item). Given a session index ℓ with N sessions in total, a session $S_\ell = [x_1, x_2, ..., x_{|S_\ell|}]$ is composed of a sequence of tracks, where $x_t \in \mathcal{V}$ is the t-th track in the session and $\mathcal{V}$ is the set of all tracks in the data. The goal of is to predict the next track (i.e., $x_{|S_\ell|+1}$) given the past interactions $[x_1, \ldots, x_{|S_\ell|}]$ in a given session S_ℓ. We aim to recommend top-K tracks for each session, given that user identity information is inaccessible due to the inherent nature of anonymous sessions.

3.1.2 ***Session-based Recommendation.*** Given an input session $S_\ell = [x_1, x_2, ..., x_{|S_\ell|}]$, recommender systems generally embed the tracks into embedding vectors, $\mathbf{E}_\ell = [\mathbf{e}_1, \mathbf{e}_2, ..., \mathbf{e}_{|S_\ell|}]$, where $\mathbf{e}_t \in \mathbb{R}^d$ is the d-dimensional embedding of the t-th track. Then, a track encoder f produces the representation of each track, $\mathbf{H}_\ell = f(\mathbf{E}_\ell) = [\mathbf{h}_1, \mathbf{h}_2, ..., \mathbf{h}_{|S_\ell|}]$, where $\mathbf{h}_t \in \mathbb{R}^d$, by modeling the interaction among tracks. Then, an aggregation layer g aggregates the track representations into a session representation $\mathbf{z}_\ell = g(\mathbf{H}_\ell)$ where $\mathbf{z}_\ell \in \mathbb{R}^d$. Given the session representation $\mathbf{z}$, a prediction layer with softmax operation produces the prediction probability for all tracks, $\hat{\mathbf{y}} = \{\hat{y}_1, \hat{y}_2, ..., \hat{y}_{|\mathcal{V}|}\}$. The training loss $\mathcal{L}_{rec}$ can be a classification loss such as cross-entropy loss. Lastly, the model recommends top-K tracks based on the prediction probability $\hat{\mathbf{y}}$.

3.1.3 ***Self-Supervised Learning (SSL) Framework.*** To leverage shuffle-play sessions during training, we propose a SSL framework, as shown in Figure 3. The framework takes a session S_ℓ as input, which can be either a shuffle or non-shuffle play session. Given the input session S_ℓ, an augmentation operation $\mathcal{A}$ augments the input session based on the transition frequency. As a result, the recommender system better captures users' preferences from the shuffle play sessions to provide more accurate recommendations. More formally, we embed the tracks into embedding vectors, $\mathbf{E}_\ell$ and $\tilde{\mathbf{E}}_\ell$, from the original and augmented sessions (i.e., S_ℓ and $\tilde{S}_\ell$), respectively. Then, a track encoder f produces track representations by modeling the interaction among the tracks in each session, i.e., $\mathbf{H}_\ell = f(\mathbf{E}_\ell)$ and $\tilde{\mathbf{H}}_\ell = f(\tilde{\mathbf{E}}_\ell)$. The aggregation layer g aggregates the track representations into session representations, i.e., $\mathbf{z}_\ell = g(\mathbf{H}_\ell)$ and $\tilde{\mathbf{z}}_\ell = g(\tilde{\mathbf{H}}_\ell)$. A basic SSL approach aligns the final representations (i.e.. $\mathbf{z}_\ell$ and $\tilde{\mathbf{z}}_\ell$) by increasing their similarity, resulting in the alignment loss (i.e., $\mathcal{L}_{align}$). We note that we employ a shared track encoder f and a shared aggregation layer g in both branches.

3.2 Transition-based Augmentation

Transition-based augmentation aims to enrich the sequential information in a given shuffle play session. To this end, we consider the transition frequency between items from all the sessions as an essential criterion for distinguishing shuffle and non-shuffle play sessions, as shown in Figure 4. We first demonstrate how we obtain a transition matrix and propose a novel session augmentation method conducted based on the transition matrix.

Transition Matrix. As shown in Figure 2, the main challenge inherent in the shuffle play sessions is their excessive amount of unique transitions within a session. To address the problem of excessive unique transitions, we introduce non-unique transition patterns observed across all sessions to shuffle play sessions. By doing so, we effectively mitigate the unique transition patterns inherent in shuffle play sessions, thereby unlocking the potential for leveraging

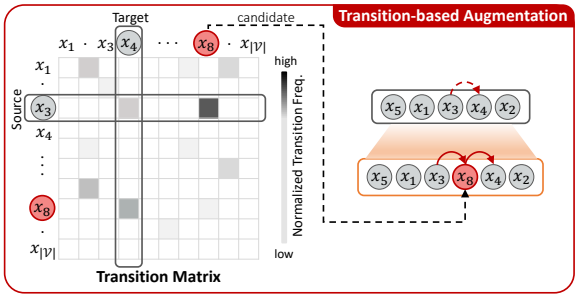

Figure 4: Our proposed transition-based augmentation showing an example of inserting a track x_8 between x_3 and x_4.

these sessions during the training process. More formally, we first generate a transition frequency matrix $\mathbf{T} \in \mathbb{R}^{|\mathcal{V}|\times|\mathcal{V}|}$, by collecting all transitions observed in the entire sessions as follows:

$$\mathbf{T}_{i,j} = \sum_{\ell=1}^{N} \sum_{t=1}^{|S_\ell|-1} \mathbb{1}([x_t, x_{t+1}] = [x_i, x_j]), \quad \forall i, j \leq |\mathcal{V}| \tag{1}$$

where $\mathbf{T}_{i,j}$ denotes the frequency of transition from source track x_i to target track x_j, $\mathbb{1}(a = b)$ denotes indicator function which outputs 1 if $a = b$ else 0, and N is the total number of sessions. We also take the logarithm to each value in $\mathbf{T}$ as the transition frequency of certain pairs, e.g., the transition between popular tracks, tends to be much higher than that of the remaining cases[4], which may incur the long-tail problem [27, 36, 39, 43, 48]. We then normalize the log-transformed matrix from the following two perspectives:

$$\bar{\mathbf{T}}_{i,\cdot} = \frac{\mathbf{T}_{i,\cdot}}{\sum_{j=1}^{|\mathcal{V}|} \mathbf{T}_{i,j}}, \quad \forall i \leq |\mathcal{V}|, \quad \bar{\mathbf{T}}_{\cdot,j} = \frac{\mathbf{T}_{\cdot,j}}{\sum_{i=1}^{|\mathcal{V}|} \mathbf{T}_{i,j}}, \quad \forall j \leq |\mathcal{V}| \tag{2}$$

where $\bar{\mathbf{T}}_{i,\cdot}$ and $\bar{\mathbf{T}}_{\cdot,j}$ denote the row-wise (i.e., source-wise) and column-wise (i.e., target-wise) normalized transition matrices, respectively. This results in the Markov Chain Transition Matrices [35], where the transition probability of each source and target node sums to one. This normalization enables us to interpret the transition matrix in terms of the probability distribution matrix and take a stochastic approach while augmenting a given session.

Transition-based Insertion. We now propose a novel session augmentation method to handle shuffle play sessions for music recommendation. The main idea is to insert frequently appearing transitions that could exist in a session. The primary goals of the augmentation are: (1) to reduce the excessive amount of unique transitions in shuffle play sessions and (2) to expose the session encoder to more diverse environments, thereby better accommodating shuffle play sessions. More precisely, our proposed augmentation method determines which items to be inserted at which locations in a given session. Here, the key idea lies in not inserting any random items but inserting relevant items that are likely to appear considering its back-and-forth context, i.e., source and target. For a clear and comprehensive understanding, the reader is encouraged to refer to Figure 4, which illustrates a toy example of inserting x_8 between x_3 and x_4. Specifically, as x_3 has a high transition probability to x_8, and x_8 has a high transition probability to x_4, we insert x_8 between x_3 and x_4.

[4]Here, we ensure log transformation is applied to non-zero frequency values.