DOMAIN ADVERSARIAL TRAINING ON CONDITIONAL VARIATIONAL AUTO-ENCODER FOR CONTROLLABLE MUSIC GENERATION

Jingwei Zhao[2,4] **Gus Xia**[3,5] **Ye Wang**[1,2,4]
[1] School of Computing, NUS [2] Institute of Data Science, NUS [3] Music X Lab, NYU Shanghai
[4] Integrative Sciences and Engineering Programme, NUS Graduate School [5] MBZUAI
jzhao@u.nus.edu, gxia@nyu.edu, wangye@comp.nus.edu.sg

ABSTRACT

The variational auto-encoder has become a leading framework for symbolic music generation, and a popular research direction is to study how to effectively *control* the generation process. A straightforward way is to control a model using different conditions during inference. However, in music practice, conditions are usually sequential (rather than simple categorical labels), involving rich information that overlaps with the learned representation. Consequently, the decoder gets confused about whether to "listen to" the latent representation or the condition, and sometimes just ignores the condition. To solve this problem, we leverage *domain adversarial training* to *disentangle* the representation from condition cues for better control. Specifically, we propose a condition corruption objective that uses the representation to denoise a corrupted condition. Minimized by a discriminator and maximized by the VAE encoder, this objective adversarially induces a condition-invariant representation. In this paper, we focus on the task of melody harmonization [1] to illustrate our idea, while our methodology can be generalized to other controllable generative tasks. Demos and experiments show that our methodology facilitates not only condition-invariant representation learning but also higher-quality controllability compared to baselines.

1. INTRODUCTION

In deep music generation, improving *controllability* has been a major challenge that gains increasing research attention [1–6]. In practice, controllability is typically implemented under a conditional architecture, where the generation process is biased by external condition inputs. For example, EC^2-VAE [7] learns a representation z_x of 8-beat melody x while the underlying chords are given as condition c. The system is controllable if the generated melody can adapt to variable chords properly. For such representation-learning architectures, however, the decoder tends to find a shortcut from z_x to x without attending to c, leading to "condition collapse". The reason for this, as we argue, is that z_x is inevitably intertwined with condition c in the representation space, as c is often an innate property of x. In the case of EC^2-VAE, the condition of chords is very much implied by the melody.

To address this problem, the representation z_x must be disentangled from condition c. A popular way to achieve this goal is to use an adversarial objective that predicts c from z_x, as shown in Figure 1. On the one hand, this objective is optimized by a discriminator; on the other hand, the encoder is trained to "fool" the discriminator by detaching c-related cues out of z_x. In this way, the decoder cannot find a shortcut in z_x but is forced to seek c to reconstruct x. Such a technique stems from *domain adversarial training* (DAT) [8], where the "domain" is interpreted as "condition" that controls the generation.

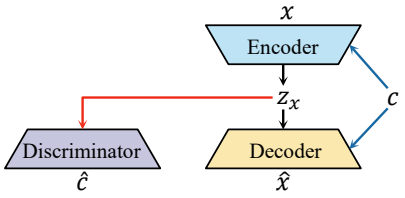


Figure 1: An illustration of domain adversarial training over a conditional generation architecture.

Apparently, DAT can be a powerful tool for controllable music generation. Previous studies [9, 10] have discussed simple scenarios where the condition is a global label (*e.g.*, note density). In music practice, however, local and sequential conditions [11] are more common. In such cases, c may not be fully implied by x, so the objective that simply predicts c from z_x does not necessarily hold.

In this paper, we focus on sequential conditions and develop a generalized form of DAT for controllable music generation. We illustrate our methodology with the task of *chord representation learning conditioned on melody*, where x stands for the chord progression, and c is the melody condition. In general, a chord progression can match many melodies, so we cannot directly predict c (melody) from z_x (chord) for the DAT objective. Instead, we leverage z_x to reconstruct c from a corrupted condition c^*. We rely on c^* to provide the melody context that cannot be hinted by chord x; on the other hand, the corrupted

[1] Demos and codes via https://zhaojw1998.github.io/DAT_CVAE.

information reveals c's harmonic dependency on x, which we enforce the discriminator to learn. With proper corruption design, our DAT objective can be generalized to more scenarios with sequential conditions.

A well-trained model with good controllability can help us harmonize a new melody using the representation (style) of an existing chord progression. Experiments show that our model performs an excellent disentanglement of data representation from the condition, and the controllability outperforms the baselines. In summary, our contributions in this paper are as follows:

- **A general approach to controllability**: Based on a novel adversarial objective with condition corruption, we generalize domain adversarial training to music generation with sequential conditions;
- **A novel harmonization methodology**: We present a representation learning-based method for melody harmonization. Our current model harmonizes pop and folk melodies with the triad and seventh chords.

2. RELATED WORKS

2.1 Domain Adversarial Training

Domain adversarial training (DAT) is a representation learning approach initially proposed for domain adaptation tasks [12–14]. Through an adversarial process as described in Section 1, DAT enforces *domain invariance* to data representation so that it can be adapted to different domains flexibly. Such adaptability to new domains is analogous to controllability with new conditions. For generation tasks, DAT has been utilized to learn a condition-invariant data representation. Such invariance enforces the decoder to use condition information for reconstruction [15]. During inference, the decoder "listens to" new conditions as well and generates new data in a controllable way.

The first attempts that incorporate DAT with generation dealt with facial image generation conditioned on binary attributes (*e.g.*, male or female) [15, 16]. Such conditions cannot be explicitly supervised because we cannot find any pair of images that represents the same person both male and female. Fortunately, DAT enforces attribute invariance at encoding and learns attribute dependency at decoding, thus circumventing this problem. Recently, DAT has been extended to symbolic music generation conditioned on various attributes. Kawai *et al.* adopts DAT to a variational auto-encoder (VAE) for melody generation conditioned on statistical attributes (*e.g.*, note density) [9]. Later, Matsuoka *et al.* generalizes this methodology to generating polyphonic music with similar conditions [10].

For previous works, the conditions are particularly a global statistical label, which only represents a limited scenario of controllable generation. In our paper, we generalize the usage of DAT to sequential conditions. Conditioned on an 8-bar melody, we aim to learn a pitch-invariant representation of an 8-bar chord progression, which can later be adapted to varied melody conditions and to harmonize them. Our main novelty lies in a special design of the adversarial objective, which is to denoise corruption rather than make full prediction. This technique greatly helps us in dealing with the nuance of sequential conditions.

2.2 Controllable Music Generation

Controllable music generation takes various forms in terms of controlling technique and music representation [17]. For controlling technique, controllability can be achieved by sampling, interpolation, conditioning, and more ways [11]. For music representation, controls can be performed over statistical music properties (pitch variability, note density, etc.) [9, 10], compositional factors (chord progression, texture and rhythmic patterns, etc.) [7, 18–20], high-level semantics (emotion, cultural style, etc.) [21], and so on. With the development of representation learning, such properties can be abstracted and disentangled for flexible control.

In this paper, we are interested in chord representation learning conditioned on melodies, which falls into the category of controlling compositional factors via conditioning. Various conditional architectures, such as conditional VAE (C-VAE) [22], have been applied for similar purposes [7, 18–21]. However, as the condition is often easily implied by the representation, the decoder tends to skip the condition, and simply reconstruct the data for whatever conditions. To eradicate this problem, we introduce domain adversarial training and generalize it to sequential conditions (in our case, an 8-bar lead melody). Our model learns a pitch-invariant chord representation so that we can generate chord progressions harmoniously conditioned on varied melodies. Such control over compositional factors is common to broader music generation scenarios, and our methodology is generally applicable as well.

3. METHODOLOGY

In this section, we introduce our methodology with domain adversarial training on learning chord representation conditioned on the melody. An overview of our model is illustrated in Figure 2. We first describe our data representation and structure in Section 3.1. Then, we introduce our proposed model in Section 3.2. Finally, we elaborate on our novel design of condition corruption in Section 3.3.

3.1 Date Representation and Structure

3.1.1 Chord Representation

Our model generates an 8-bar chord progression conditioned on the melody. We quantize the chord progression at 1-beat unit and derive $T = 32$ timesteps. The maximum note count P for each chord is 4, which means we can flexibly represent any type of triad and seventh chords. Specifically, we treat chord progression as a piece of polyphony and follow [18] to represent it in both a surface structure (as model input) and a deep structure (for encoding).

The surface structure is a nested array of pitch attributes, denoted by $\{x_p^t | 1 \leq t \leq T, 1 \leq p \leq P\}$. Concretely, x_p^t is the p^{th} lowest pitch onset at time step t. We

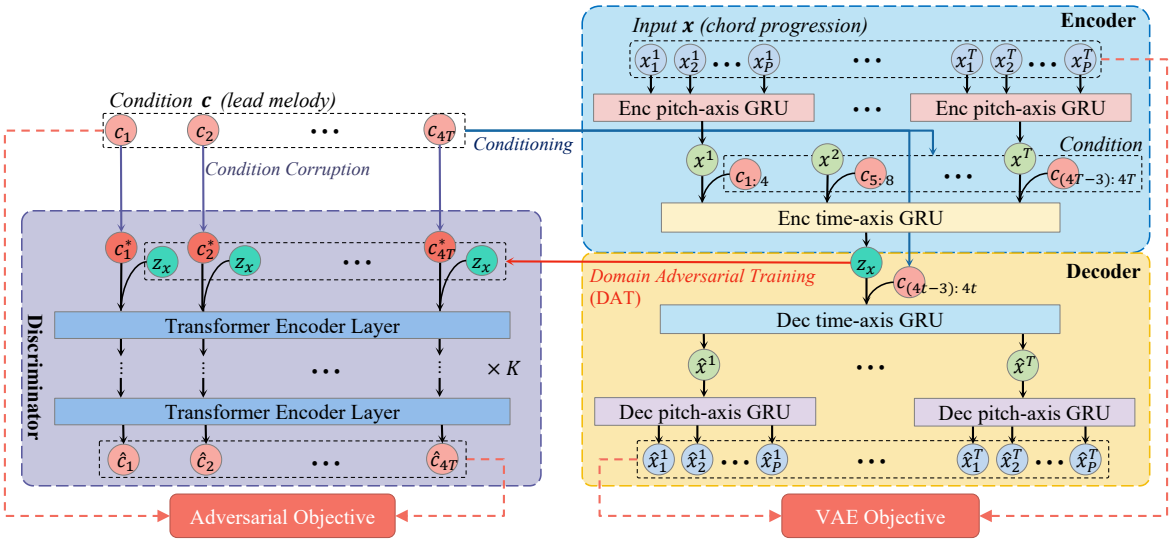


Figure 2: Chord representation learning with adversarial intervention for melody control.

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 $1 \leq t \leq T, 1 \leq p \leq P$, x_p^t itself constitutes the bottom layer of the tree. Then, for $1 \leq t \leq T$, we define x^t as the summary of $x_{1\leq p\leq P}^t$, which lies at the middle layer of the tree. Finally, we define z_x as the summary of $x^{1\leq t\leq T}$, 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.

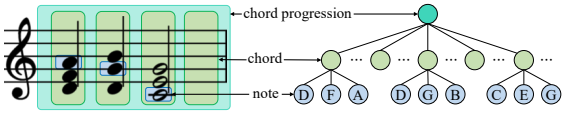


Figure 3: Tree-structure data representation of chord progression, reproduced from [18] with permission.

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 $x_{1\leq p\leq P}^t$ into an intermediate representation x^t (chord representation) for each time step t, and then encodes $x^{1\leq t\leq T}$ 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 $x^{1\leq t\leq T}$ during encoding and to z_x during decoding. The loss function of our vanilla VAE architecture is:

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

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:

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

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:

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

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 L(θ_enc, θ_dec),
    for j iterations do
        for k iteration do
            Optimize discriminator with L(θ_dis),
        for l iterations do
            Optimize encoder with L(θ_enc | θ_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.

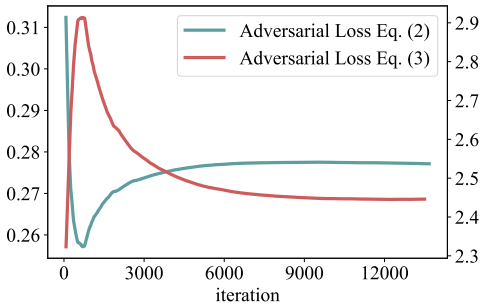
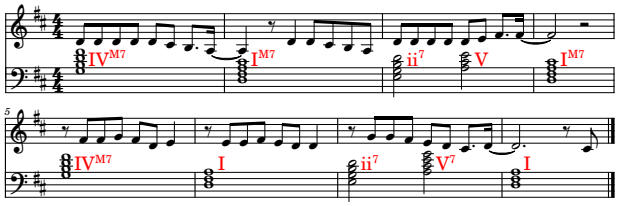


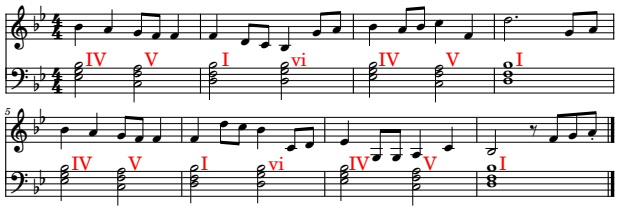
Figure 4: Adversarial loss curves with DAT. Such a trend is driven by the disentanglement of z_x from c.

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.



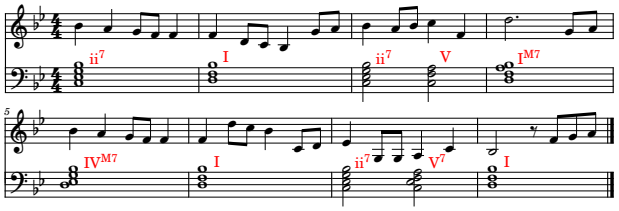
(a) Source A: a D major song accompanied by seventh chords.



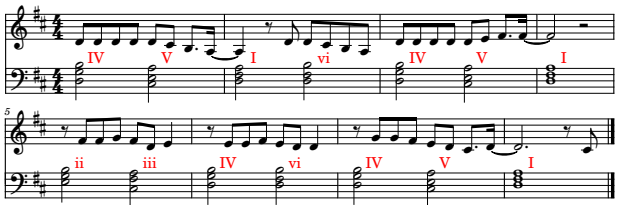
(b) Source B: a B♭ major song accompanied by triads.

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

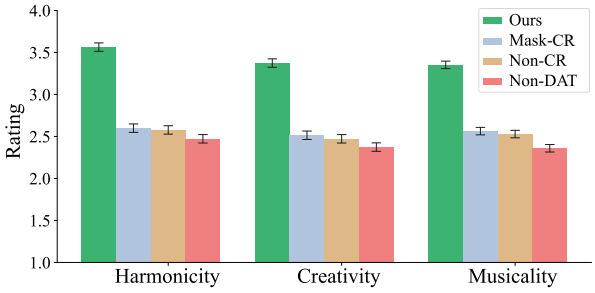


Figure 7: Subjective evaluation on the harmonization performance of our model and baseline models.

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,

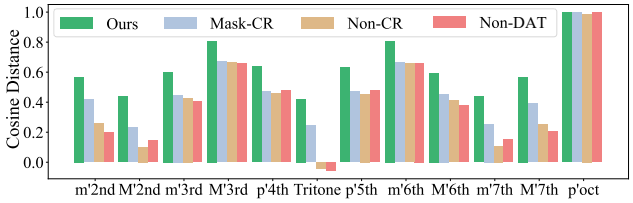


Figure 8: Object evaluation on representation similarity (invariance) against pitch transposition. A higher value denotes better disentanglement.

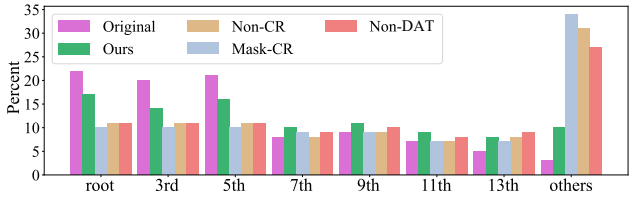


Figure 9: Objective evaluation on harmony histogram upon melody swapping. A higher ratio in root, 3rd, and 5th notes indicates a higher degree of controllability.

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.

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.