Birmingham 2023 CIKM

Full Paper

MUSE: Music Recommender System with Shuffle Play Recommendation Enhancement

Yunhak Oh*, Sukwon Yun*, Dongmin Hyun, Sein Kim, and Chanyoung Park†

* Both authors contributed equally to this research
† Corresponding author

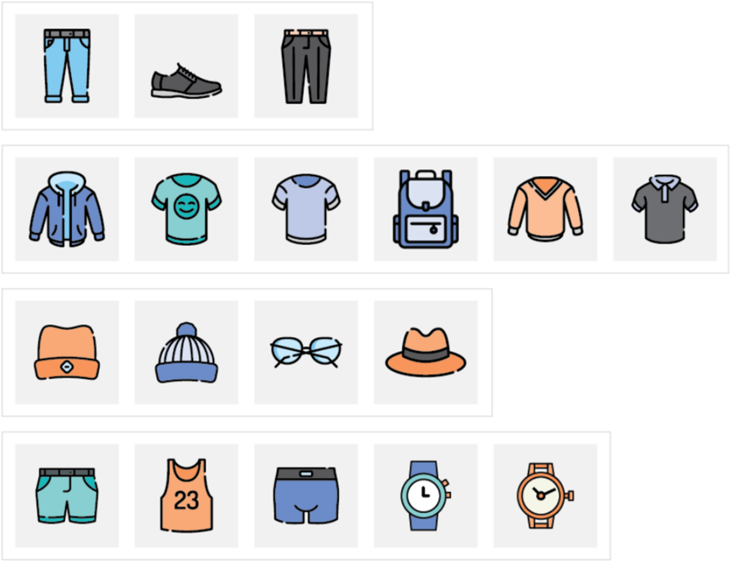
KAIST

DSAIL Data Science & Artificial Intelligence

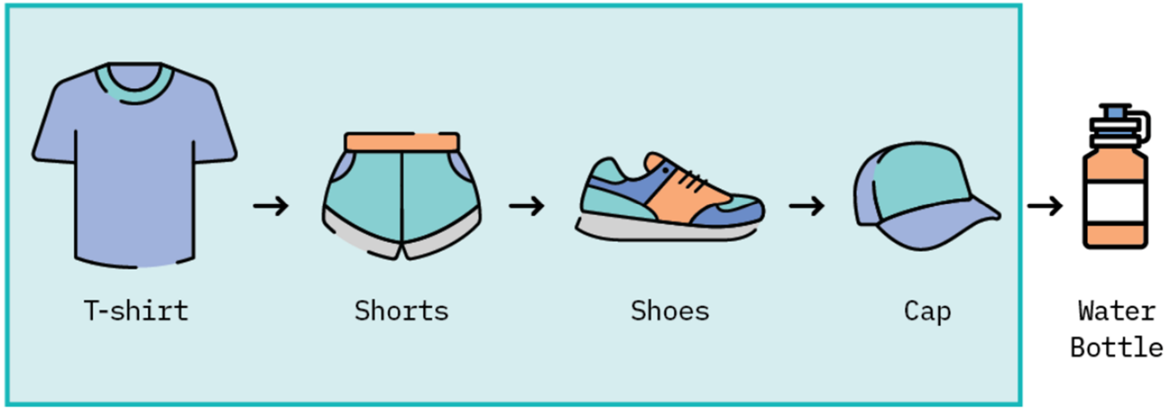
CONTENTS

BACKGROUND

- **Session-based Recommendation (SBR)**
  - Anonymous (No user profiles) & Short
  - Solely based on a user's interactions in an ongoing session



(a) Session

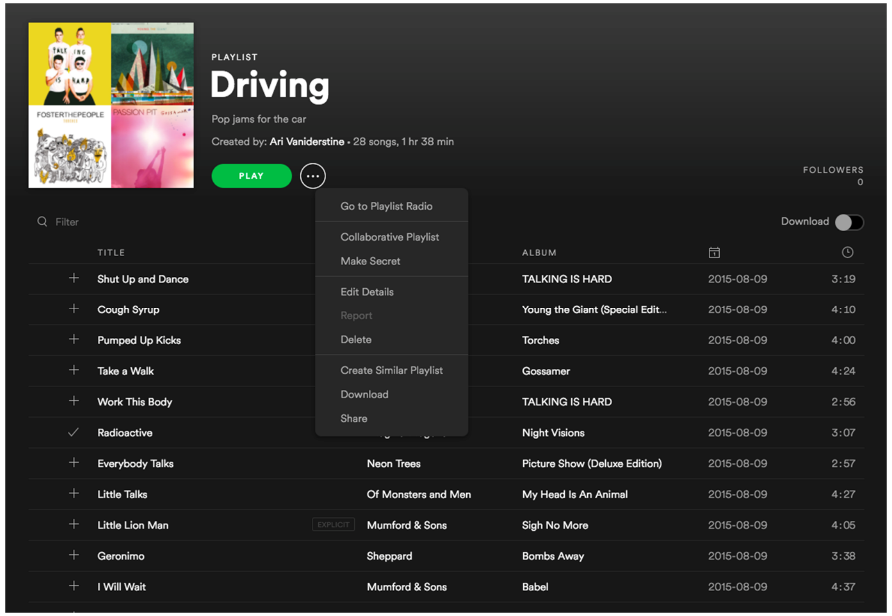


(b) Next Item Prediction

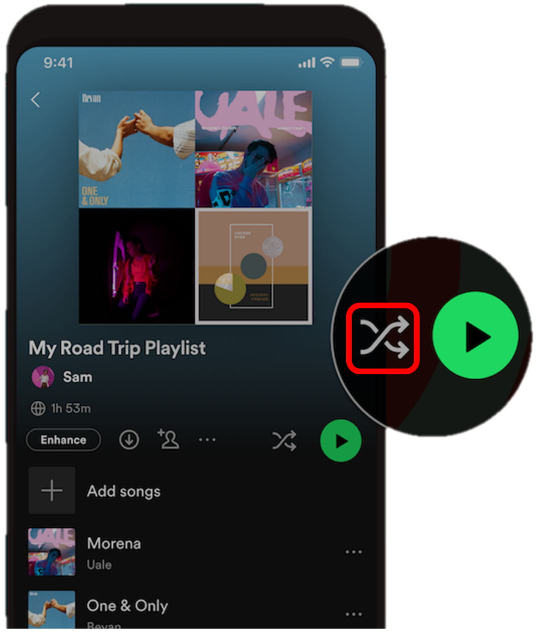
Image Credit: https://session-based-recommenders.fastforwardlabs.com/

MOTIVATION SHUFFLE PLAY

- Recommender Systems (RS) have become indispensable in music streaming services
  - Personalize playlists
  - Facilitate the serendipitous discovery of new music
- Unique Challenge in Music Domain: **Shuffle Play**



(a) Playlist



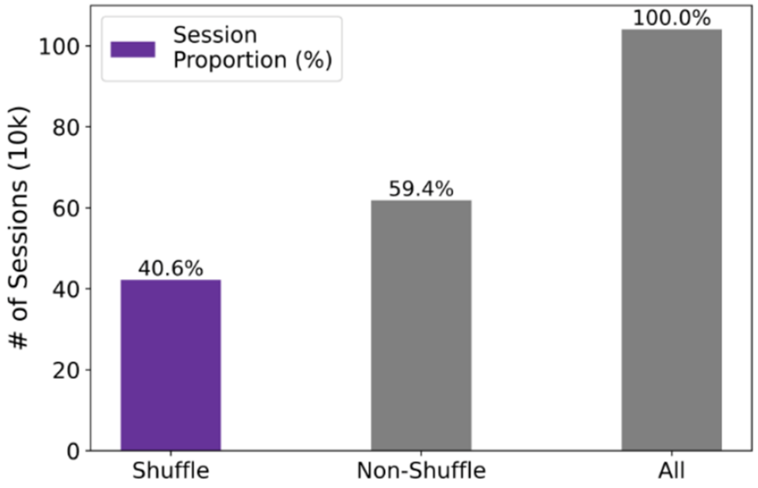
(b) Shuffle Play

Image Credit: https://developer.spotify.com/documentation/web-api/concepts/playlists
https://newsroom.spotify.com/2022-08-01/spotify-is-launching-individual-buttons-for-shuffle-and-play-for-spotify-premium-users-so-its-simpler-to-choose-the-way-you-listen/

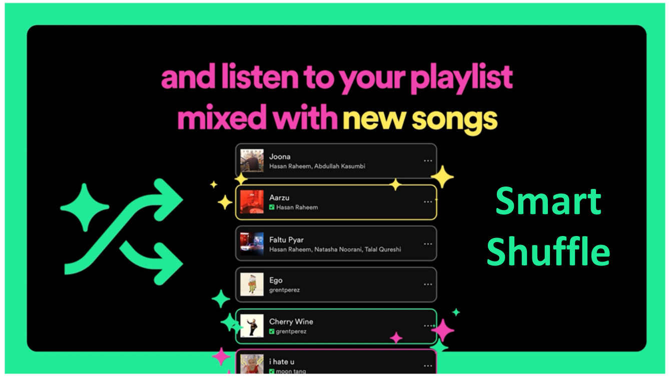
MOTIVATION WHY SHUFFLE PLAY

[Smart Shuffle](#)

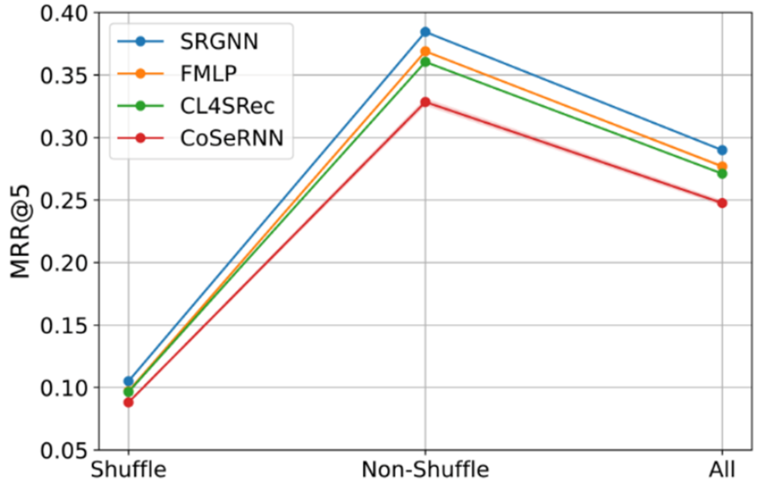
- **Users enjoy Shuffle Play**
  - Substantial proportion (i.e., 40.6%)
  - Mitigate listening monotony [2]
  - Present serendipity in the user's auditory journey [2]
  - Spotify announced new play mode: Smart Shuffle
- Existing methods performed **poorly** in **shuffle play** sessions



(a) Proportion of session by play mode



(b) "Smart Shuffle" by Spotify

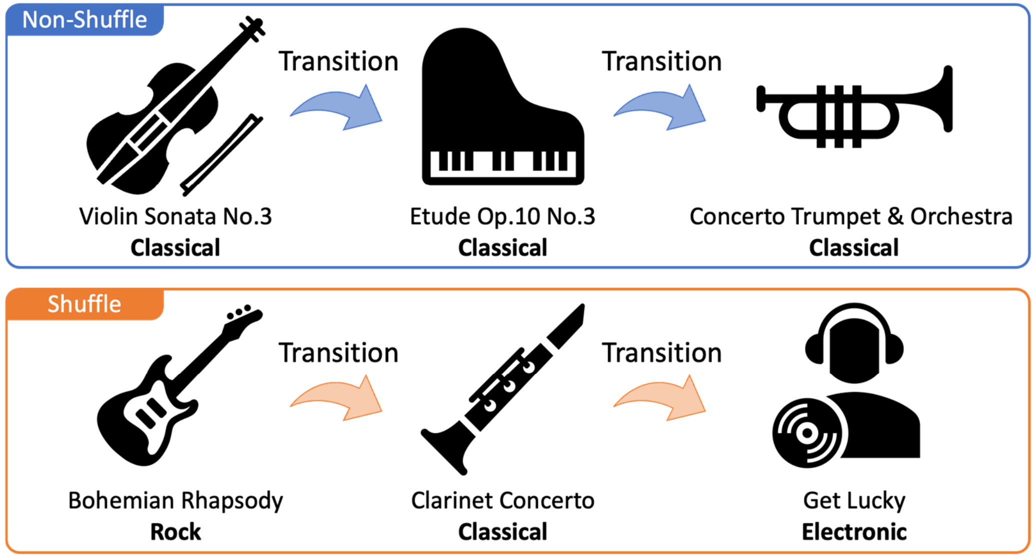


(c) Performance on each play mode

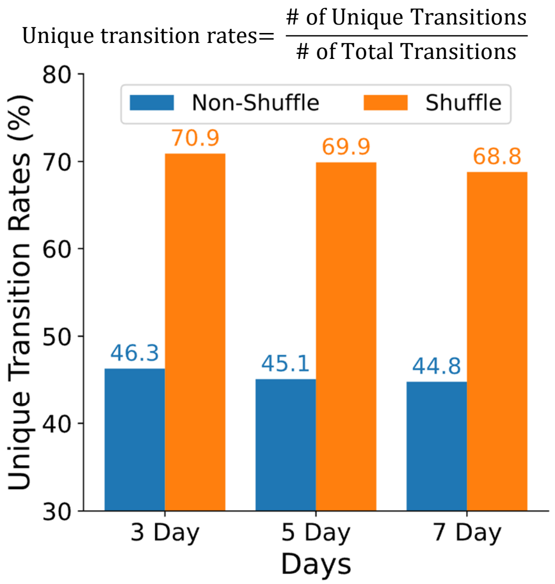
[2] T. W. Leong, F. Vetere, and S. Howard. The serendipity shuffle, *In Proceedings of the 17th Australia conference on Computer-Human Interaction: Citizens Online: Considerations for Today and the Future.* 2015.

MOTIVATION UNIQUE TRANSITION

- Why Shuffle Play is a bottleneck?
  - **High Unique Transition Rate**
    - 1.5 times higher than non-shuffle play
    - transition between tracks that appears only once
  - Track sequences could shift dramatically in shuffle play session



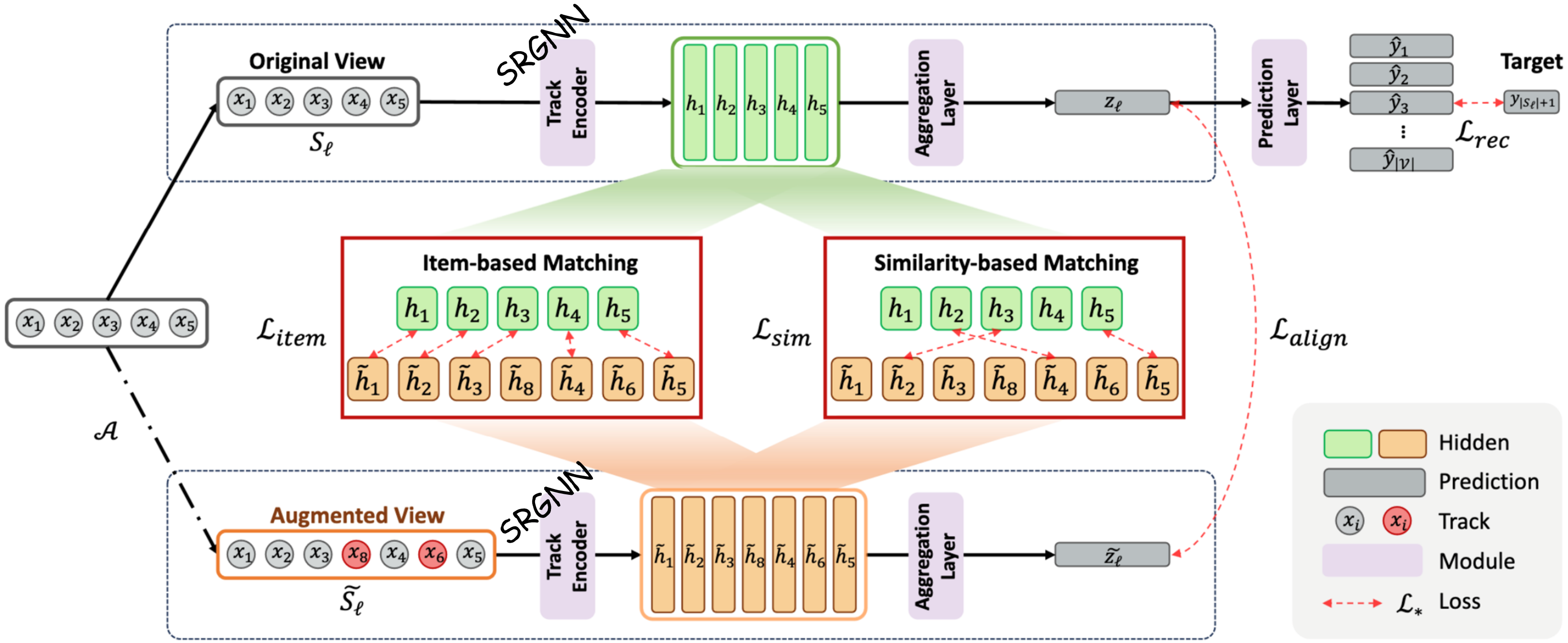
(a) Item Transition



(b) Unique Transition Comparison

MUSE Music Recommender System with Shuffle Play Recommendation Enhancement

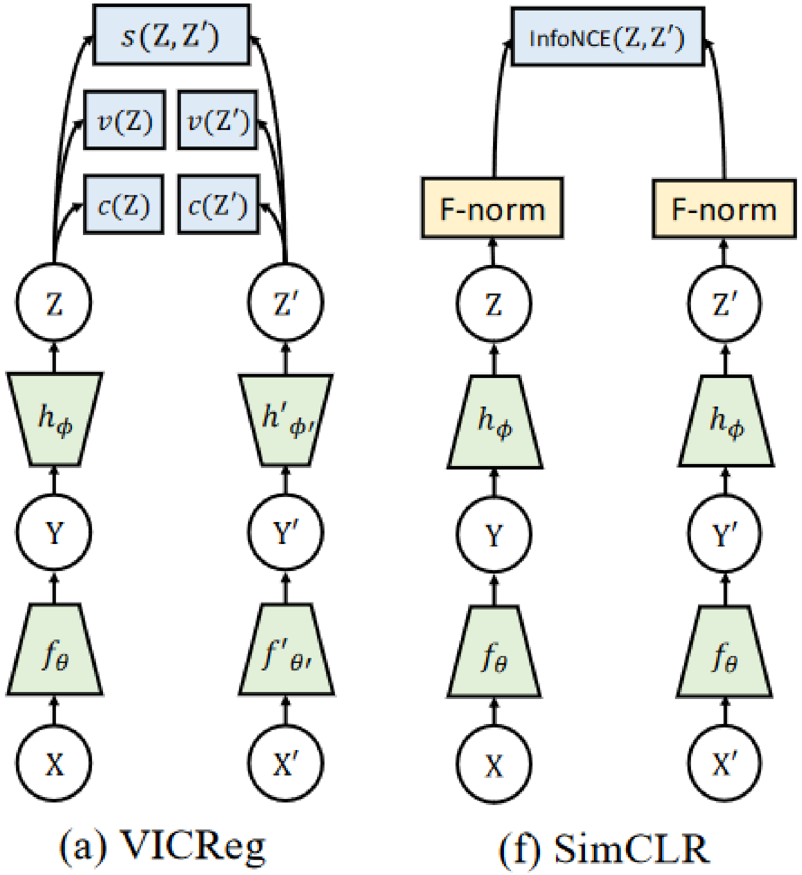
- To **tackle** the inherent challenges posed by **shuffle play** session
  - Transition-based Augmentation (Shuffle play session) / Reordering-based Augmentation (Non-shuffle play session)
  - Fine-grained matching strategies
    - Item-based matching
    - Similarity-based matching



Overall Architecture of **MUSE**

MUSE Music Recommender System with Shuffle Play Recommendation Enhancement

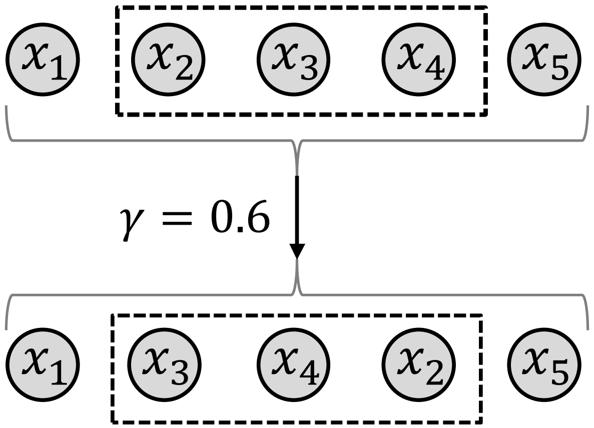
- **Self-supervised Learning (SSL)**
  - Joint Embedding Architecture with Augmentation
  - Maximize the agreement between different views
  - To prevent collapse
    - Contrastive methods (e.g., SimCLR, SimSiam, BYOL)
    - Information maximization method (e.g., Barlow Twins)
    - Regularization (e.g., VICReg [1])



Self-Supervised Learning Frameworks

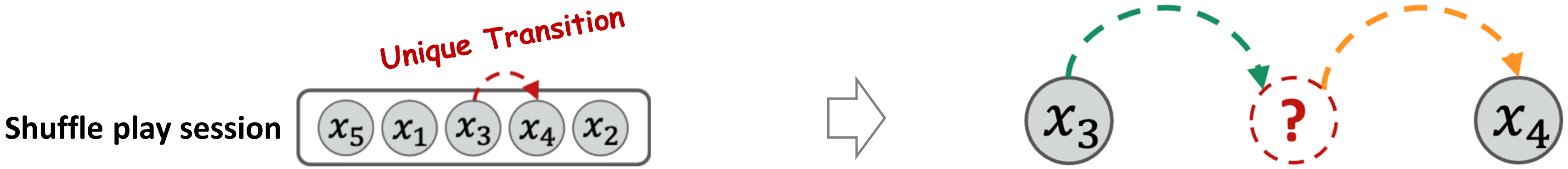
[1] [ICLR22] VICReg: Variance-Invariance-Covariance Regularization for Self-Supervised Learning

MUSE Music Recommender System with Shuffle Play Recommendation Enhancement

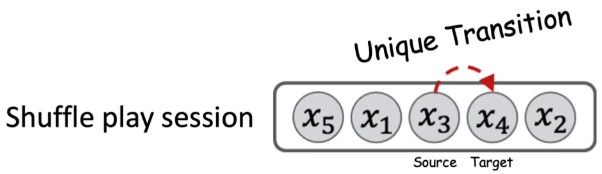


Reordering-based augmentation

- **Transition-based Augmentation**
  - Enrich the sequential information in a given shuffle play session
  - Mitigate the unique transition patterns inherent in shuffle play sessions

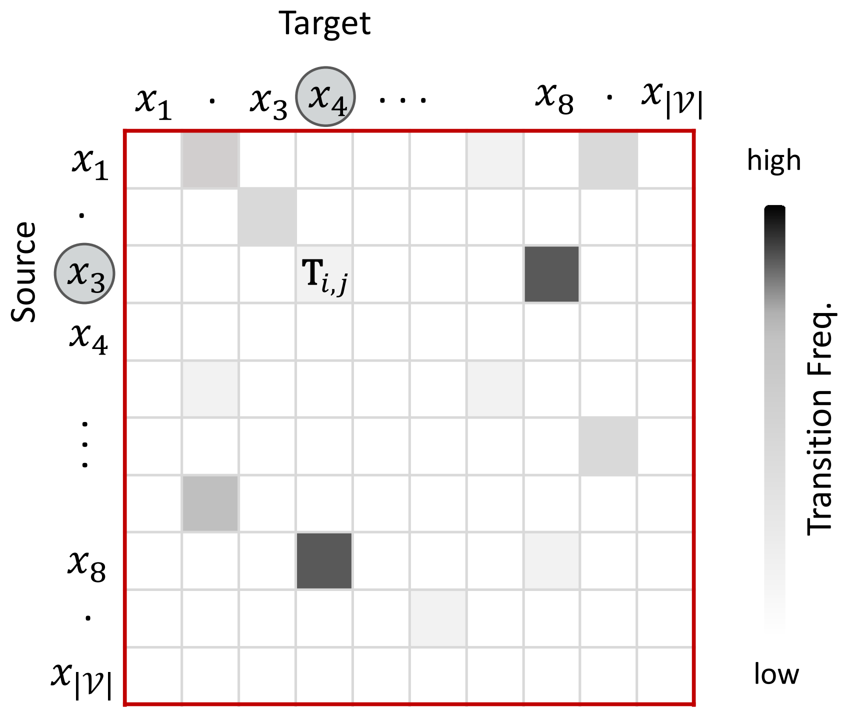


MUSE Music Recommender System with Shuffle Play Recommendation Enhancement

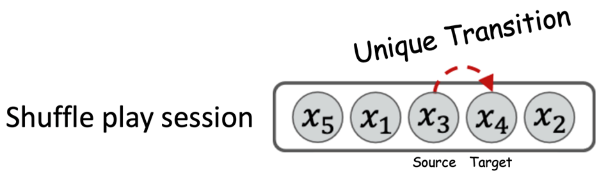


- **Transition-based Augmentation**
  - Enrich the sequential information in a given shuffle play session
  - Mitigate the unique transition patterns inherent in shuffle play sessions
    - Consider the transition frequency between items from all the sessions

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



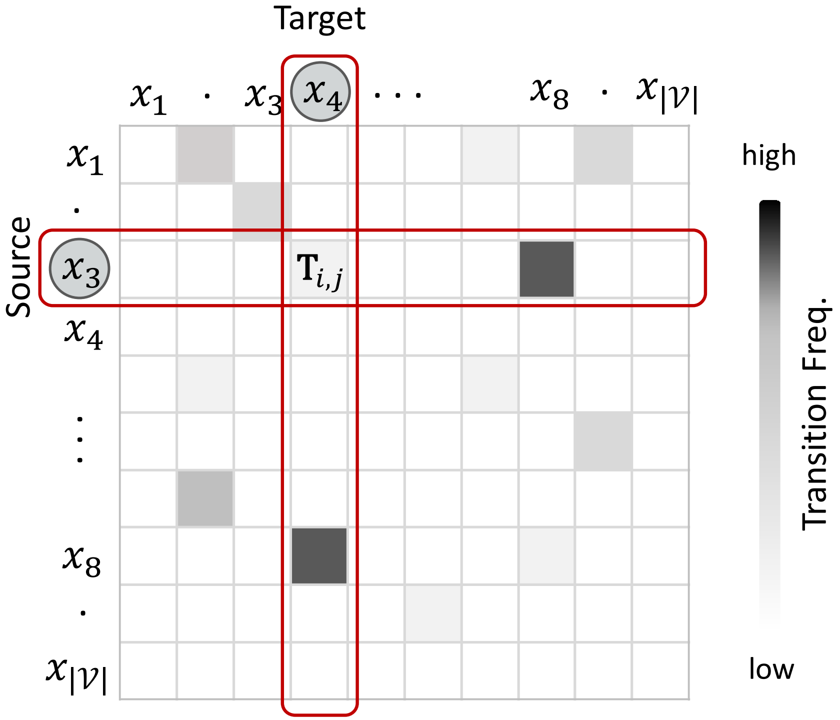
Transition Frequency Matrix



- **Transition-based Augmentation**
  - Enrich the sequential information in a given shuffle play session
  - Mitigate the unique transition patterns inherent in shuffle play sessions
    - Consider the transition frequency between items from all the sessions
    - Normalization - transition matrix in terms of the probability distribution matrix

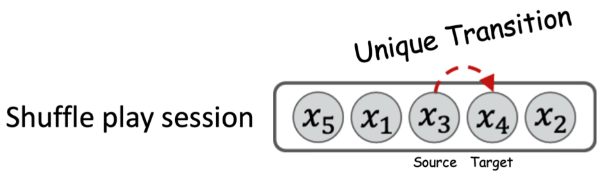
Column-wise Row-wise

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

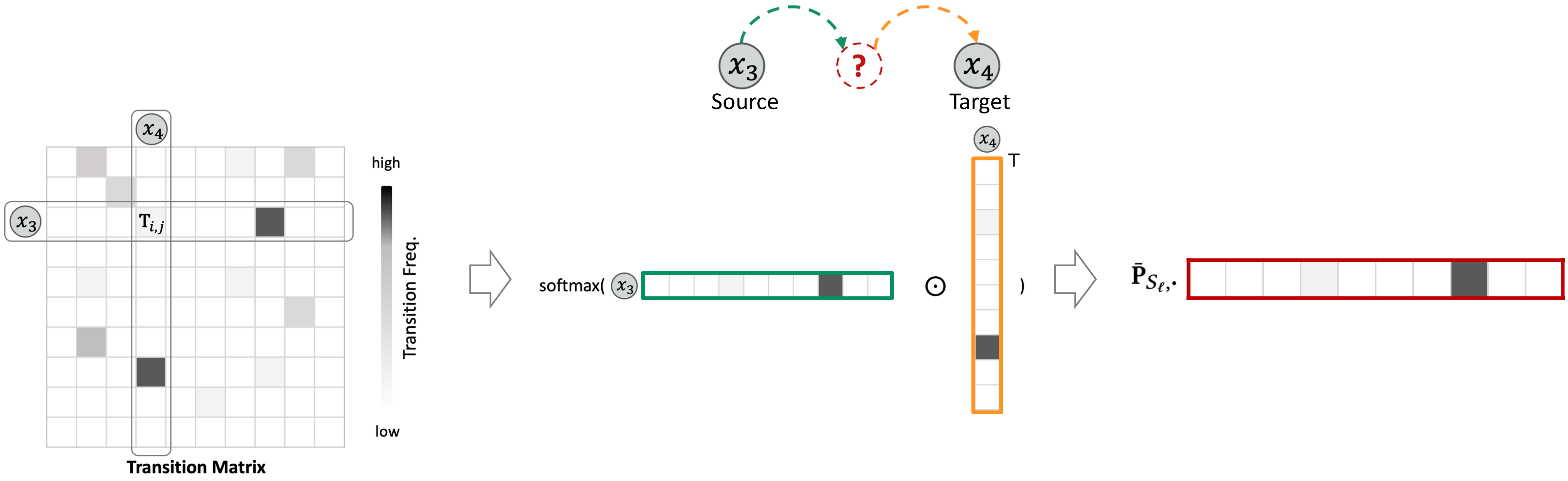


Transition Matrix

MUSE Music Recommender System with Shuffle Play Recommendation Enhancement



- **Transition-based Augmentation**
  - Enrich the sequential information in a given shuffle play session
  - Mitigate the unique transition patterns inherent in shuffle play sessions
    - Consider its back-and-forth context, i.e., source and target $\bar{\mathbf{P}}_{S_\ell,\cdot} = \mathrm{softmax}(\bar{\mathbf{T}}_{S_\ell^s,\cdot} \odot \bar{\mathbf{T}}^\top_{\cdot,S_\ell^t})$

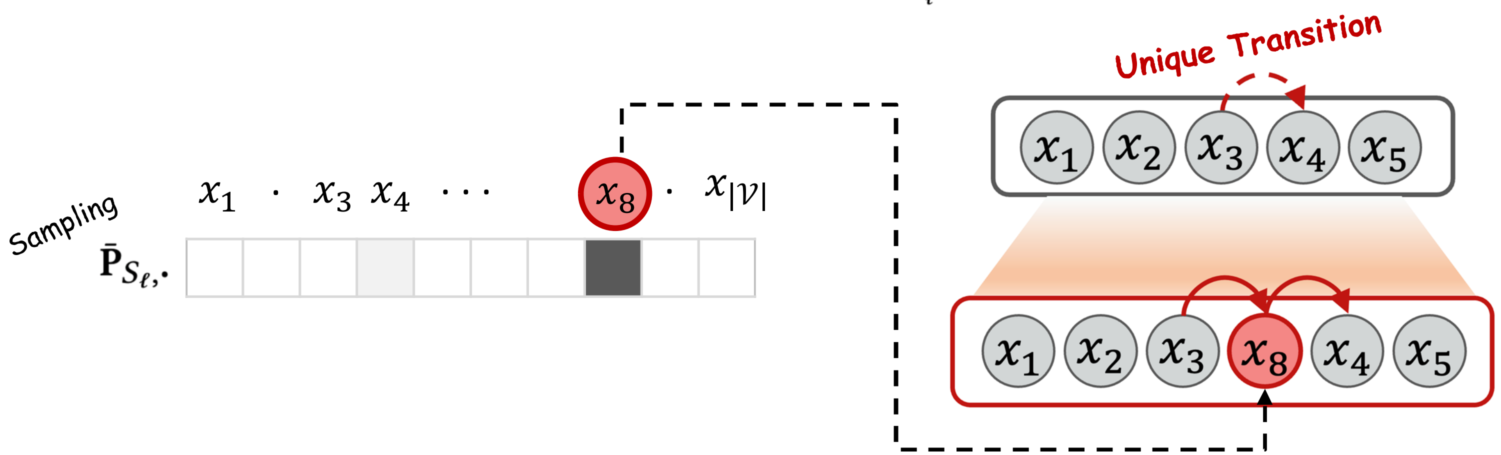


MUSE Music Recommender System with Shuffle Play Recommendation Enhancement

- **Transition-based Augmentation**
  - Enrich the sequential information in a given shuffle play session
  - Mitigate the unique transition patterns inherent in shuffle play sessions
    - Considering its back-and-forth context (i.e., source and target)
    - Insert frequently appearing transitions that could potentially exist in a session

$$\mathbf{c}_i = \begin{cases} \text{Multinomial}(\bar{\mathbf{P}}_{S_\ell}[i,:]), & \text{if } \text{sum}(\bar{\mathbf{P}}_{S_\ell}[i,:]) > 0 \\ \emptyset, & \text{otherwise} \end{cases}, \forall i \leq |S_\ell| - 1$$

$$\text{where } \bar{\mathbf{P}}_{S_\ell,\cdot} = \text{softmax}(\bar{\mathbf{T}}_{S_\ell^s,\cdot} \odot \bar{\mathbf{T}}^\top_{\cdot,S_\ell^t})$$



MUSE Music Recommender System with Shuffle Play Recommendation Enhancement

- **Item-based Matching**
  - To make the encoder to be invariant to augmentations
    - Align the two views' hidden representations derived from the same items

$$\mathcal{L}_{item} = \frac{1}{|\mathbf{I}_\ell|} \sum_{x_t \in \mathbf{I}_\ell} \sum_{x_k \in \tilde{\mathbf{I}}_\ell} \mathbb{1}(x_t = x_k) \|\mathbf{h}_t - \tilde{\mathbf{h}}_k\|^2$$

- **Similarity-based Matching**
  - To supplement item-based matching
    - Align representations of similar items
      - Nearest Neighbor based on $l2$-distance

$$\mathcal{L}_{sim} = \sum_{(\mathbf{h}_i, \mathrm{NN}(\mathbf{h}_i, \tilde{\mathbf{H}}_\ell)) \in \mathcal{P}^K} \|\mathbf{h}_i - \mathrm{NN}(\mathbf{h}_i, \tilde{\mathbf{H}}_\ell)\|^2 + \sum_{(\tilde{\mathbf{h}}_i, \mathrm{NN}(\tilde{\mathbf{h}}_i, \mathbf{H}_\ell)) \in \tilde{\mathcal{P}}^K} \|\tilde{\mathbf{h}}_i - \mathrm{NN}(\tilde{\mathbf{h}}_i, \mathbf{H}_\ell)\|^2$$

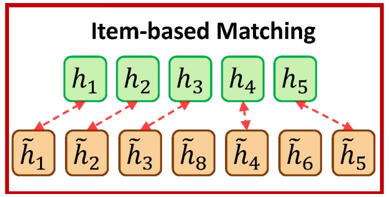
$$\text{where } \mathcal{P}(\mathbf{H}_\ell, \tilde{\mathbf{H}}_\ell) = \{ (\mathbf{h}_i, \mathrm{NN}(\mathbf{h}_i, \tilde{\mathbf{H}}_\ell)) \mid \mathbf{h}_i \in \mathbf{H}_\ell \}$$

- **Regularization**
  - To avoid the representation collapse problem
    - inspired by VICReg

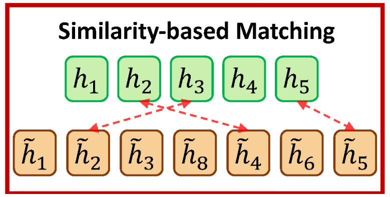
$$\mathcal{L}_{VICReg} = \lambda \cdot s(\mathbf{H}_\ell, \tilde{\mathbf{H}}_\ell) + \mu[v(\mathbf{H}_\ell) + v(\tilde{\mathbf{H}}_\ell)] + \nu[c(\mathbf{H}_\ell) + c(\tilde{\mathbf{H}}_\ell)]$$

$$\mathcal{L}_{matching} = \mathcal{L}_{item} + \mathcal{L}_{sim} + \mathcal{L}_{VICReg}$$

$\mathcal{L}_{item}$



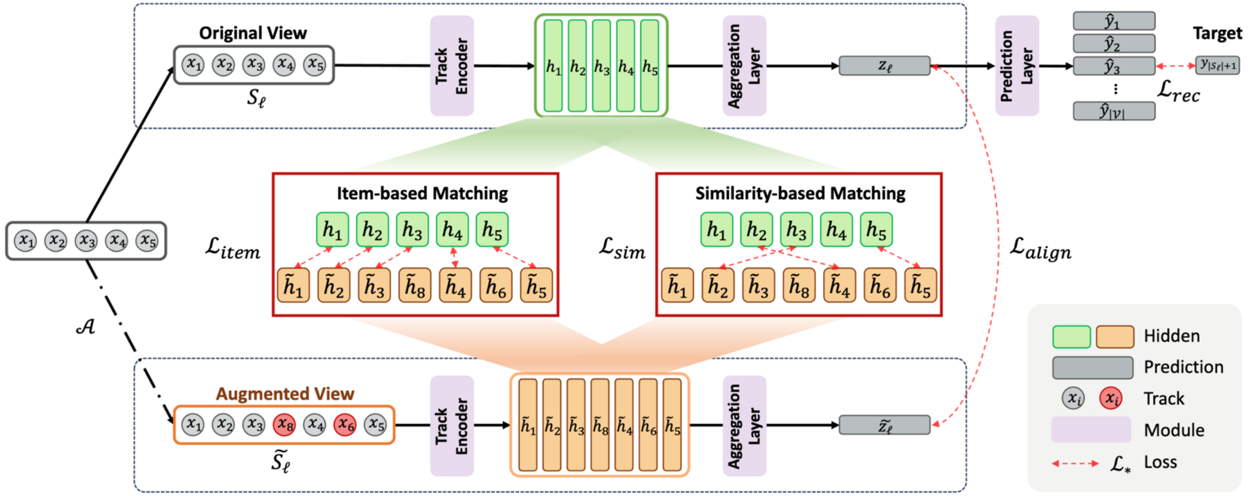
$\mathcal{L}_{sim}$



MUSE Music Recommender System with Shuffle Play Recommendation Enhancement

- **Aggregation Layer**
  - Local embedding: $\mathbf{z}_\ell^{(\text{local})} = \mathbf{h}_{|S_\ell|}$
  - Global embedding (soft-attention): $\mathbf{z}_\ell^{(\text{global})} = \sum_i^{|S_\ell|} \beta_i \mathbf{h}_i,\ \beta_i = \mathbf{W}_1^T \sigma(\mathbf{W}_2 \mathbf{h}_i + \mathbf{W}_3 \mathbf{h}_{|S_\ell|} + \mathbf{b})$
  - $\mathbf{z}_\ell = \mathbf{W}_4(\mathbf{z}_\ell^{(\text{local})} \oplus \mathbf{z}_\ell^{(\text{global})})$
  - Alignment of Self-Supervised Learning $\mathcal{L}_{align} = \lambda \cdot s(\mathbf{z}_\ell, \tilde{\mathbf{z}}_\ell) + \mu[v(\mathbf{z}_\ell) + v(\tilde{\mathbf{z}}_\ell)] + \nu[c(\mathbf{z}_\ell) + c(\tilde{\mathbf{z}}_\ell)]$
- **Prediction Layer**
  - To recommend top-*K* tracks for each session $\hat{\mathbf{y}} = \text{softmax}(\mathbf{z}_\ell^T \mathbf{e}_i)$

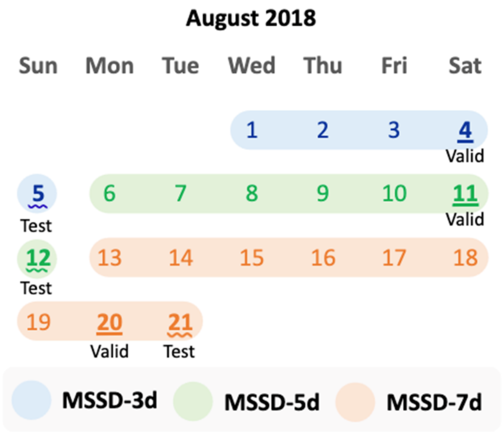
$$\mathcal{L}_{rec} = -\sum_{i=1}^{|\mathcal{V}|} \mathbf{y}_i \log(\hat{\mathbf{y}}_i) + (1 - \mathbf{y}_i) \log(1 - \hat{\mathbf{y}}_i)$$



$$\mathcal{L}_{\text{final}} = \alpha \mathcal{L}_{matching} + (1 - \alpha)\mathcal{L}_{align} + \mathcal{L}_{rec}$$

EXPERIMENTS SETTING

- **Dataset**: Music Streaming Sessions Dataset from Spotify [3]
  - 160 million listening sessions with 20 billion plays, accompanied by user actions
  - Select data belonging to a few days as adopted in a conventional work [4]
    - used partial data due to its large size



(a) Day split

- **Preprocessing**
  - Filter out non-premium users, cold-start items ($\mathrm{frequency} \leq 5$), and short session ($\mathrm{len(session)} \leq 1$)
  - $S = [x_1, x_2, \ldots, x_{|S|}, x_{|S|+1}] \rightarrow ([x_1], x_2), ([x_1, x_2], x_3), \ldots, ([x_1, x_2, \ldots, x_{|S|}], x_{|S|+1})$
    - where $([*], \cdot)$ denotes a input squence $*$ and target $\cdot$ (target must be listened by the user)
    - Especially, input in Shuffle play must be listened by the user
      - $S_\ell^{(Shuffle)} = [x_1, x_2, x_3, x_4, x_5] \rightarrow ([x_1], x_3), ([x_1, x_3], x_5)$
        [listen, skip, listen, skip, listen]

Statistics	MSSD-3d	MSSD-5d	MSSD-7d
# of plays	11,858,262	16,701,958	19,366,448
# of shuffle play sessions	301,814	422,221	501,875
# of non-shuffle play sessions	442,726	618,701	713,300
# of training sessions	613,308	909,818	1,061,274
# of test sessions	131,232	131,104	153,901
# of tracks	199,177	253,693	280,079
Average length	15.93	16.05	15.94

(b) Statistics of datasets

[3] [WWW19] The Music Streaming Sessions Dataset
[4] [CIKM17] Neural attentive session-based recommendation

EXPERIMENTS OVERALL PERFORMANCE

- **MUSE** achieves **state-of-the-art performance** in the real-world, large-scale dataset (i.e., MSSD)
  - **MUSE** significantly outperforms backbone, i.e., SRGNN, due to SSL framework with transition-based augmentation
  - **MUSE** significantly surpasses other SSL approaches due to fine-grained matching strategies
- Graph-based methods, e.g., SRGNN and GCSAN, show relatively high performance
  - Utilize the transition between tracks by constructing graphs
- CoSeRNN deteriorate due to the dependence on contextual information which is exclusive
  - e.g., device type, time since last session

SBR Setting		Attention		Graph		SSL		Music	Ours		
Dataset	Metric	NARM	SASRec	SRGNN	GCSAN	CL4SRec	DuoRec	CoSeRNN	MUSE	$\Delta_{Backbone}$	Δ_{SOTA}
MSSD 5d	R@5	0.3394 (0.0016)	0.3350 (0.0017)	0.3529 (0.0010)	0.3562 (0.0012)	0.3352 (0.0016)	0.3378 (0.0020)	0.3159 (0.0020)	**0.3636*** (0.0005)	3.03%	2.08%
	R@10	0.3941 (0.0032)	0.3891 (0.0021)	0.4040 (0.0020)	0.4065 (0.0015)	0.3886 (0.0019)	0.3926 (0.0026)	0.3747 (0.0012)	**0.4153*** (0.0008)	2.80%	2.16%
	M@5	0.2764 (0.0005)	0.2701 (0.0014)	0.2899 (0.0007)	0.2939 (0.0011)	0.2711 (0.0010)	0.2717 (0.0015)	0.2476 (0.0023)	**0.2993*** (0.0006)	3.24%	1.84%
	M@10	0.2836 (0.0007)	0.2772 (0.0013)	0.2967 (0.0007)	0.3006 (0.0011)	0.2781 (0.0010)	0.2790 (0.0015)	0.2554 (0.0022)	**0.3062*** (0.0005)	3.20%	1.86%
	N@5	0.2920 (0.0008)	0.2863 (0.0014)	0.3056 (0.0006)	0.3094 (0.0011)	0.2870 (0.0011)	0.2882 (0.0016)	0.2646 (0.0022)	**0.3154*** (0.0005)	3.21%	1.94%
	N@10	0.3096 (0.0012)	0.3037 (0.0013)	0.3221 (0.0008)	0.3257 (0.0011)	0.3042 (0.0011)	0.3059 (0.00118)	0.2836 (0.0019)	**0.3320*** (0.0004)	3.07%	1.93%

* indicates a paired t-test results with $p < 0.01$

EXPERIMENTS OVERALL PERFORMANCE

- **MUSE** achieves **state-of-the-art performance** in the real-world, large-scale dataset (i.e., MSSD)
  - **MUSE** significantly outperforms backbone, i.e., SRGNN, due to SSL framework with transition-based augmentation
  - **MUSE** significantly surpasses other SSL approaches due to fine-grained matching strategies
- Graph-based methods, e.g., SRGNN and GCSAN, show relatively high performance
  - Utilize the transition between tracks by constructing graphs
- CoSeRNN deteriorate due to the dependence on contextual information which is exclusive
  - e.g., device type, time since last session

SBR Setting		Attention		Graph		SSL		Music	**Ours**		
Dataset	Metric	NARM	SASRec	SRGNN	GCSAN	CL4SRec	DuoRec	CoSeRNN	**MUSE**	$\Delta_{Backbone}$	Δ_{SOTA}
MSSD 5d	R@5	0.3394 (0.0016)	0.3350 (0.0017)	0.3529 (0.0010)	0.3562 (0.0012)	0.3352 (0.0016)	0.3378 (0.0020)	0.3159 (0.0020)	**0.3636*** (0.0005)	3.03%	2.08%
	R@10	0.3941 (0.0032)	0.3891 (0.0021)	0.4040 (0.0020)	0.4065 (0.0015)	0.3886 (0.0019)	0.3926 (0.0026)	0.3747 (0.0012)	**0.4153*** (0.0008)	2.80%	2.16%
	M@5	0.2764 (0.0005)	0.2701 (0.0014)	0.2899 (0.0007)	0.2939 (0.0011)	0.2711 (0.0010)	0.2717 (0.0015)	0.2476 (0.0023)	**0.2993*** (0.0006)	3.24%	1.84%
	M@10	0.2836 (0.0007)	0.2772 (0.0013)	0.2967 (0.0007)	0.3006 (0.0011)	0.2781 (0.0010)	0.2790 (0.0015)	0.2554 (0.0022)	**0.3062*** (0.0005)	3.20%	1.86%
	N@5	0.2920 (0.0008)	0.2863 (0.0014)	0.3056 (0.0006)	0.3094 (0.0011)	0.2870 (0.0011)	0.2882 (0.0016)	0.2646 (0.0022)	**0.3154*** (0.0005)	3.21%	1.94%
	N@10	0.3096 (0.0012)	0.3037 (0.0013)	0.3221 (0.0008)	0.3257 (0.0011)	0.3042 (0.0011)	0.3059 (0.00118)	0.2836 (0.0019)	**0.3320*** (0.0004)	3.07%	1.93%

* indicates a paired t-test results with $p < 0.01$

EXPERIMENTS OVERALL PERFORMANCE

- MUSE achieves **state-of-the-art performance** in the real-world, large-scale dataset (i.e., MSSD)
  - MUSE significantly outperforms backbone, i.e., SRGNN, due to SSL framework with transition-based augmentation
  - MUSE significantly surpasses other SSL approaches due to fine-grained matching strategies
- Graph-based methods, e.g., SRGNN and GCSAN, show relatively high performance
  - Utilize the transition between tracks by constructing graphs
- CoSeRNN deteriorate due to the dependence on contextual information which is exclusive
  - e.g., device type, time since last session

SBR Setting		Attention		Graph		SSL		Music	**Ours**		
Dataset	Metric	NARM	SASRec	SRGNN	GCSAN	CL4SRec	DuoRec	CoSeRNN	**MUSE**	$\Delta_{Backbone}$	Δ_{SOTA}
MSSD 5d	R@5	0.3394 (0.0016)	0.3350 (0.0017)	0.3529 (0.0010)	0.3562 (0.0012)	0.3352 (0.0016)	0.3378 (0.0020)	0.3159 (0.0020)	**0.3636*** (0.0005)	3.03%	2.08%
	R@10	0.3941 (0.0032)	0.3891 (0.0021)	0.4040 (0.0020)	0.4065 (0.0015)	0.3886 (0.0019)	0.3926 (0.0026)	0.3747 (0.0012)	**0.4153*** (0.0008)	2.80%	2.16%
	M@5	0.2764 (0.0005)	0.2701 (0.0014)	0.2899 (0.0007)	0.2939 (0.0011)	0.2711 (0.0010)	0.2717 (0.0015)	0.2476 (0.0023)	**0.2993*** (0.0006)	3.24%	1.84%
	M@10	0.2836 (0.0007)	0.2772 (0.0013)	0.2967 (0.0007)	0.3006 (0.0011)	0.2781 (0.0010)	0.2790 (0.0015)	0.2554 (0.0022)	**0.3062*** (0.0005)	3.20%	1.86%
	N@5	0.2920 (0.0008)	0.2863 (0.0014)	0.3056 (0.0006)	0.3094 (0.0011)	0.2870 (0.0011)	0.2882 (0.0016)	0.2646 (0.0022)	**0.3154*** (0.0005)	3.21%	1.94%
	N@10	0.3096 (0.0012)	0.3037 (0.0013)	0.3221 (0.0008)	0.3257 (0.0011)	0.3042 (0.0011)	0.3059 (0.00118)	0.2836 (0.0019)	**0.3320*** (0.0004)	3.07%	1.93%

* indicates a paired t-test results with $p < 0.01$

EXPERIMENTS FINE-GRAINED PERFORMANCE

- **MUSE substantially bolsters** the performance on the **shuffle play** sessions
  - Transition-based augmentation and fine-grained matching strategies are beneficial to shuffle play sessions
- **MUSE** boosts the performance on non-shuffle play sessions as well
  - Even though our framework is designed for shuffle play session
- In contrast, the state-of-the-art baseline, GCSAN, is biased towards non-shuffle play session

Setting		Rec. SBR	SSL SBR	Graph-based SBR		Ours	Relative Gap	
Dataset	Metric	FMLP	CL4SRec	SRGNN	GCSAN	**MUSE**	$\Delta_{Backbone}$	Δ_{SOTA}
MSSD 3d	R@10	0.2256 (0.0009)	0.2297 (0.0025)	<u>0.2304</u> (0.0024)	0.2283 (0.0020)	**0.2401*** (0.0015)	4.21%	5.17%
	M@10	0.1071 (0.0008)	0.1080 (0.0014)	<u>0.1140</u> (0.0010)	0.1137 (0.0013)	**0.1181*** (0.0008)	3.60%	3.87%
	N@10	0.1345 (0.0007)	0.1362 (0.0016)	<u>0.1410</u> (0.0013)	0.1402 (0.0014)	**0.1464*** (0.0009)	3.83%	4.42%
MSSD 5d	R@10	0.2265 (0.0011)	0.2250 (0.0015)	<u>0.2330</u> (0.0023)	0.2295 (0.0017)	**0.2400*** (0.0012)	3.00%	4.58%
	M@10	0.1069 (0.0010)	0.1061 (0.0008)	<u>0.1146</u> (0.0010)	0.1136 (0.0010)	**0.1179*** (0.0004)	2.88%	3.79%
	N@10	0.1345 (0.0008)	0.1337 (0.0007)	<u>0.1420</u> (0.0011)	0.1404 (0.0010)	**0.1462*** (0.0003)	2.96%	4.13%

(a) Performance on **Shuffle Play** Session

Setting		Rec. SBR	SSL SBR	Graph-based SBR		Ours	Relative Gap	
Dataset	Metric	FMLP	CL4SRec	SRGNN	GCSAN	**MUSE**	$\Delta_{Backbone}$	Δ_{SOTA}
MSSD 3d	R@10	0.4868 (0.0032)	0.4776 (0.0029)	0.4885 (0.0029)	<u>0.4963</u> (0.0038)	**0.5034*** (0.0037)	3.05%	1.43%
	M@10	0.3728 (0.0026)	0.3620 (0.0032)	0.3841 (0.0034)	<u>0.3943</u> (0.0024)	**0.3992*** (0.0030)	3.93%	1.24%
	N@10	0.4001 (0.0028)	0.3897 (0.0028)	0.4091 (0.0031)	<u>0.4188</u> (0.0026)	**0.4242*** (0.0031)	3.69%	1.29%
MSSD 5d	R@10	0.4872 (0.0017)	0.4724 (0.0021)	0.4916 (0.0025)	<u>0.4972</u> (0.0014)	**0.5051*** (0.0007)	2.75%	1.59%
	M@10	0.3751 (0.0006)	0.3662 (0.0008)	0.3899 (0.0007)	<u>0.3963</u> (0.0008)	**0.4026*** (0.0008)	3.26%	1.59%
	N@10	0.4019 (0.0005)	0.3916 (0.0011)	0.4143 (0.0010)	<u>0.4205</u> (0.0010)	**0.4272*** (0.0007)	3.11%	1.59%

(b) Performance on **Non-Shuffle Play** Session

EXPERIMENTS FINE-GRAINED PERFORMANCE

- **MUSE substantially bolsters** the performance on the **shuffle play** sessions
  - Transition-based augmentation and fine-grained matching strategies are beneficial to shuffle play sessions
- **MUSE** boosts the performance on non-shuffle play sessions as well
  - Even though our framework is designed for shuffle play session
- In contrast, the state-of-the-art baseline, GCSAN, is biased towards non-shuffle play session

Setting		Rec. SBR	SSL SBR	Graph-based SBR		Ours	Relative Gap	
Dataset	Metric	FMLP	CL4SRec	SRGNN	GCSAN	**MUSE**	$\Delta_{Backbone}$	Δ_{SOTA}
MSSD 3d	R@10	0.2256 (0.0009)	0.2297 (0.0025)	0.2304 (0.0024)	0.2283 (0.0020)	**0.2401*** (0.0015)	4.21%	5.17%
	M@10	0.1071 (0.0008)	0.1080 (0.0014)	0.1140 (0.0010)	0.1137 (0.0013)	**0.1181*** (0.0008)	3.60%	3.87%
	N@10	0.1345 (0.0007)	0.1362 (0.0016)	0.1410 (0.0013)	0.1402 (0.0014)	**0.1464*** (0.0009)	3.83%	4.42%
MSSD 5d	R@10	0.2265 (0.0011)	0.2250 (0.0015)	0.2330 (0.0023)	0.2295 (0.0017)	**0.2400*** (0.0012)	3.00%	4.58%
	M@10	0.1069 (0.0010)	0.1061 (0.0008)	0.1146 (0.0010)	0.1136 (0.0010)	**0.1179*** (0.0004)	2.88%	3.79%
	N@10	0.1345 (0.0008)	0.1337 (0.0007)	0.1420 (0.0011)	0.1404 (0.0010)	**0.1462*** (0.0003)	2.96%	4.13%

(a) Performance on **Shuffle Play** Session

Setting		Rec. SBR	SSL SBR	Graph-based SBR		Ours	Relative Gap	
Dataset	Metric	FMLP	CL4SRec	SRGNN	GCSAN	**MUSE**	$\Delta_{Backbone}$	Δ_{SOTA}
MSSD 3d	R@10	0.4868 (0.0032)	0.4776 (0.0029)	0.4885 (0.0029)	0.4963 (0.0038)	**0.5034*** (0.0037)	3.05%	1.43%
	M@10	0.3728 (0.0026)	0.3620 (0.0032)	0.3841 (0.0034)	0.3943 (0.0024)	**0.3992*** (0.0030)	3.93%	1.24%
	N@10	0.4001 (0.0028)	0.3897 (0.0028)	0.4091 (0.0031)	0.4188 (0.0026)	**0.4242*** (0.0031)	3.69%	1.29%
MSSD 5d	R@10	0.4872 (0.0017)	0.4724 (0.0021)	0.4916 (0.0025)	0.4972 (0.0014)	**0.5051*** (0.0007)	2.75%	1.59%
	M@10	0.3751 (0.0006)	0.3662 (0.0008)	0.3899 (0.0007)	0.3963 (0.0008)	**0.4026*** (0.0008)	3.26%	1.59%
	N@10	0.4019 (0.0005)	0.3916 (0.0011)	0.4143 (0.0010)	0.4205 (0.0010)	**0.4272*** (0.0007)	3.11%	1.59%

(b) Performance on **Non-Shuffle Play** Session

EXPERIMENTS FINE-GRAINED PERFORMANCE

- **MUSE substantially bolsters** the performance on the **shuffle play** sessions
  - Transition-based augmentation and fine-grained matching strategies are beneficial to shuffle play sessions
- **MUSE** boosts the performance on non-shuffle play sessions as well
  - Even though our framework is designed for shuffle play session
- In contrast, the state-of-the-art baseline, GCSAN, is biased towards non-shuffle play session

Setting		Rec. SBR	SSL SBR	Graph-based SBR		Ours	Relative Gap	
Dataset	Metric	FMLP	CL4SRec	SRGNN	GCSAN	**MUSE**	$\Delta_{Backbone}$	Δ_{SOTA}
MSSD 3d	R@10	0.2256 (0.0009)	0.2297 (0.0025)	0.2304 (0.0024)	0.2283 (0.0020)	**0.2401*** (0.0015)	4.21%	5.17%
	M@10	0.1071 (0.0008)	0.1080 (0.0014)	0.1140 (0.0010)	0.1137 (0.0013)	**0.1181*** (0.0008)	3.60%	3.87%
	N@10	0.1345 (0.0007)	0.1362 (0.0016)	0.1410 (0.0013)	0.1402 (0.0014)	**0.1464*** (0.0009)	3.83%	4.42%
MSSD 5d	R@10	0.2265 (0.0011)	0.2250 (0.0015)	0.2330 (0.0023)	0.2295 (0.0017)	**0.2400*** (0.0012)	3.00%	4.58%
	M@10	0.1069 (0.0010)	0.1061 (0.0008)	0.1146 (0.0010)	0.1136 (0.0010)	**0.1179*** (0.0004)	2.88%	3.79%
	N@10	0.1345 (0.0008)	0.1337 (0.0007)	0.1420 (0.0011)	0.1404 (0.0010)	**0.1462*** (0.0003)	2.96%	4.13%

(a) Performance on **Shuffle Play** Session

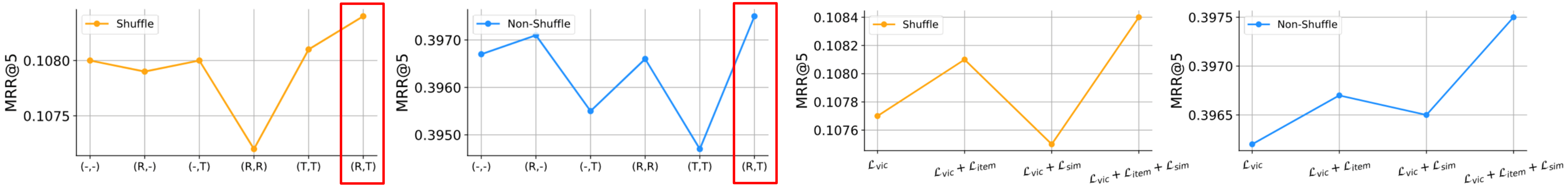
Setting		Rec. SBR	SSL SBR	Graph-based SBR		Ours	Relative Gap	
Dataset	Metric	FMLP	CL4SRec	SRGNN	GCSAN	**MUSE**	$\Delta_{Backbone}$	Δ_{SOTA}
MSSD 3d	R@10	0.4868 (0.0032)	0.4776 (0.0029)	0.4885 (0.0029)	0.4963 (0.0038)	**0.5034*** (0.0037)	3.05%	1.43%
	M@10	0.3728 (0.0026)	0.3620 (0.0032)	0.3841 (0.0034)	0.3943 (0.0024)	**0.3992*** (0.0030)	3.93%	1.24%
	N@10	0.4001 (0.0028)	0.3897 (0.0028)	0.4091 (0.0031)	0.4188 (0.0026)	**0.4242*** (0.0031)	3.69%	1.29%
MSSD 5d	R@10	0.4872 (0.0017)	0.4724 (0.0021)	0.4916 (0.0025)	0.4972 (0.0014)	**0.5051*** (0.0007)	2.75%	1.59%
	M@10	0.3751 (0.0006)	0.3662 (0.0008)	0.3899 (0.0007)	0.3963 (0.0008)	**0.4026*** (0.0008)	3.26%	1.59%
	N@10	0.4019 (0.0005)	0.3916 (0.0011)	0.4143 (0.0010)	0.4205 (0.0010)	**0.4272*** (0.0007)	3.11%	1.59%

(b) Performance on **Non-Shuffle Play** Session

EXPERIMENTS ABLATION STUDY

- **Ablation on Augmentation**
  - Non-shuffle play sessions benefit from re-ordering-based augmentation
    - Mimic the shuffle play session environment
  - Shuffle play sessions especially benefit from transition-based augmentation
    - Mitigate the unique transition pattern inherent in shuffle play session

- **Ablation on Matching**
  - Item-based matching facilitates the alignment of the track embeddings of the identical items between two views
  - Similarity-based matching complements item-based matching by considering the similarity of track representations



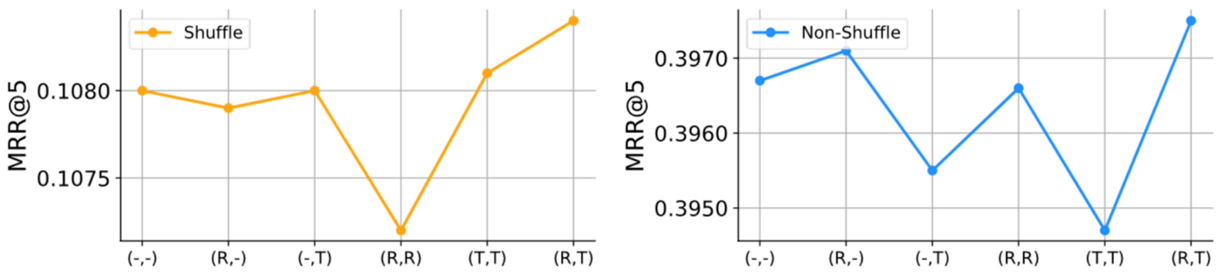
(a) Ablation on Augmentation

(Augmentation to Non-shuffle play, Augmentation to Shuffle play)

(b) Ablation on Matching

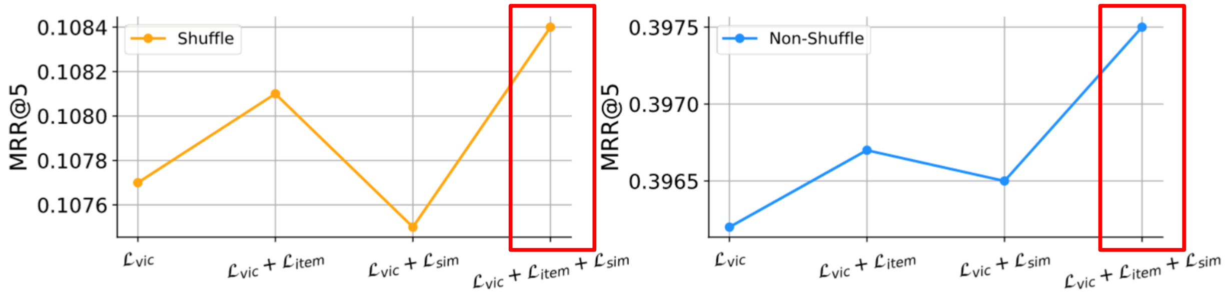
EXPERIMENTS ABLATION STUDY

- **Ablation on Augmentation**
  - Non-shuffle play sessions benefit from re-ordering-based augmentation
    - Mimic the shuffle play session environment
  - Shuffle play sessions especially benefit from transition-based augmentation
    - Mitigate the unique transition pattern inherent in shuffle play session
- **Ablation on Matching**
  - Item-based matching facilitates the alignment of the track embeddings of the identical items between two views
  - Similarity-based matching complements item-based matching by considering the similarity of track representations



(a) Ablation on Augmentation

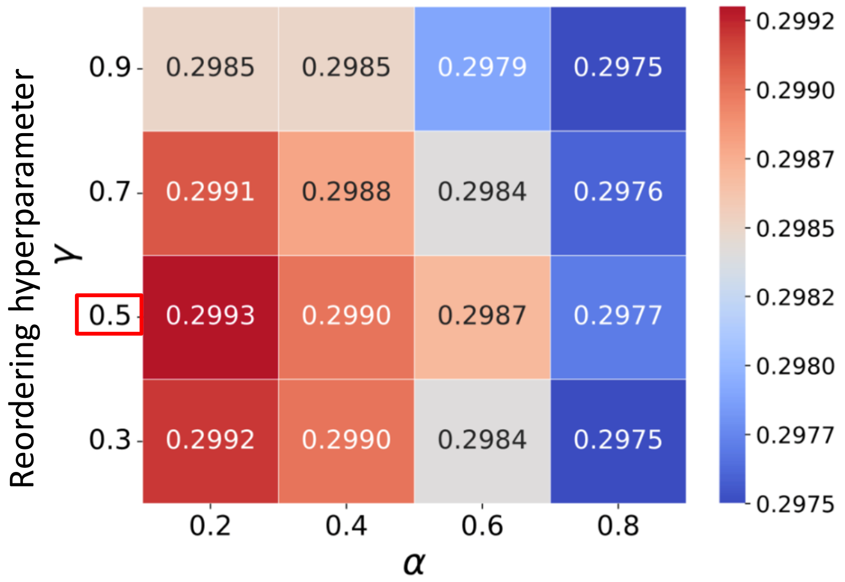
(Augmentation to Non-shuffle play, Augmentation to Shuffle play)



(b) Ablation on Matching

EXPERIMENTS SENSITIVITY ANALYSIS

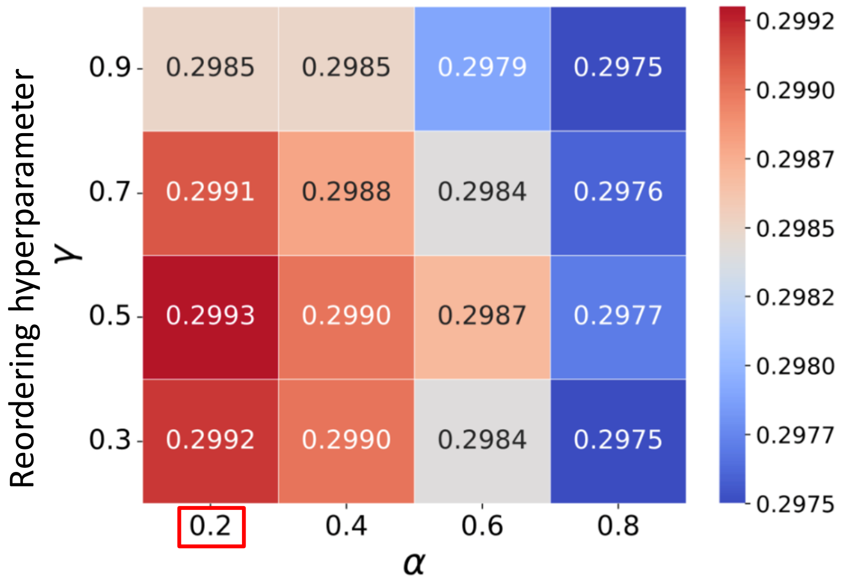
- **Moderate reordering hyperparameter γ** (i.e., 0.5) is advantageous
  - Excessive reordering (i.e., high γ) could hamper the original session's semantic
  - Too little reordering (i.e., low γ) might hinder the augmentation's potential for enhancing generalizability
- **Low loss controlling hyperparameter α**, (i.e., 0.2) is advantageous $\mathcal{L}_{\text{final}} = \alpha\mathcal{L}_{matching} + (1-\alpha)\mathcal{L}_{align} + \mathcal{L}_{rec}$
  - this choice acts effectively as a regularizer, contributing to the overall performance



loss controlling hyperparameter

EXPERIMENTS SENSITIVITY ANALYSIS

- **Moderate reordering hyperparameter γ** (i.e., 0.5) is advantageous
  - Excessive reordering (i.e., high γ) could hamper the original session's semantic
  - Too little reordering (i.e., low γ) might hinder the augmentation's potential for enhancing generalizability
- **Low loss controlling hyperparameter α**, (i.e., 0.2) is advantageous $\mathcal{L}_{\text{final}} = \alpha\mathcal{L}_{matching} + (1-\alpha)\mathcal{L}_{align} + \mathcal{L}_{rec}$
  - this choice acts effectively as a regularizer, contributing to the overall performance



loss controlling hyperparameter

CONCLUSION

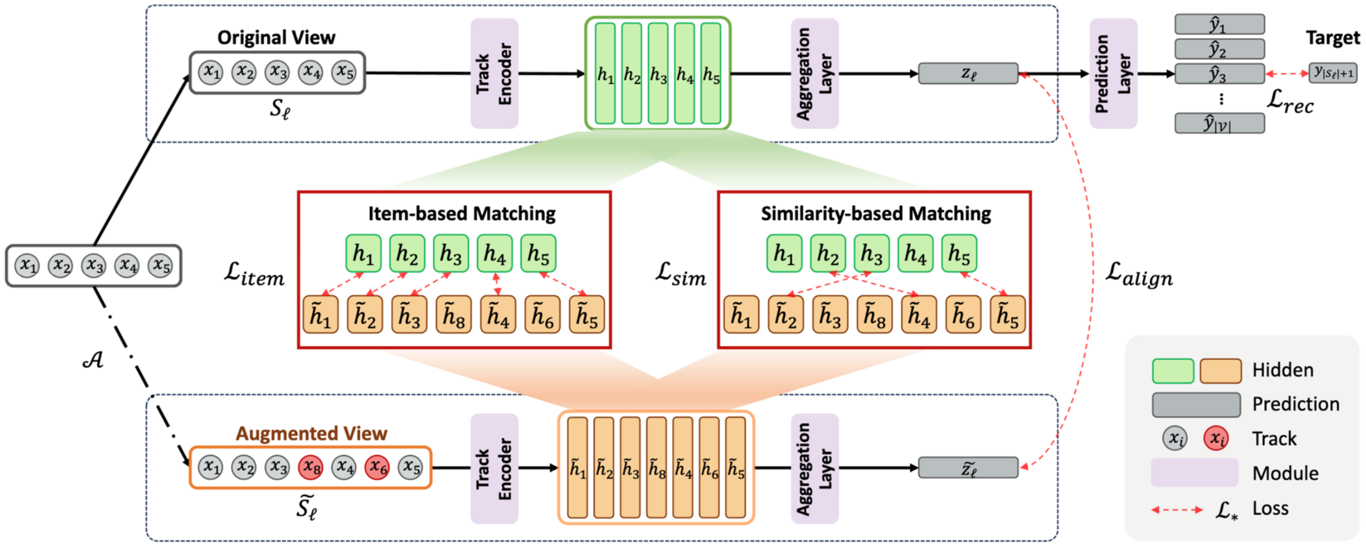
[Full Paper] https://arxiv.org/abs/2308.09649

[Source Code] https://github.com/yunhak0/MUSE

[Author Email] yunhak.oh@kaist.ac.kr

- **The first work** that attempts to enhance prevailing **shuffle-play environments** in the music domain
- **Transition-based augmentation**
  - Mitigate the unique transition pattern inherent in shuffle play session
- **Fine-grained matching strategies**: Item- and Similarity-based Matching
  - Identical items and similar items between the two views to be close in the embedding space
- Demonstrate the superiority of **MUSE** in **a real-world music streaming dataset**

Paper



Code

Appendix

EXPERIMENTS EVALUATION PROTOCOL

- **Recall@K**
  - A measure of completeness, determines the fraction of relevant items retrieved out of all relevant items
- Mean Reciprocal Rank (**MRR@K**)
  - Relevance based on inverse of the rank of the relevant items (hit) in a given list
- Normalized Discounted Cumulative Gain (**NDCG@K**)
  - Relevance applied to logarithmic reduction factor

Predict

Actual	Positive	Negative
Positive	TP	FN
Negative	FP	TN

$$Recall = \frac{TP}{TP + FN} = \frac{|listen\ tracks\ recommended|}{|all\ listen\ tracks|}$$

(a) Recall

Rank	Hit?
1	
2	X
3	X
4	X
5	

$$RR = \frac{1}{2} + \frac{1}{3} + \frac{1}{4}$$

Rank	Hit?
1	X
2	
3	
4	X
5	X

$$RR = 1 + \frac{1}{4} + \frac{1}{5}$$

$$MRR = \frac{1}{|U|} \sum_{u=1}^{|U|} RR(u)$$

$$RR(u) = \sum_{i=1}^{k} \frac{relevance_i}{rank_i}$$

(b) Mean Reciprocal Rank (MRR)

- **Discounted cumulative gain (DCG)**
  - Logarithmic reduction factor

$$DCG_{pos} = rel_1 + \sum_{i=2}^{pos} \frac{rel_i}{\log_2 i}$$

  - pos denotes the position up to which relevance is accumulated
  - rel_i returns the relevance of recommendation at position i

Rank	Hit?
1	
2	X
3	X
4	X
5	

- **Idealized discounted cumulative gain (IDCG)**
  - Assumption that items are ordered by decreasing relevance

$$IDCG_{pos} = rel_1 + \sum_{i=2}^{|h|-1} \frac{rel_i}{\log_2 i}$$

$$DCG_5 = \frac{1}{\log_2 2} + \frac{1}{\log_2 3} + \frac{1}{\log_2 4} = 2.13$$

$$IDCG_5 = 1 + \frac{1}{\log_2 2} + \frac{1}{\log_2 3} = 2.63$$

- **Normalized discounted cumulative gain (nDCG)**
  - Normalized to the interval [0..1]

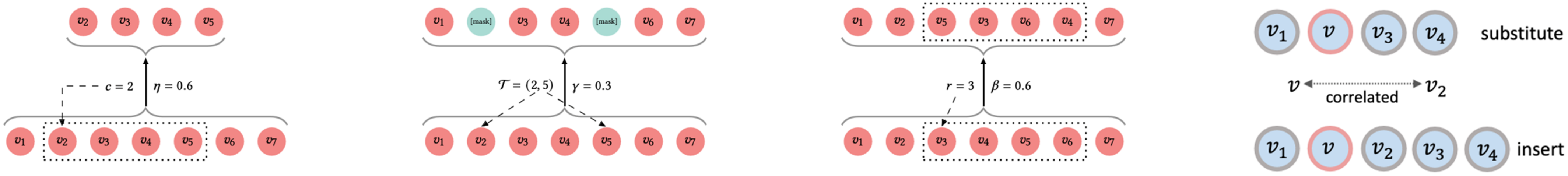
$$nDCG_{pos} = \frac{DCG_{pos}}{IDCG_{pos}}$$

$$nDCG_5 = \frac{DCG_5}{IDCG_5} \approx 0.81$$

(c) Normalized Discounted Cumulative Gain (NDCG)

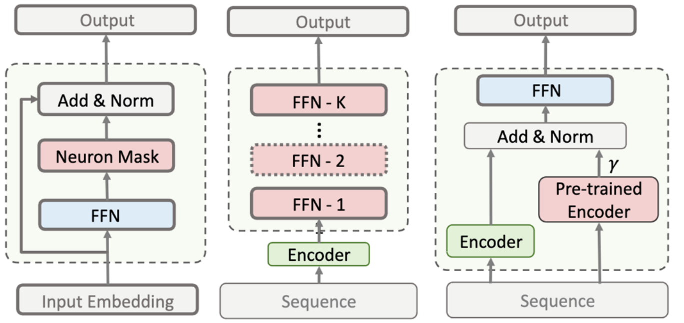
AUGMENTATION SEQUENCE DATA

- **Data** augmentation techniques



(a) item crop (b) item mask (c) item reorder

- **Model** augmentation techniques



(b) Neuron Masking **(c) Layer Dropping** **(d) Encoder Complementing**

[SIGIR21] Contrastive Learning for Sequential Recommendation (CL4SRec)
[CoRR21] Contrastive Self-supervised Sequential Recommendation with Robust Augmentation
[CoRR22] Self-supervised Learning for Sequential Recommendation with Model Augmentation