Using Adversarial Autoencoders for Multi-Modal Automatic Playlist Continuation

Iacopo Vagliano, ZBW Kiel
Lukas Galke, University of Kiel
Florian May, University of Kiel
Ansgar Scherp, University of Stirling

ACM RecSys Challenge, 7 October 2018

Leibniz-Informationszentrum Wirtschaft
Leibniz Information Centre for Economics

Motivation

- Adversarial regularization improves autoencoders on images (Makhzani et al. 2015)
- Adversarial autoencoders effective in recommendation tasks (Galke et al. 2018)
  - Smoothness on the code aids autoencoders to reconstruct highly sparse item vectors

Makhzani, A. et al. (2015). “Adversarial Autoencoders”. In: CoRR abs/1511.05644.
Galke, L. et al. (2018). Multi-Modal Adversarial Autoencoders for Recommendations of Citations and Subject Labels. ACM UMAP.

Motivation

MOVING
www.moving-project.eu

- Adversarial regularization improves autoencoders on images (Makhzani et al. 2015)
- Adversarial autoencoders effective in recommendation tasks (Galke et al. 2018)
  - Smoothness on the code aids autoencoders to reconstruct highly sparse item vectors

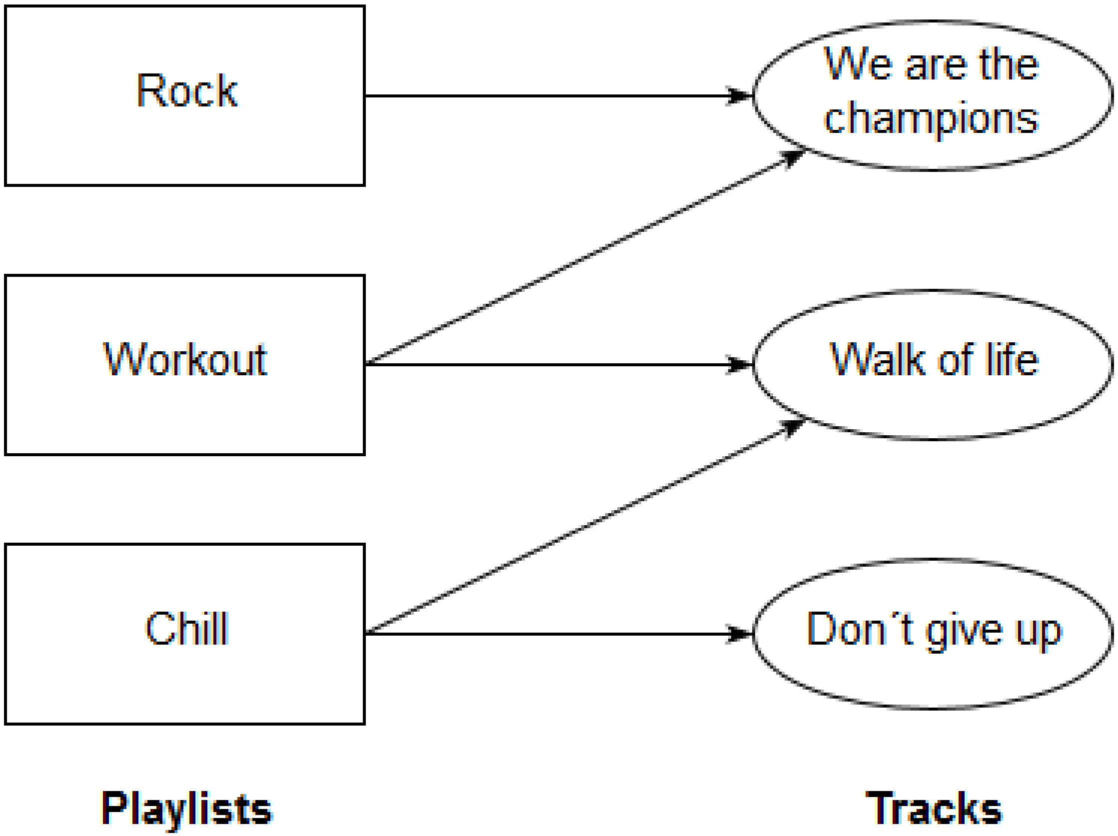
Research Questions

- *Are adversarial autoencoders also effective for automatic playlist continuation?*
- *Is it beneficial aggregating item attributes (track title, album title, artist name)?*

Makhzani, A. et al. (2015). “Adversarial Autoencoders”. In: CoRR abs/1511.05644.
Galke, L. et al. (2018). Multi-Modal Adversarial Autoencoders for Recommendations of Citations and Subject Labels. ACM UMAP.

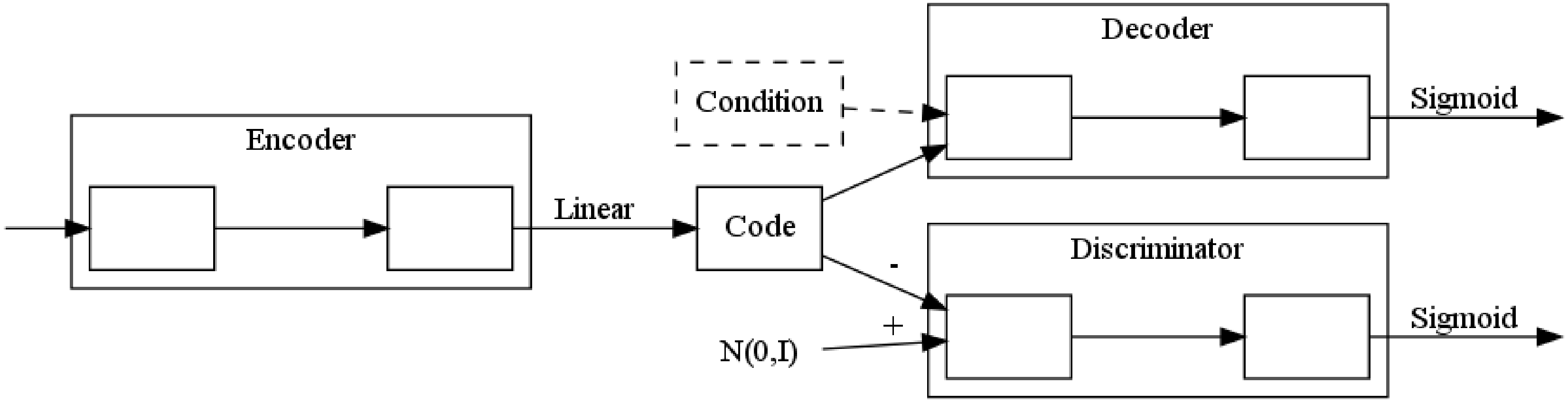
Problem statement

- Set of *m* playlist **P**
- Set of *n* tracks **T**
- Sparse matrix $\mathbf{X} \in \{0,1\}^{m \times n}$ in the spanned space **P** x **T**
  - $X_{jk} = 1$ if the track *k* is in the playlist *j* (binary occurrence)



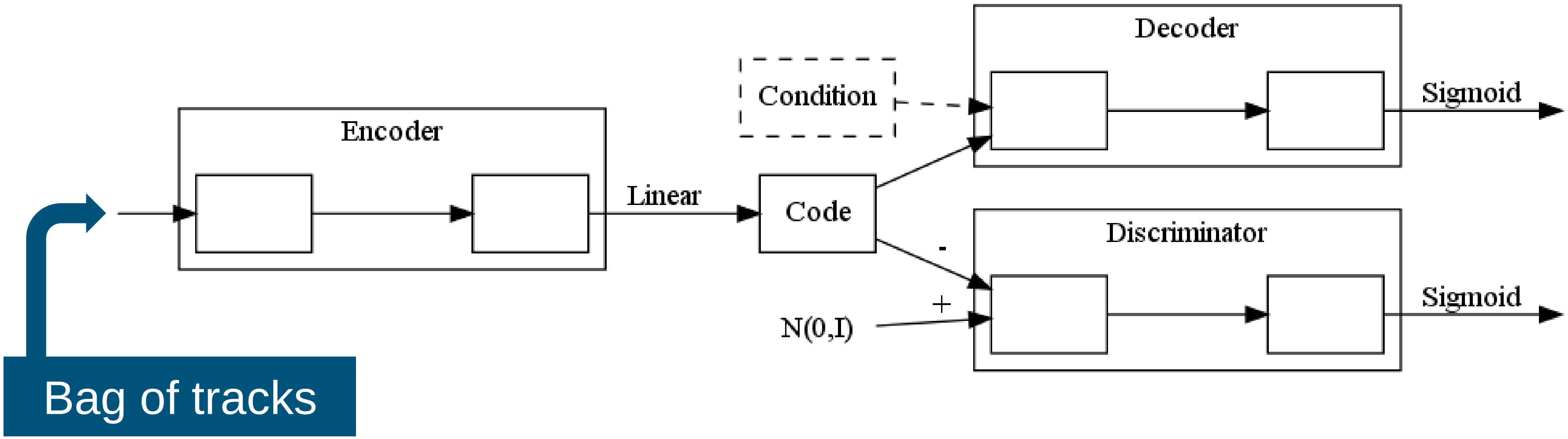
Approach

- Multi-Modal Adversarial Autoencoders (AAE)
  - Train autoencoder on track sets (playlists)
  - Supply condition to the decoder (multi-modal)
  - Match code with a normal distribution for smooth representations (adversarial)



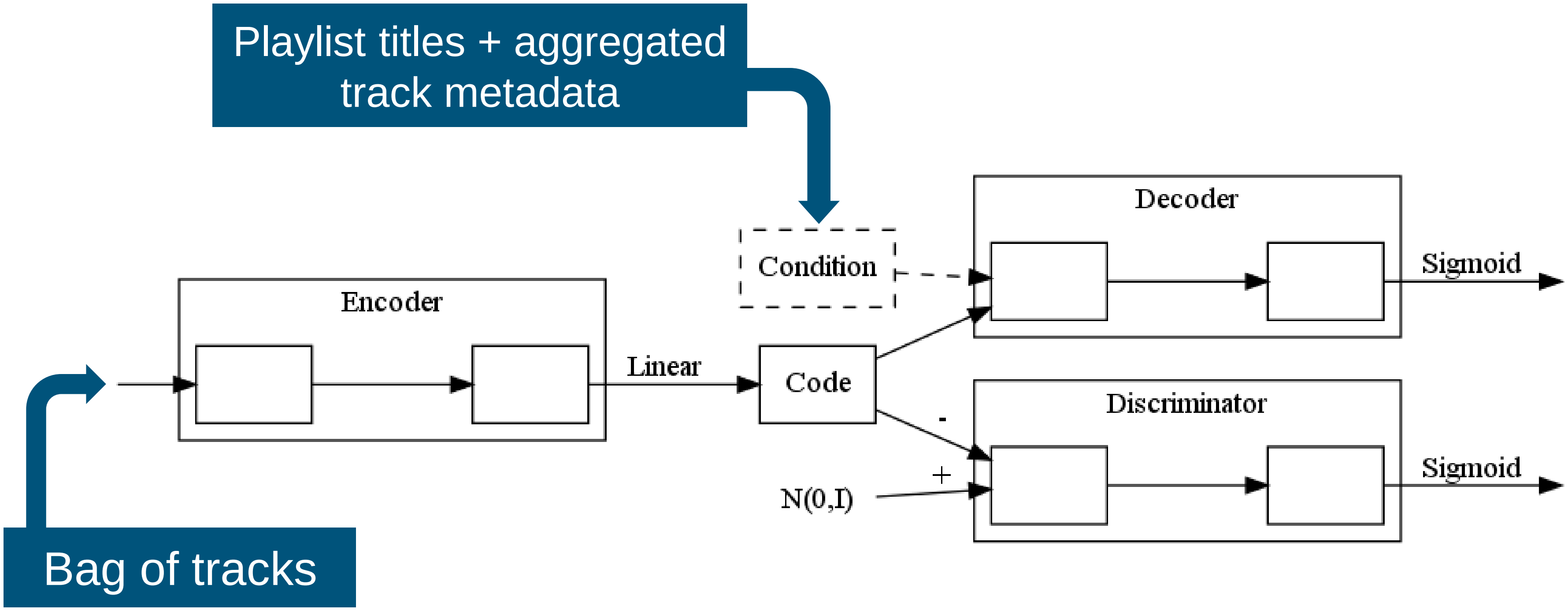
Approach

- Multi-Modal Adversarial Autoencoders (AAE)
  - Train autoencoder on track sets (playlists)
  - Supply condition to the decoder (multi-modal)
  - Match code with a normal distribution for smooth representations (adversarial)



Approach

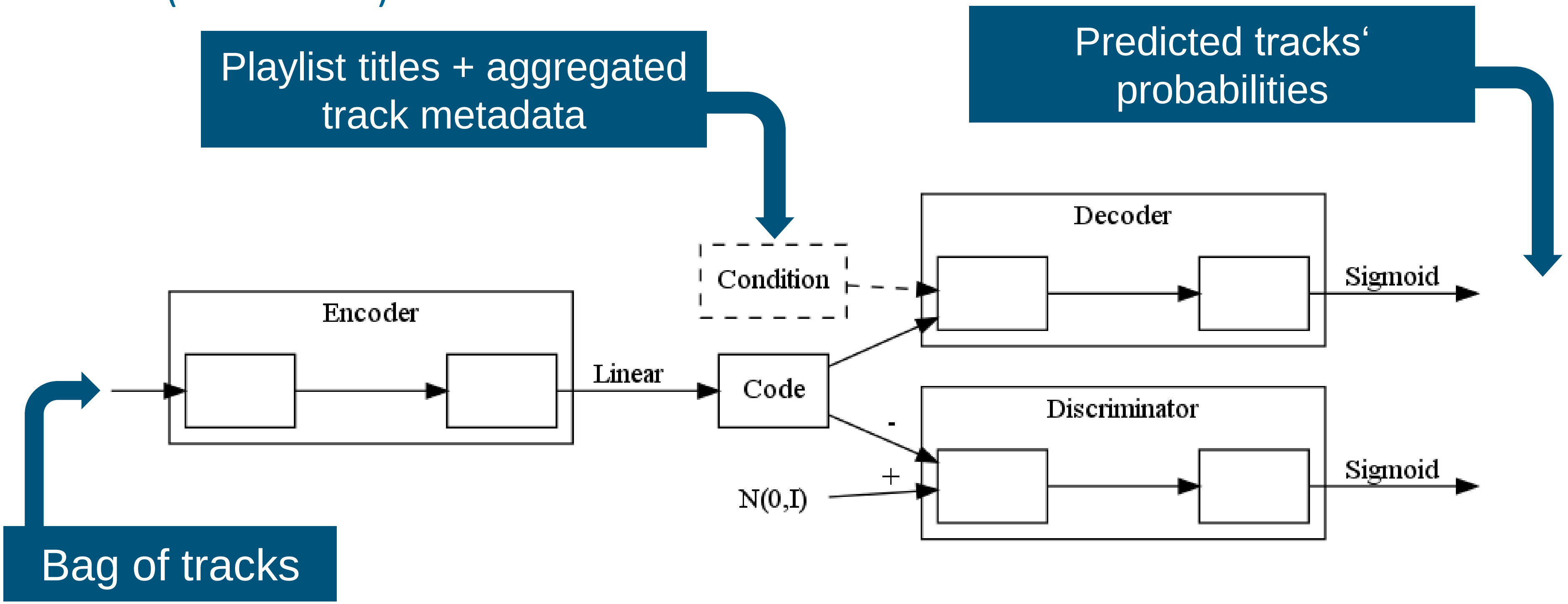
- Multi-Modal Adversarial Autoencoders (AAE)
  - Train autoencoder on track sets (playlists)
  - Supply condition to the decoder (multi-modal)
  - Match code with a normal distribution for smooth representations (adversarial)



Approach

www.moving-project.eu

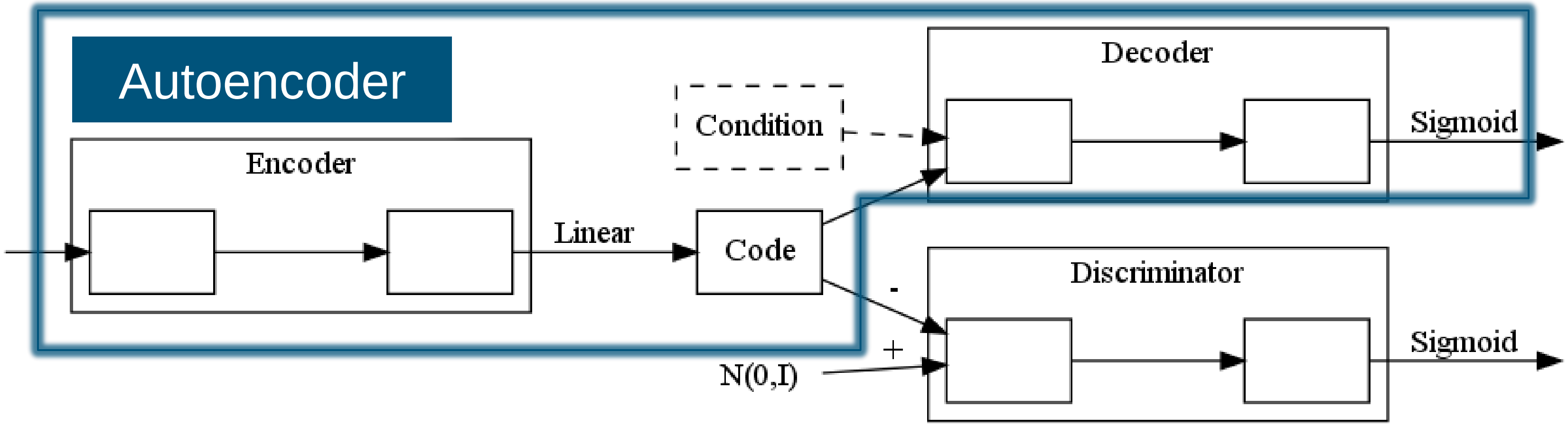
- Multi-Modal Adversarial Autoencoders (AAE)
  - Train autoencoder on track sets (playlists)
  - Supply condition to the decoder (multi-modal)
  - Match code with a normal distribution for smooth representations (adversarial)



Approach

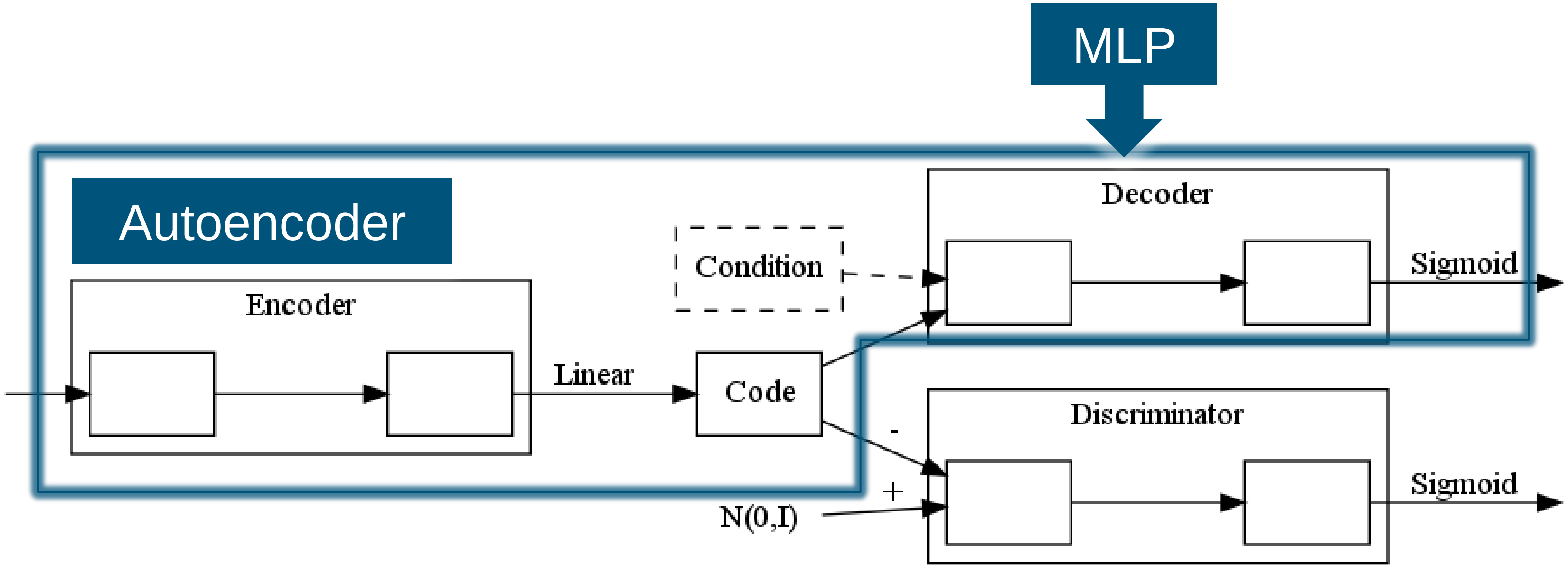
MOVING
www.moving-project.eu

- Multi-Modal Adversarial Autoencoders (AAE)
  - Train autoencoder on track sets (playlists)
  - Supply condition to the decoder (multi-modal)
  - Match code with a normal distribution for smooth representations (adversarial)



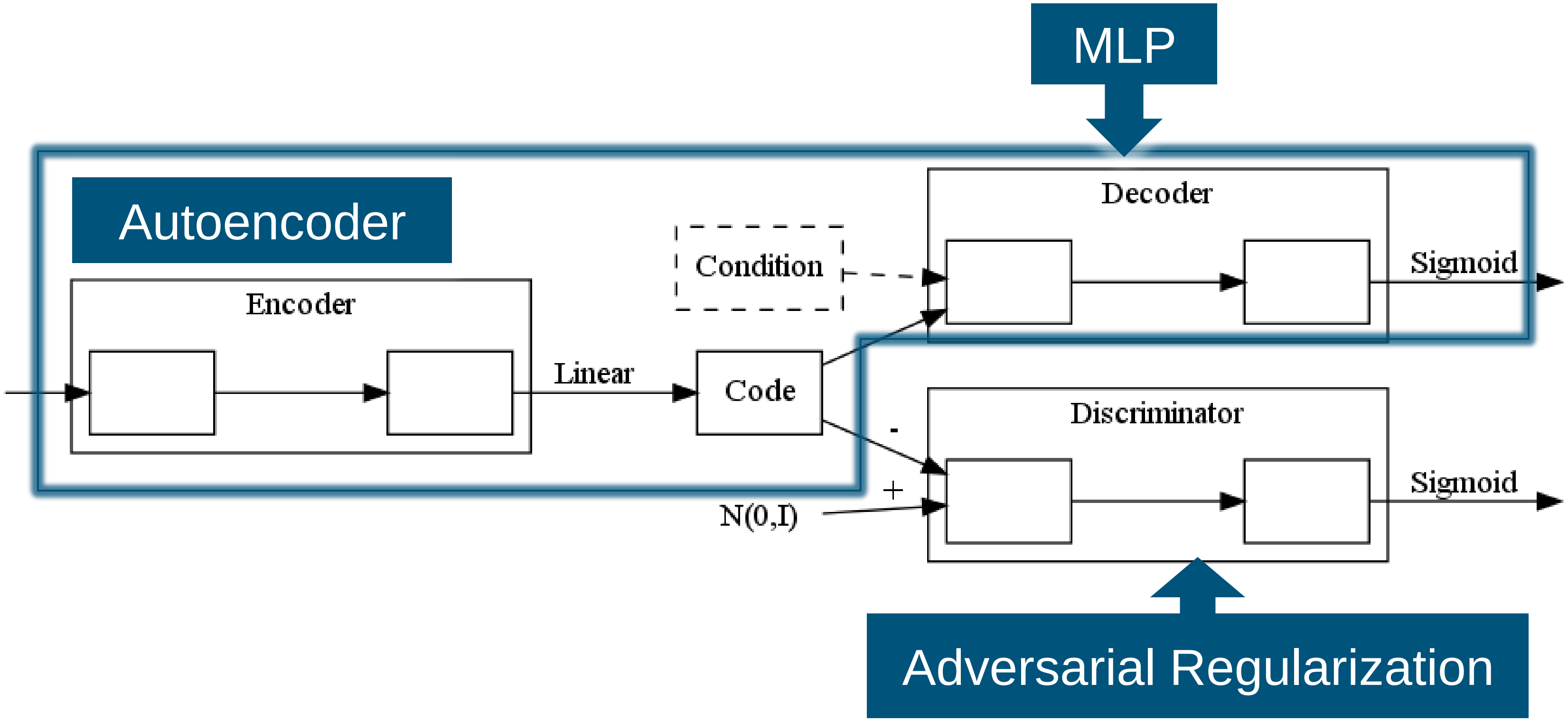
Approach

- Multi-Modal Adversarial Autoencoders (AAE)
  - Train autoencoder on track sets (playlists)
  - Supply condition to the decoder (multi-modal)
  - Match code with a normal distribution for smooth representations (adversarial)



Approach

- Multi-Modal Adversarial Autoencoders (AAE)
  - Train autoencoder on track sets (playlists)
  - Supply condition to the decoder (multi-modal)
  - Match code with a normal distribution for smooth representations (adversarial)



Example

Bag of tracks for "Walk of Life"

Rock	0
Workout	1
Chill	0

Example

Bag of tracks for “Walk of Life“

Rock	0
Workout	1
Chill	0

+

Bag of words for “Workout“

Workout	1
	0
Walk	1
of	1
life	1
	…

Example

Bag of tracks for “Walk of Life“

Rock	0
Workout	1
Chill	0

\+

p(“We are the champions” |“Workout”)

Bag of words for “Workout“

Workout	1
	0
Walk	1
of	1
life	1
	…

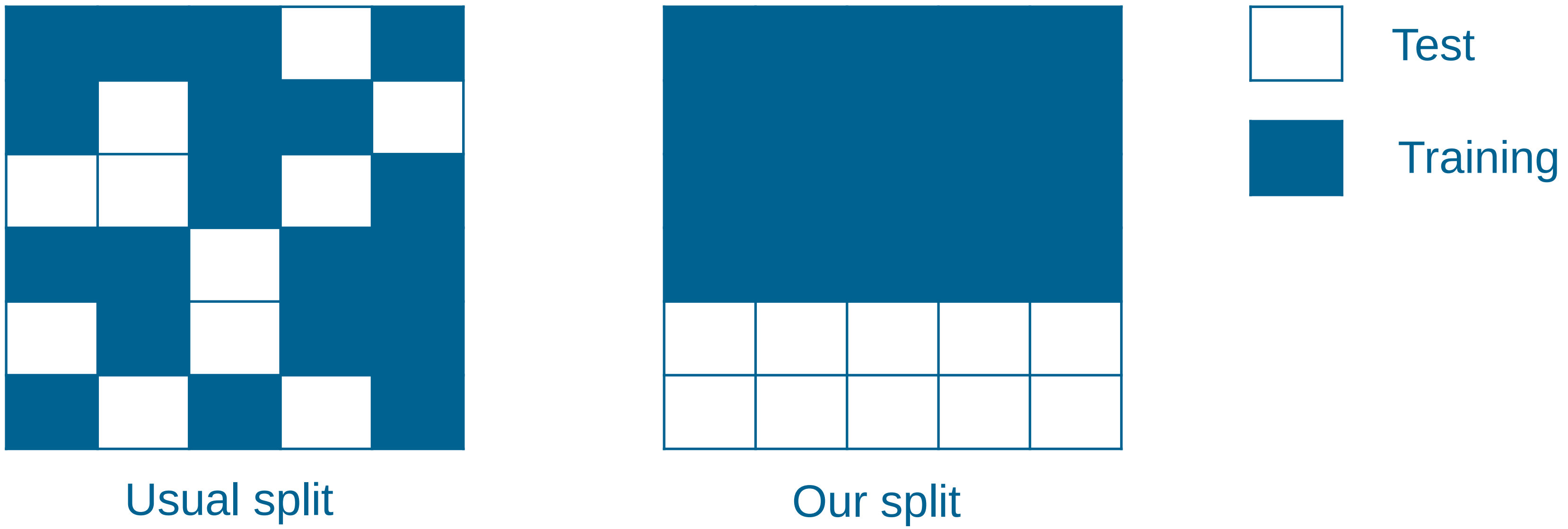
Experimental Procedure

- Preliminary experiments
- AAE optimization on a development set
- Final experiments on the challenge set

Preliminary experiments

MOVING
www.moving-project.eu

- Goals
  - verify that the approach is effective
  - check if using additional metadata is beneficial
- Comparison with 4 state-of-the-art methods
- Run every methods with and without playlist titles
- New user settings



Results of Preliminary Experiments

Method	MRR	
	No Titles	Titles
IC	0.0515 (0.1700)	-
SVD	0.0658 (0.1946)	0.0662 (0.1953)
AE	0.0645 (0.1855)	0.0679 (0.1913)
AAE	**0.0682 (0.1937)**	**0.0700 (0.1958)**
MLP	-	0.0300 (0.1310)

- Adversarial regularization consistently improves the performance of autoencoders for automatic playlist continuation
- Using playlist titles is beneficial

AAE optimization on the development set

MOVING
www.moving-project.eu

- Test 20 configurations
  1. Different values of hidden units, epochs and code size with n_{tracks} = 50,000
  2. Choice of best-performing values while varying n_{tracks}
- Run every configuration with playlist titles only and with playlist titles + track metadata
- Best performing configuration
  - n_{tracks} = 75,000, 200 hidden units, 20 epochs, code size = 100

Hyperparameters	Values
n_{tracks}	25 k, 50 k, 75 k, 100 k
Hidden units	50, 100, 200
Epochs	10, 20
Code size	50, 100

Final experiments on the challenge set

MOVING
www.moving-project.eu

- Test of several configurations varying n_{tracks}
- Setting other parameters to best-performing on the dev set
  - 200 hidden units
  - 20 epochs
  - code size = 100
- Only considering aggregated metadata

Results on the Dev and Challenge Set

MOVING
www.moving-project.eu

Set	R-Prec		NDCG		Clicks	
	Titles	Aggr.	Titles	Aggr.	Titles	Aggr.
Dev	0.1063	0.1205	0.2092	0.2319	9.9477	7.9350
Challenge	**-**	**0.1787**	**-**	**0.3201**	**-**	**5.3510**

- Using aggregated metadata is beneficial

Conclusions

- Adversarial Autoencoders are effective for automatic playlist continuation
- Aggregating items attributes is beneficial

Conclusions

- Adversarial Autoencoders are effective for automatic playlist continuation
- Aggregating items attributes is beneficial

Code at https://github.com/lgalke/mpd-aae-recommender

i.vagliano@zbw.eu @maponaso

MOVING is funded by the EU Horizon 2020 Programme under the project number INSO-4-2015: 693092

Parameters on preliminary experiments

- 100 hidden layers
- ReLU activation function
- Drop probabilities after each layer 0.2
- Code size 50
- Adam for optimization
- Initial learning rate 0.001
- Gaussian prior distribution

Development set characteristics

- 10,000 random playlists
- 2,000 without title and either five tracks or ten tracks retained at random (0.5 probability)
- 8,000 with a randomly selected number of retained tracks
  - either 100 or 25 tracks with a 0.2 probability each
  - either zero, one, five, or ten tracks, with probability 0.1 each
- Random sampling approach
  - resulting distribution of tracks slightly different from the challenge set
- Selection of tracks always random
  - no distinction between selecting the first tracks or random tracks
- Naive approach for playlists with few tracks
  - Cannot remove more tracks than available
- Negligible effects

Hyperparameters on Dev set

- Vocabulary with the 50,000 most frequent distinct words from the metadata
  - playlist title, track title, artist name, and album title
- Different values of hidden units, epochs and code size on a predefined vocabulary based on Google
  - n_{tracks} = 50,000.
- Best-performing values (200, 20 and 100) chosen while varying n_{tracks}

Hyperparameters	Values
n_{tracks}	25 k, 50 k, 75 k, 100 k
Hidden units	50, 100, 200
Epochs	10, 20
Code size	50, 100