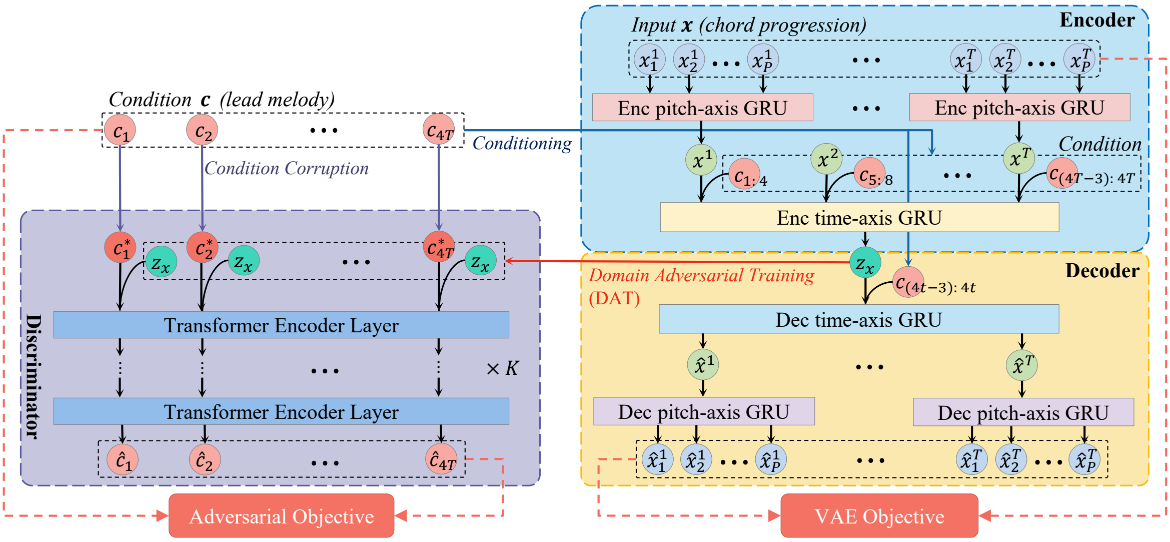

Input **x** (chord progression)
Encoder
x_1^1 x_2^1 ... x_P^1 ... x_1^T x_2^T ... x_P^T
Enc pitch-axis GRU
Enc pitch-axis GRU
Condition **c** (lead melody)
c_1 c_2 ... c_{4T}
Conditioning
Condition Corruption
Condition
x^1 $c_{1:4}$ x^2 $c_{5:8}$ ... x^T $c_{(4T-3):4T}$
Enc time-axis GRU
z_x
Domain Adversarial Training (DAT)
$c_{(4t-3):4t}$
Decoder
Dec time-axis GRU
$\hat{x}^1$ ... $\hat{x}^T$
Dec pitch-axis GRU
Dec pitch-axis GRU
$\hat{x}_1^1$ $\hat{x}_2^1$ ... $\hat{x}_P^1$ ... $\hat{x}_1^T$ $\hat{x}_2^T$ ... $\hat{x}_P^T$
Discriminator
c_1^* z_x c_2^* z_x ... c_{4T}^* z_x
Transformer Encoder Layer
× K
Transformer Encoder Layer
$\hat{c}_1$ $\hat{c}_2$ ... $\hat{c}_{4T}$
Adversarial Objective
VAE Objective

Figure 2: Chord representation learning with adversarial intervention for melody control.

represent x_p^t as a 13-D one-hot vector corresponding to 12 pitch classes plus a padding state. For most of our chord progression data, the offset of the last chord is precisely followed by the onset of the next one. Hence we do not explicitly consider the duration attributes.

For the deep structure, we build a syntax tree as in [18] to reveal the hierarchy from note via chord to chord progression. First, for $1 \le t \le T, 1 \le p \le P$, x_p^t itself constitutes the bottom layer of the tree. Then, for $1 \le t \le T$, we define x^t as the summary of $x_{1\le p\le P}^t$, which lies at the middle layer of the tree. Finally, we define z_x as the summary of $x^{1\le t\le T}$, which is the root of the tree. Such a deep structure is illustrated in Figure 3. Conceptually, while x^t is a compact representation of a single chord, z_x represents the complete chord progression.

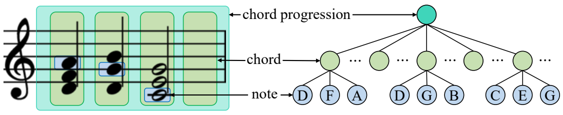


Figure 3: Tree-structure data representation of chord progression, reproduced from [18] with permission.

3.1.2 *Melody Representation*

Our model receives an 8-bar lead melody as the condition. we quantize the melody at $\frac{1}{4}$-beat unit and derive $4T = 128$ time steps. Following [7], we represent the melody as a sequence of note onsets plus a hold and a rest state. Each note onset consists of two one-hot vectors each representing 12 pitch classes and 10 octave ranges (registers). In our model, the melody pitch shares the same learnable embedding with the chord pitch.

3.2 Proposed Model

Our model applies a similar VAE architecture as PianoTree VAE [18], which learns representation for polyphonic music in a hierarchical manner. We use the surface structure of chord progression as the model input. The VAE architecture is built upon the deep tree-like structure.

We first illustrate the vanilla VAE design in the right half of Figure 2. Let x be the input chord progression and x_p^t be the p^{th} lowest pitch onset at time step t. The encoder first summarizes $x_{1\le p\le P}^t$ into an intermediate representation x^t (chord representation) for each time step t, and then encodes $x^{1\le t\le T}$ to the complete representation z_x. The decoder is basically a mirrored version of the encoder. The melody condition c, with its every four timesteps summed together, is concatenated to $x^{1\le t\le T}$ during encoding and to z_x during decoding. The loss function of our vanilla VAE architecture is:

$$\begin{aligned}\mathcal{L}(\theta_{\text{enc}}, \theta_{\text{dec}}) = &-\mathbb{E}_Q\left[\log P_{\theta_{\text{dec}}}(x \mid z_x, c)\right] \\ &+ \alpha\mathbb{KL}(Q_{\theta_{\text{enc}}}(z_x \mid x, c) \parallel \mathcal{N}(\mathbf{0}, \mathbf{1})),\end{aligned} \tag{1}$$

where $P_{\theta_{\text{dec}}}$ and $Q_{\theta_{\text{enc}}}$ refer to the VAE decoder and encoder. θ_{dec} and θ_{enc} are the learnable parameters. α is a balancing parameter for the regularization of KL loss [23].

Ideally, z_x should be a *relative* progression representation whose absolute pitch is controlled by melody c. However, as the input chord, x already has absolute pitch, this information is preserved in z_x as a redundant melody cue and confuses the decoder from attending to the condition.

To solve this problem, we assign a *discriminator* (left in Figure 2) to the VAE architecture. Instead of predicting c from z_x as conventional DAT objectives do, we bias the discriminator to denoise a *corrupted* melody condition. The corruption is done by transposing the melody to 12 keys with equal chance, which breaks the harmonic relation to the chord. In this way, we learn and extract the chord's dependency on its melody condition.

Formally, our discriminator leverages z_x to reconstruct melody condition c from a corrupted one c^*. Our DAT objective with condition corruption is trained in an adversarial manner. We optimize the discriminator by *minimizing* the reconstruction loss:

$$\mathcal{L}(\theta_{\text{dis}}) = -\mathbb{E}_Q\left[\log R_{\theta_{\text{dis}}}(c \mid z_x, c^*)\right], \tag{2}$$

where $R_{\theta_{\text{dis}}}$ is the discriminator with parameters θ_{dis}.

On the other hand, we optimize the VAE encoder by *maximizing* condition reconstruction error:

$$\begin{aligned}\mathcal{L}(\theta_{\text{enc}} \mid \theta_{\text{dis}}) = -\mathbb{E}_Q\left[\log R_{\theta_{\text{dis}}}(\mathbf{1} - c \mid z_x, c^*)\right] \\ + \alpha\mathbb{KL}(Q_{\theta_{\text{enc}}}(z_x \mid x, c) \parallel \mathcal{N}(\mathbf{0}, \mathbf{1})),\end{aligned} \tag{3}$$

where $\mathbf{1} - c$ is a confusion criterion that encourages the encoder to "fool" the discriminator. $\mathcal{L}(\theta_i \mid \theta_j)$ means we optimize θ_i while fixing θ_j. The KL loss in Equation (3) and (1) ensures a consistent posterior regularization.

During domain adversarial training, Equation (2) and Equation (3) are iteratively optimized aside from the main VAE objective (1). In this way, the encoder is explicitly biased to disentangle z_x from c. The decoder learns to retrieve missing cues from c to reconstruct x, and thus guarantees controllability in the conditional architecture.

3.3 Condition Corruption

The main novelty of our architecture over previous applications of DAT [9,10,15] is that we incorporate a corrupted condition term to generalize this method to sequential conditions. The necessity of condition corruption is that, when c is not fully implied by x, the conventional DAT objective which predicts c from z_x no longer holds. In our case, x (chord) can be accompanied with various unique c (melodies), and a melody is largely independent of the chord in terms of sequential rhythmic patterns.

Condition corruption aims to reveal the dependency of c on x when a direct predictive inference from x to c cannot be established. The corrupted condition c^* serves as a *context* to fill in such prediction gap, and the *dependency* is highlighted when using z_x to denoise c^*. It may require field knowledge to design a proper corruption method for a specific scenario. Such corruption should keep the context part while blocking the dependency.

In our case, we corrupt the melody by transposing it to 12 keys with equal probability. The transposed melody c^* keeps the original rhythm and pitch curve shape while distorting the harmonic relation to the chord progression. Here the rhythm and the curve shape are the contexts, and the harmonic relation is the dependency. We compare our corruption method with a corruption-by-masking baseline in Section 4.6 to support the effectiveness of our design.

4. EXPERIMENTS

4.1 Dataset

We collect a total of 2K lead sheet pieces (melody with chord progression) for folk and pop songs from Nottingham [24] and POP909 [25] datasets. We only keep the pieces with $\frac{2}{4}$ and $\frac{4}{4}$ meters and slice them into 32-beat snippets at an 8-beat hop size, deriving a total of 35K samples. We quantize chords at 4^{th} note and melodies at 16^{th}. We randomly split the dataset (at song level) into training (95%) and validation (5%) sets. We further augment the training data by transposing each sample to all 12 keys.

4.2 Architecture Details

The VAE framework of our model is consistent with PianoTree VAE [18]. We implement the encoder with two bi-directional Gated Recurrent Unit (GRU) networks. The pitch-axis GRU and time-axis GRU each has a hidden dimension $d_{\text{p,enc}} = 256$ and $d_{\text{t,enc}} = 512$. The input embedding dimension d_{emb} and latent representation dimension d_{z} are both set to 128. The decoder mirrors the encoder with uni-directional GRUs, with hidden dimensions $d_{\text{t,dec}} = 1024$ and $d_{\text{p,dec}} = 512$. We set the KL balancing weight $\alpha = 0.1$ in Equation (1) and (3).

We implement the discriminator using BERT [26] with relative positional embedding [27–29], as our condition corruption is conceptually similar to language masking. For our model, we use 4 Transformer encoder layers with 4 heads [30] and 10% dropout [31]. The hidden dimensions of self-attention and feed-forward layers are $d_{\text{model}} = 256$ and $d_{\text{ff}} = 1024$. Our VAE and BERT discriminator each have 12.55M and 3.24M trainable parameters.

4.3 Training

Our model is trained using Adam optimizer [32], with a mini-batch of 256 samples and a learning rate from 1e-3 exponentially decayed to 1e-5. We use teacher forcing [33] for training the GRU-based decoder, with teacher forcing rate from 0.8 exponentially decayed to 0. We introduce domain adversarial training as an iterative process aside from the main VAE objective, as shown in Algorithm 1. We set $i = 10$, $j = 1$, $k = 5$, and $l = 5$. Our model is trained on a Geforce-2080Ti-12GB GPU. It takes 20 epochs (in around 15 hours) for our model to fully converge.

Algorithm 1: Domain Adversarial Training

```
while training do
    for i iterations do
        Optimize VAE with $\mathcal{L}(\theta_{\text{enc}}, \theta_{\text{dec}})$,
    for j iterations do
        for k iteration do
            Optimize discriminator with $\mathcal{L}(\theta_{\text{dis}})$,
        for l iterations do
            Optimize encoder with $\mathcal{L}(\theta_{\text{enc}} \mid \theta_{\text{dis}})$.
```

Figure 4 shows the trends of adversarial loss $\mathcal{L}(\theta_{\text{dis}})$ (in Equation (2)) and $\mathcal{L}(\theta_{\text{enc}} \mid \theta_{\text{dis}})$ (in Equation (3)). In the early stage, the discriminator learns to reconstruct c based on z_x, so the green curve decreases. However, as the adversarial procedure goes on, z_x is gradually disentangled from c-related cues. Consequently, the discriminator acquires less and less relevant information to reconstruct c well, and thus the green curve increases. The red curve exhibits an inverse trend, as it is supervised by $\mathbf{1} - c$. When each loss curve converges, we interpret it as an equilibrium that indicates a successful disentanglement of chord representation z_x from melody condition c.

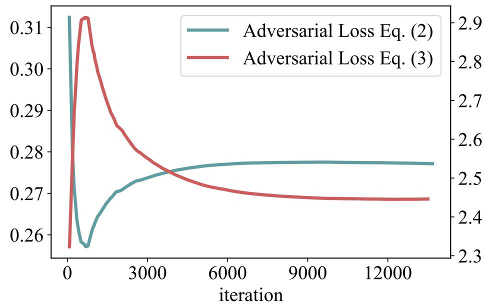
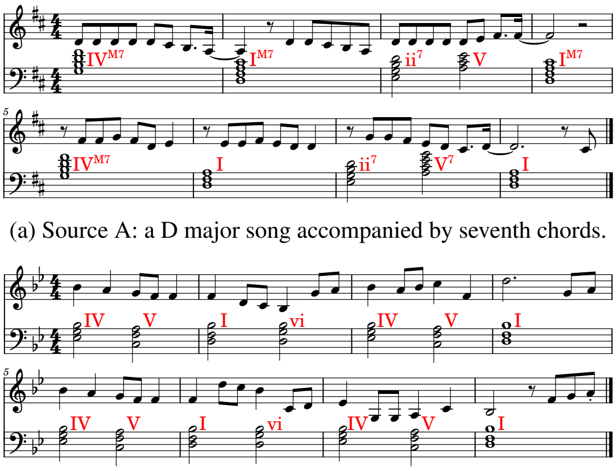


Figure 4: Adversarial loss curves with DAT. Such a trend is driven by the disentanglement of z_x from c.

4.4 Controllable Generation Results

Through domain adversarial training, our model gains reliable melody control over chord generation. Our model can harmonize a new melody using the representation of an existing chord progression. We hence develop a novel representation learning-based harmonization methodology. For example, Figure 5 presents two source lead sheets selected from our validation dataset. Both source samples are pop song phrases which share similar (but not exactly the same) chord progressions. However, the tonality and chromatic colours of these two pieces are quite different.

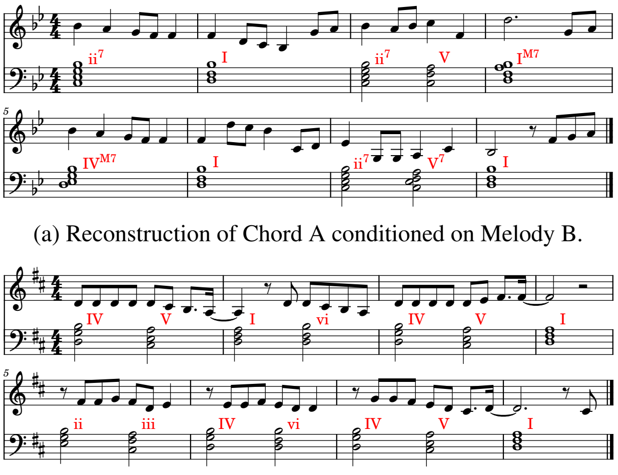


(a) Source A: a D major song accompanied by seventh chords.

(b) Source B: a B♭ major song accompanied by triads.

Figure 5: Source lead sheets.

Figure 6a is the result where we reconstruct chord A conditioned on melody B, i.e., to harmonize melody B with the harmonic style in A. Here the "style" includes tensions with seventh chords and a typical cadence progression of ii-V-I. We see these features properly fitted to melody B in the correct tone. In other words, the generation of chord progression is controlled by the melody. Figure 6b is the result where we reconstruct chord B conditioned on melody A. For this case, the original seventh chords in A are replaced by triads with a IV-V-I cadence. These results suggest that our learned chord representation can well discern relative progression and chromatic colour, while our model is controllable in terms of tonality.



(a) Reconstruction of Chord A conditioned on Melody B.

(b) Reconstruction of Chord B conditioned on Melody A.

Figure 6: Chord generation conditioned on exchanged melody conditions. This process can also be viewed as melody harmonization using exchanged harmonic styles.

4.5 Subjective Evaluation

In this section, we evaluate our model's performance on the task of *harmonization*. We first derive the following three baseline models for an ablation study:

Non-DAT: Compared with our model, Non-DAT has the same VAE framework but does not have a discriminator. It does not explicitly try to disentangle z_x from c using domain adversarial training (DAT);

Mask-CR: Mask-CR has the same architecture as our model but uses a different condition corruption technique. Specifically, it applies *masking corruption* (as in [26]) rather than pitch transposition;

Non-CR: Compared with our model, Non-CR uses the conventional DAT objective *without condition corruption*. It predicts c directly from z_x with a GRU discriminator.

To compare our model with the baselines, we survey on rating the harmonization quality of all models. Our survey has 10 groups of harmonization results and each subject is required to listen to 4. In each group, the subjects first listen to an original lead sheet A and a single melody B. Both A and B are 8-bar long (16 seconds) and are randomly selected from different musical pieces from our validation set. As in Section 4.4, we harmonize melody B with the harmonic style of A using our model and the baseline models. Subjects are then required to evaluate each version of harmonization. The rating is based on a five-point scale from 1 (very poor) to 5 (very high) over three metrics: harmonicity, creativity, and musicality.

A total of 38 subjects with diverse music backgrounds participated in our survey and we obtain 142 effective ratings for each metric. As shown in Figure 7, the height of the bars represents the mean value of the ratings. The error bars represent the mean square errors (MSEs) computed by within-subject ANOVA [34]. We report a significantly better harmonization performance of our model than all three baselines in each metric (p-value $p < 0.05$). Specifically, we note that our model achieves such performance based

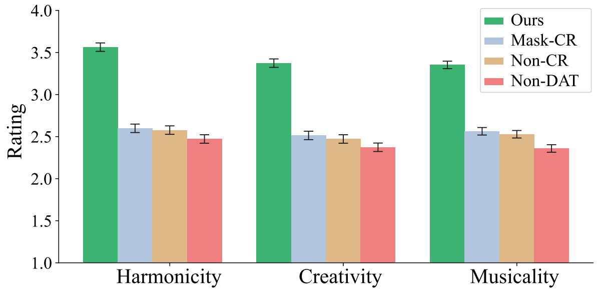


Figure 7: Subjective evaluation on the harmonization performance of our model and baseline models.

on a higher degree of representation disentanglement and controllability. We evaluate these methodological aspects with finer objective metrics in the following section.

4.6 Objective Evaluation

In this section, we objectively compare our model with the baselines in terms of *disentanglement* and *controllability*. The baseline models are as defined in Section 4.5.

4.6.1 Disentanglement

Our model disentangles chord representation z_x from melody condition c. In our case, the melody controls the absolute pitch of the chord progression. A satisfied disentanglement should derive a *pitch-invariant* representation. Following [7,35], we develop a similarity criterion to evaluate the performance on disentanglement.

Let $\mathrm{T}_i(\cdot)$ be a transposition operator with i semitones. We calculate cosine similarity $\cos(z_x, z_{\mathrm{T}_i(x)}), i = 1, 2, \cdots, 12$ for our model and for each baseline. In Figure 8, a higher similarity means representation z_x is less affected by the absolute pitch and thus is better disentangled. Our model outperforms all three baselines, including Mask-CR. This finding corroborates that a proper corruption strategy is crucial to applying domain adversarial training to concrete tasks. In our case, masking is not the best way to corrupt, as it is less aware of the harmonization context or dependency discussed in Section 3.3.

It is also worth noting that the similarity of z_x reflects human pitch perception. For each model, transposing a tritone ($\mathrm{T}_6(\cdot)$) derives the lowest similarity. Figure 8 shows that $z_{\mathrm{T}_6(x)}$ is literally orthogonal to z_x for Non-DAT and Non-CR. Interestingly, tritone is the most dissonant among all musical intervals in human perception. Such observation indicates that our model learns non-trivial music rules.

4.6.2 Controllability

A pitch-invariant representation helps us improve the model controllability by enforcing the decoder to rely on external conditions. In our case of harmonization, a good control generates harmonic chord progression conditioned on the lead melody. Aside from the subjective evaluation in Section 4.5, we introduce *harmony histogram* to objectively interpret the quality of control. Concretely, the harmony histogram is defined as the ratio of within-chord note positions on which the lead melody lies. For tonal music, there should be more root, 3rd, and 5th notes appearing in the melody compared to 7th and higher, so that the music is considered harmonic.

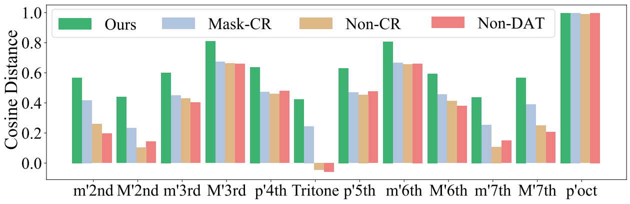


Figure 8: Object evaluation on representation similarity (invariance) against pitch transposition. A higher value denotes better disentanglement.

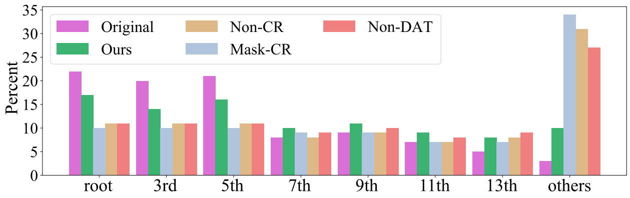


Figure 9: Objective evaluation on harmony histogram upon melody swapping. A higher ratio in root, 3rd, and 5th notes indicates a higher degree of controllability.

In our experiment, we arrange our validation data into random pairs and reconstruct the chord progression with swapped melody conditions. We compare the harmony histogram of generated results from our model and all baselines. Additionally, we compute the histogram for the original (human-composed) data as ground truth. In Figure 9, we first observe that the histogram distribution has a larger portion in the root, 3rd, and 5th notes for the original data. For the baseline models, over 25% melody notes are beyond all chord notes and tensions (shown by "others" in Figure 9), which indicates excessive disharmony. Our proposed model, on the other hand, keeps a more consistent pattern with the ground truth.

5. CONCLUSION

In conclusion, we contribute a generalized form of domain adversarial training for controllable music generation, especially when complex sequential conditions are involved. The main novelty lies in the condition corruption objective, which contextualizes the exact dependency between representation z_x and condition c, and therefore assists disentanglement and control. Our method shows excellent performance in chord representation learning, where we learn a pitch-invariant representation conditioned on the melody and develop a novel harmonization strategy. Our improvement in disentanglement and controllability is elaborated with extensive subjective and objective evaluation. With the proposal of our methodology, we hope to bring a new perspective not only to music generation but also to more general scenarios of conditional representation learning.

6. REFERENCES

[1] A. Pati and A. Lerch, "Is disentanglement enough? on latent representations for controllable music generation," in *Proceedings of the 22nd International Society for Music Information Retrieval Conference, ISMIR*, 2021, pp. 517–524.

[2] S. Dai, Z. Jin, C. Gomes, and R. B. Dannenberg, "Controllable deep melody generation via hierarchical music structure representation," in *Proceedings of the 22nd International Society for Music Information Retrieval Conference, ISMIR*, 2021, pp. 143–150.

[3] K. Chen, C. Wang, T. Berg-Kirkpatrick, and S. Dubnov, "Music sketchnet: Controllable music generation via factorized representations of pitch and rhythm," in *Proceedings of the 21th International Society for Music Information Retrieval Conference, ISMIR*, 2020, pp. 77–84.

[4] J. Jiang, G. Xia, D. B. Carlton, C. N. Anderson, and R. H. Miyakawa, "Transformer VAE: A hierarchical model for structure-aware and interpretable music representation learning," in *2020 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP*. IEEE, 2020, pp. 516–520.

[5] M. Xu, Z. Wang, and G. Xia, "Transferring piano performance control across environments," in *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP*. IEEE, 2019, pp. 221–225.

[6] J. W. Kim, R. M. Bittner, A. Kumar, and J. P. Bello, "Neural music synthesis for flexible timbre control," in *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP*. IEEE, 2019, pp. 176–180.

[7] R. Yang, D. Wang, Z. Wang, T. Chen, J. Jiang, and G. Xia, "Deep music analogy via latent representation disentanglement," in *Proceedings of the 20th International Society for Music Information Retrieval Conference, ISMIR*, 2019, pp. 596–603.

[8] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. S. Lempitsky, "Domain-adversarial training of neural networks," *Journal of Machine Learning Research*, vol. 17, pp. 59:1–59:35, 2016.

[9] L. Kawai, P. Esling, and T. Harada, "Attributes-aware deep music transformation," in *Proceedings of the 21th International Society for Music Information Retrieval Conference, ISMIR*, 2020, pp. 670–677.

[10] Y. Matsuoka and S. Sako, "Attribute-aware deep music transformation for polyphonic music," 2021, Late Breaking Demo in the 22nd International Society for Music Information Retrieval Conference, ISMIR. [Online]. Available: https://archives.ismir.net/ismir2021/latebreaking/000035.pdf

[11] J. Briot, G. Hadjeres, and F. Pachet, *Deep learning techniques for music generation*. Springer, 2020.

[12] G. Louppe, M. Kagan, and K. Cranmer, "Learning to pivot with adversarial networks," in *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems*, 2017, pp. 981–990.

[13] W. Wei, H. Zhu, E. Benetos, and Y. Wang, "A-CRNN: A domain adaptation model for sound event detection," in *2020 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP*. IEEE, 2020, pp. 276–280.

[14] F. J. Castellanos, A.-J. Gallego, and J. Calvo-Zaragoza, "Unsupervised domain adaptation for document analysis of music score images," in *Proceedings of the 22nd International Society for Music Information Retrieval Conference, ISMIR*, 2021, pp. 81–87.

[15] G. Lample, N. Zeghidour, N. Usunier, A. Bordes, L. Denoyer, and M. Ranzato, "Fader networks: Manipulating images by sliding attributes," in *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems*, 2017, pp. 5967–5976.

[16] M. Li, W. Zuo, and D. Zhang, "Deep identity-aware transfer of facial attributes," *arXiv preprint arXiv:1610.05586*, 2016.

[17] Y. Zhang, "Representation learning for controllable music generation: A survey," 2020. [Online]. Available: https://doi.org/10.13140/RG.2.2.34458.11208

[18] Z. Wang, Y. Zhang, Y. Zhang, J. Jiang, R. Yang, G. Xia, and J. Zhao, "PIANOTREE VAE: structured representation learning for polyphonic music," in *Proceedings of the 21th International Society for Music Information Retrieval Conference, ISMIR*, 2020, pp. 368–375.

[19] Z. Wang, D. Wang, Y. Zhang, and G. Xia, "Learning interpretable representation for controllable polyphonic music generation," in *Proceedings of the 21th International Society for Music Information Retrieval Conference, ISMIR*, 2020, pp. 662–669.

[20] Y. Chen, H. Lee, Y. Chen, and H. Wang, "Surprisenet: Melody harmonization conditioning on user-controlled surprise contours," in *Proceedings of the 22nd International Society for Music Information Retrieval Conference, ISMIR*, 2021, pp. 105–112.

[21] Y. Zhang, Z. Wang, D. Wang, and G. Xia, "Butter: A representation learning framework for bi-directional music-sentence retrieval and generation," in *Proceedings of the 1st workshop on NLP for music and audio (NLP4MusA)*, 2020, pp. 54–58.

[22] K. Sohn, H. Lee, and X. Yan, "Learning structured output representation using deep conditional generative

models," in *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems*, 2015, pp. 3483–3491.

[23] I. Higgins, L. Matthey, A. Pal, C. P. Burgess, X. Glorot, M. M. Botvinick, S. Mohamed, and A. Lerchner, "beta-vae: Learning basic visual concepts with a constrained variational framework," in *5th International Conference on Learning Representations, ICLR, Conference Track Proceedings*. OpenReview.net, 2017.

[24] E. Foxley, "Nottingham database," [EB/OL], https://ifdo.ca/~seymour/nottingham/nottingham.html Accessed May 25, 2021.

[25] Z. Wang*, K. Chen*, J. Jiang, Y. Zhang, M. Xu, S. Dai, and G. Xia, "POP909: A pop-song dataset for music arrangement generation," in *Proceedings of the 21th International Society for Music Information Retrieval Conference, ISMIR*, 2020, pp. 38–45.

[26] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT, Volume 1*. Association for Computational Linguistics, 2019, pp. 4171–4186.

[27] P. Shaw, J. Uszkoreit, and A. Vaswani, "Self-attention with relative position representations," in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT, Volume 2*. Association for Computational Linguistics, 2018, pp. 464–468.

[28] C. A. Huang, A. Vaswani, J. Uszkoreit, I. Simon, C. Hawthorne, N. Shazeer, A. M. Dai, M. D. Hoffman, M. Dinculescu, and D. Eck, "Music transformer: Generating music with long-term structure," in *7th International Conference on Learning Representations, ICLR*. OpenReview.net, 2019.

[29] Z. Wang and G. Xia, "Musebert: Pre-training music representation for music understanding and controllable generation," in *Proceedings of the 22nd International Society for Music Information Retrieval Conference, ISMIR*, 2021, pp. 722–729.

[30] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems*, 2017, pp. 5998–6008.

[31] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors," *arXiv preprint arXiv:1207.0580*, 2012.

[32] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *3rd International Conference on Learning Representations, ICLR, Conference Track Proceedings*, 2015.

[33] N. B. Toomarian and J. Barhen, "Learning a trajectory using adjoint functions and teacher forcing," *Neural networks*, vol. 5, no. 3, pp. 473–484, 1992.

[34] H. Scheffe, *The analysis of variance*. John Wiley & Sons, 1999, vol. 72.

[35] S. Wei and G. Xia, "Learning long-term music representations via hierarchical contextual constraints," in *Proceedings of the 22nd International Society for Music Information Retrieval Conference, ISMIR*, 2021, pp. 738–745.

MODELING PERCEPTUAL LOUDNESS OF PIANO TONE: THEORY AND APPLICATIONS

Yang Qu[1,2,*] Yutian Qin[1,3,*]
Lecheng Chao[1] Hangkai Qian[1] Ziyu Wang[1,4] Gus Xia[1,4]
[1] Music X Lab, NYU Shanghai [2] City University of Hong Kong [3] New York University
[4] Mohamed Bin Zayed University of Artificial Intelligence
yangqu7-c@my.cityu.edu.hk, {yq2120, lc4087, hq443, ziyu.wang, gxia}@nyu.edu

ABSTRACT

The relationship between perceptual loudness and physical attributes of sound is an important subject in both computer music and psychoacoustics. Early studies of "equal-loudness contour" can trace back to the 1920s and the measured loudness with respect to intensity and frequency has been revised many times since then. However, most studies merely focus on synthesized sound, and the induced theories on natural tones with complex timbre have rarely been justified. To this end, we investigate both theory and applications of natural-tone loudness perception in this paper via modeling piano tone. The theory part contains: 1) an accurate measurement of piano-tone equal-loudness contour of pitches, and 2) a machine-learning model capable of inferring loudness purely based on spectral features trained on human subject measurements. As for the application, we apply our theory to *piano control transfer*, in which we adjust the MIDI velocities on two different player pianos (in different acoustic environments) to achieve the same perceptual effect. Experiments show that both our theoretical loudness modeling and the corresponding performance control transfer algorithm significantly outperform their baselines. [1]

1. INTRODUCTION

Sound *intensity* and *loudness* are two relevant but very different terms. Intensity is the physical feature of sound derived from sound pressure. Loudness, however, is the perceptual measure dependent on our auditory systems, where other physical factors also contribute to human perception. For example, a 1 kHz tone is measured louder than a 100 Hz tone of equal intensity, but softer than an equal-intensity white noise. Previous psychoacoustic studies provide knowledge of loudness perception via human subject experiments and various computational models have been proposed to account for loudness estimation [1–6].

However, existing theories mainly focus on synthesized tones and thus fall short of explaining natural tone loudness such as the piano tone. The generation of piano tone involves a complicated physical process [7] and the sound contains rich timbral variations hard to be synthesized from both frequency and time domain perspectives [8, 9]. On the other hand, recently we see a growing number of music information retrieval tasks involving feature extraction of piano tone loudness, such as automatic music transcription [10–12] and performance rendering [13, 14]. In most of these studies, loudness is sometimes confused with intensity or even the MIDI velocity. This motivates us to investigate loudness perception specific to piano tone, beneficial for various downstream applications.

In this paper, we investigate both theory and applications on loudness perception specific to piano tone. The theory is studied in a hybrid method containing a *psychoacoustic* procedure and a *machine-learning* procedure. In the psychoacoustic procedure, we measure the "equal-loudness contour" on piano based on a human subject experiment similar to the pure-tone equal-loudness measurement [1]. Since piano tones cannot be directly controlled by frequency and intensity, we instead study pitch and velocity, two corresponding discrete performance controls. In our experiment, the controls are executed by (acoustic) player pianos for accuracy and reproducibility. We show that different pianos (in different acoustic environments) have distinctive equal-loudness contour patterns, and existing methods have limited power to explain the measured pattern.

In the machine-learning procedure, we extend the piano-tone equal-loudness contour to a computational model, capable of inferring loudness purely from spectral features. Traditionally, loudness models are based on mathematical approximation of our auditory systems [3–6]. We argue such an approach is laborious and even intractable for natural tones such as the piano tone to take into account all contributing facets of sound. Instead, we propose a machine-learning approach including a loudness model trained on the human subject measurement in the previous experiment. The model takes in the spectrogram and outputs the estimated loudness, calibrated to standard loudness unit *sone* in post-processing.

[*] The two authors have equal contribution.

[1] Code and models can be accessed via https://github.com/yangqu2000/ModelingPerceptualLoudnessOfPianoTone

We show our theory of piano tone loudness provides a more reasonable explanation of piano sounds in downstream applications. Specifically, we apply our loudness model to *performance control transfer* [14], in which we adjust the MIDI velocities on two different player pianos (in different acoustic environments) to achieve the same perceptual effect. The original intensity-based loudness estimator is replaced with the proposed method, and experimental results show a significant improvement in terms of transfer quality.

2. RELATED WORK

The study of the equal-loudness contour (ELC) of pure tone is a serious subject in psychoacoustics, since it reveals fundamental facts of human loudness perception with respect to frequency spectrum and intensity. The experiment settings have been modified throughout the century, including the tuning of listening conditions [1, 15, 16], the improvement of experiment procedures [1,17,18], and the extension of measured frequency range [1,16,19,20]. Suzuki *et al.* [2] provides a detailed literature review of the existing experiments. Our piano-tone ELC experiment is modified from the original one, including careful adjustment to account for discrete frequency and intensity control.

The psychoacoustic listening tests of pure tone and many others [1, 16–18, 21, 22] provide fundamental understanding of our auditory systems. In return, the study of *loudness models* takes in the auditory system hypotheses and yields loudness estimation of the different types of tone, including [1, 16–18] for pure tone, [21] for complex tone, and [22, 23] for complex tone with amplitude modulations. So far, the state-of-the-art loudness model ISO 532-3 [24, 25] has the theoretical power to estimate the loudness of arbitrary waveforms. In our paper, we validate and compare with this model particularly on piano tone loudness estimation.

The majority of research on piano tone mainly focuses on the relationship between instrument control and the physical attributes of the generated sound, either statistically [26–28] or via physical modeling [7, 29, 30]. However, there is a lack of formal theory to cover the perception of piano tone. In this paper, we provide the first approach to study the relationship between instrument control and piano tone loudness in a psychoacoustic approach.

3. PIANO TONE EQUAL-LOUDNESS CONTOUR

Unlike pure tone which is determined by frequency and intensity, piano tone is largely dependent on the environment including the instrument itself that generates the sound (e.g., upright or grand piano) and the surrounding acoustic environment (e.g., concert hall or small room). Given a fixed environment, a piano tone can be controlled by four factors, namely *pitch*, *velocity*, *duration* and *pedal* [31]. Simple as they are, the four control parameters interact with the acoustic environment and produce a wide spectrum of piano timbre, resulting in an unexplored loudness perception pattern. In this paper, we study how perceptual loudness is affected by the first two factors, i.e., pitch and velocity, which can be roughly understood as the "frequency" and "intensity" of piano tone, respectively. We leave the others for future study.

The piano tones in our experiment are produced by player pianos, which can execute the control parameter in a more accurate manner than human pianists. Pitch is controlled by 88 MIDI pitches in $[21..108]$ and velocity is controlled by 128 velocity levels in $[0..127]$.

3.1 Method of Measurement

The goal of this experiment is to measure the *equal-loudness contours* (ELC) of piano tone on a specific piano and acoustic environment, that is, a sequence of piano keys under possibly different velocities that have equal perceptual loudness.

The experiment is modified from the original pure-tone ELC experiments [1]. Specifically, we first select a reference tone, denoted by $(p_{\text{ref}}, v_{\text{ref}})$, where p_{ref} is a reference pitch and v_{ref} is the velocity level that we are interested in. Then, we enumerate piano pitches and for each pitch p_{var}, we ask subjects to listen to $(p_{\text{var}}, v_{\text{var}}), v_{\text{var}} \in [0..127]$ and choose the louder note until they find a velocity v^*_{var} such that $(p_{\text{var}}, v^*_{\text{var}})$ and $(p_{\text{ref}}, v_{\text{ref}})$ have equal loudness.

In psychoacoustic terms, the reference tone $(p_{\text{ref}}, v_{\text{ref}})$ is called a *reference stimulus*, and given the pitch, the candidate tones under multiple velocities to compare (i.e., $(p_{\text{var}}, v_{\text{var}})$) are called *variable stimuli*. The solution $(p_{\text{var}}, v^*_{\text{var}})$ is called the *point of subjective equality* (PSE). The curve that connects the PSEs at multiple pitches is the equal-loudness contour at reference velocity v_0.

To adjust the velocity of reference stimuli and search for the PSE of each pitch, we adopt *randomized maximum likelihood sequential procedure* (RMLSP) [1], a common procedure used in pure-tone ELC experiments consisting of a series of test rounds. At each test round, the subject is asked to compare the loudness of a pair of piano tones, including the reference stimulus and a variable stimulus. Then we fit an online logistic regression model to predict the PSE according to the subject's response so far. The velocity of the variable stimulus in the next test round is randomly selected within a velocity range centered at this PSE velocity. The algorithm ensures the procedure will converge to the correct PSE.

3.2 Experiment Setting

In our experiment, we use A4 in the middle of the keyboard as the reference pitch (i.e., $p_{\text{ref}} = 69$). We measure equal loudness contours at four velocity levels: $v_{\text{ref}} \in \{32, 44, 60, 80\}$. The four velocities lie in the common range of velocity usage and are evenly distributed with respect to the statistical distribution [11].

We measure equal-loudness contours on 9 variable pitches: $p_{\text{var}} \in \{21, 33, 45, 57, 69, 81, 93, 105, 108\}$. The pitches are the A's in all octaves together with the highest note C8. The measurement on all 88 keys is ideal though not affordable, since RMLSP normally takes 20 minutes to find the PSE with respect to one single variable pitch.