting the listening situation using available device data. In order to employ the personalized autotagging approach in an actual real-world setting, it is also important to be able to predict when a specific listening situation is being experienced.

### 3. OBJECTIVE AND PROPOSED APPROACH

A *session* consists of a sequence of audio-tracks  $a$  a given user  $u$  is listening to over time  $t$  on a music streaming service in a continuous time span<sup>1</sup>. A session is therefore defined as a sequence of *streams* which are each a tuple (audio-track  $a$ , user  $u$ , device data  $\mathbf{d}_u^{(t)}$ ).

A *situational (or contextual) session* is a session resulting from listening to tracks in a certain situation (or context)  $c$  such as “gym”. However, in our case, in order to gather a ground-truth dataset, we consider that a situational session can also result from listening to a playlist that contains a context-related keyword in the title<sup>2</sup>. A situational (or contextual) session is defined as a sequence of tuples (audio-track  $a$ , user  $u$ , device data  $\mathbf{d}_u^{(t)}$ , situation  $c$ ).

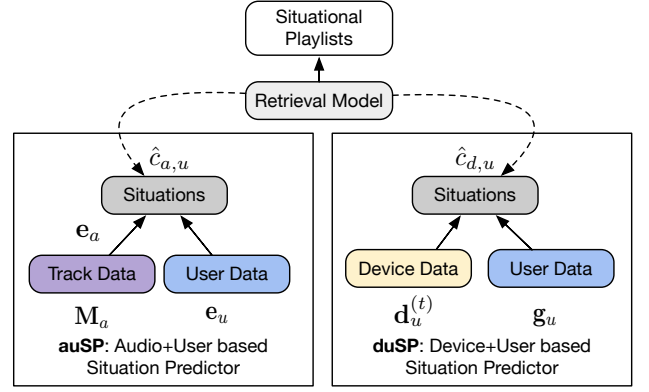
Our objectives is to propose for a given user  $u$  a session (sequence of audio-tracks  $a$ ) that fits their current situation  $c$ . However, since we don’t know its current situation  $c$  we estimate it based on its device data  $\mathbf{d}_u^{(t)}$  (such as the time of the day, day of the week, or type of network connections).

#### 3.1 Proposed approach

To do so, we first estimate for each pair audio-track/user  $(a, u)$  its situation  $\hat{c}_{a,u}$ . In other words, we estimate in which situation  $c$  the user  $u$  would intend to use the track

<sup>1</sup> without any break longer than a pre-defined gap, defined here as 20 minutes as proposed by [5]

<sup>2</sup> The underlying assumption is that if the user started streaming a playlist with a certain title related to a situation (or context), most likely the intention of the user was to play something suitable for that situation (or context) [12].



**Figure 2.** Overview of the system to generate a situational playlist. The left side (auSP) tags each track/user pair with a situational tag. The right side (duSP) ranks the potential situations for a device/user pair to be presented to the user.

$a$ . This is done using an Audio+User based Situation Prediction (auSP) trained to estimate situation tags  $c$  given as input a pair (audio-track  $a$ , user  $u$ ). This is done offline on the server side and stored in a database.

We then estimate in real-time (with a lightweight model on the client side) for a given user  $u$  and the transmitted data from its device to the service  $\mathbf{d}_u^{(t)}$ , its potential current situation  $\hat{c}_{d,u}$ . This is done using a Device+User based Situation Prediction (duSP) trained to estimate situation tags  $c$  given as input a pair (device-data  $\mathbf{d}_u^{(t)}$ , user  $u$ ). duSP provides us with a list of the most likely situations  $\hat{c}_{d,u}$  (ranked from the most to the less likely).

Finally, to create the situational playlist, we simply select the audio tracks  $a$  for which situation  $\hat{c}_{a,u}$  matches the most-likely current situations of the user  $\hat{c}_{d,u}$ .

Figure 2 indicates the overall architecture.

#### 3.2 Data description

We first describe what are exactly the data for the tracks  $a$ , the users  $u$  and the devices.

**Track data  $\mathbf{M}_a$ .** For each audio-track  $a$ , we retrieve its 30 s. snippet from the Deezer API. We represent  $a$  by its 96 Mel-bands  $\times$  646 frames matrix  $\mathbf{M}_a$ .

**User data  $\mathbf{e}_u$  and  $\mathbf{g}_u$ .** Representing the users can be achieved through various versatile techniques. Consistent with our requirements (lightweight model and preserving privacy), we choose to represent the users using the basic data available during streaming. We use two different representations of the user that will be used for estimating  $\hat{c}_{a,u}$  and  $\hat{c}_{d,u}$  respectively.

For the auSP (estimation of  $\hat{c}_{a,u}$ ) we use a user embedding  $\mathbf{e}_u$ . Similar to previous works on auSP, we used the users’ listening history to derive user embeddings that encode their listening preferences. We compute these embeddings through matrix factorization of the user/track interactions matrix, leading to a 128-d embedding vector per user, which is commonly used to generate representations [13]. The constructed matrix uses all the tracks available in the catalogue to model the user preferences, i.e. it