Discrete Representation of Behaviors

Semester Project

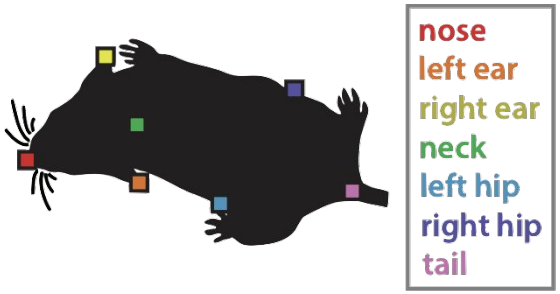
Célia Benquet

Mathis Laboratory of adaptive motor control

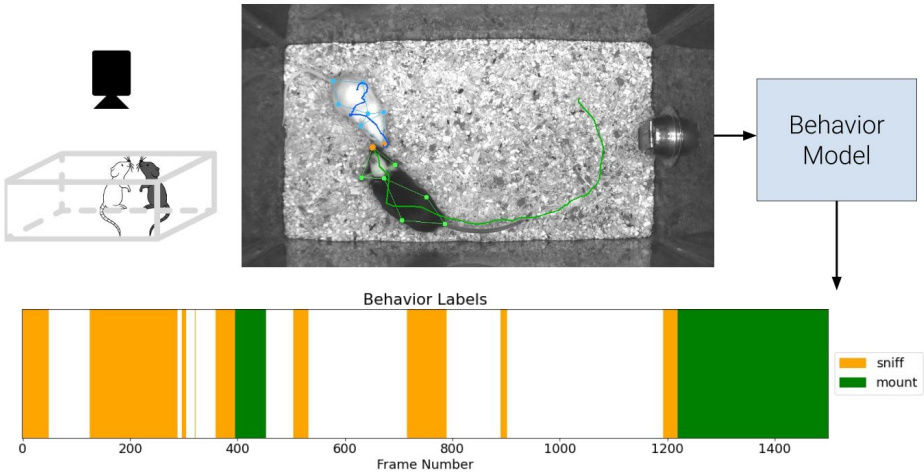
Célia Benquet

Caltech Mouse Social Interactions (CalMS21) Dataset

- Multi-agent behavior dataset, Sun et al., April 2021
- Resident-Intruder format



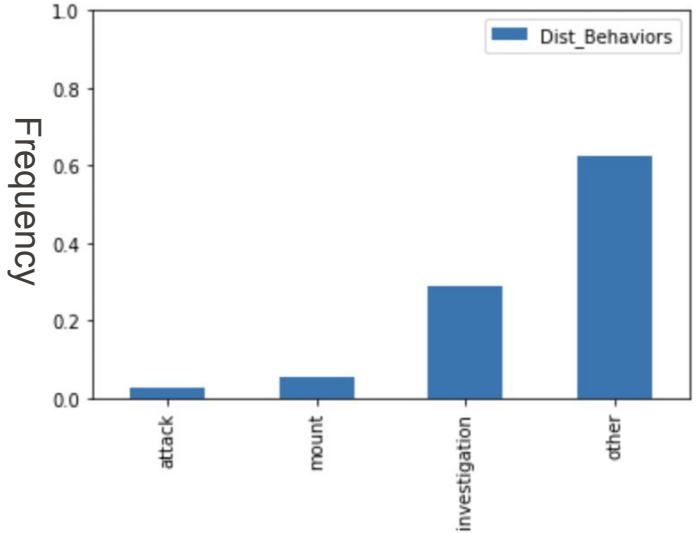
- Keypoints locations using Mouse Action Recognition System (MARS, Segalin et al. 2020)
- 4 labelled behavior classes:
  - Attack
  - Mount
  - Investigation
  - Other



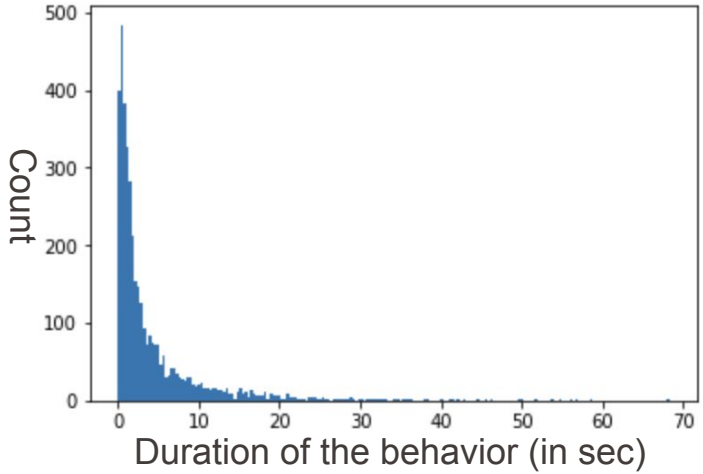
Célia Benquet

Caltech Mouse Social Interactions (CalMS21) Dataset

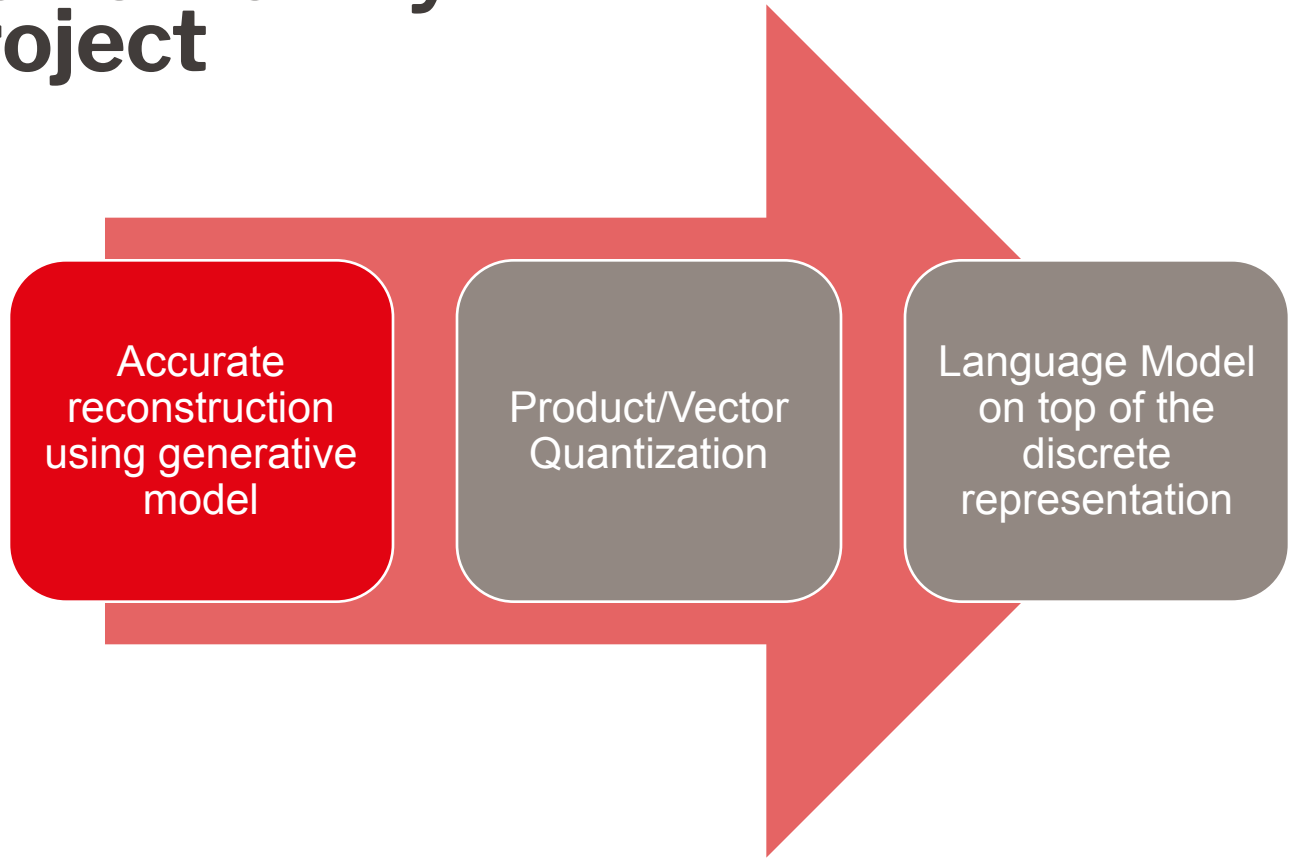
- Imbalanced classes
  → low proportion of ‘attack’



- Mean duration of behavior ~4sec
- Videos between 1min/10min



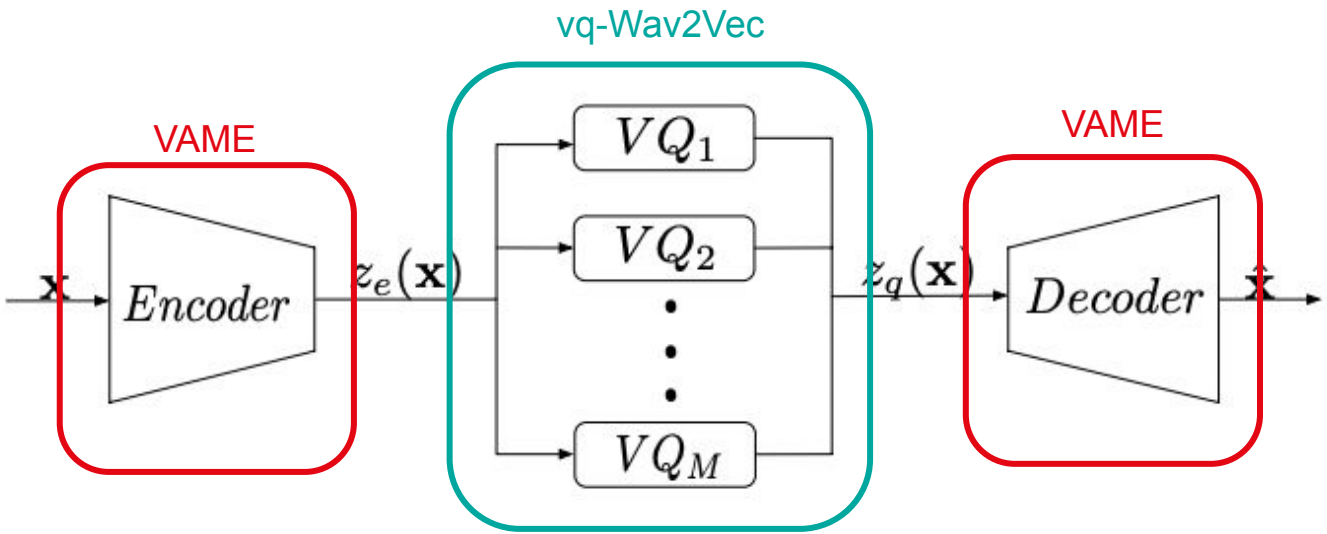
Review of my project



Célia Benquet

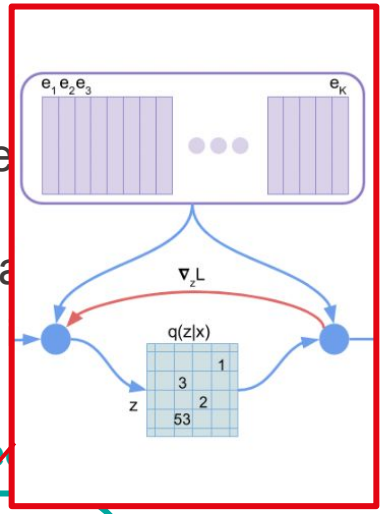
My model

- Encoder/Decoder from VAME (Luxem et al. 2018)
  - RNNs
- K-means Product Quantiser from vq-Wav2Vec (Schneider et al. 2020)
  - Online differential version of K-means in the latent space

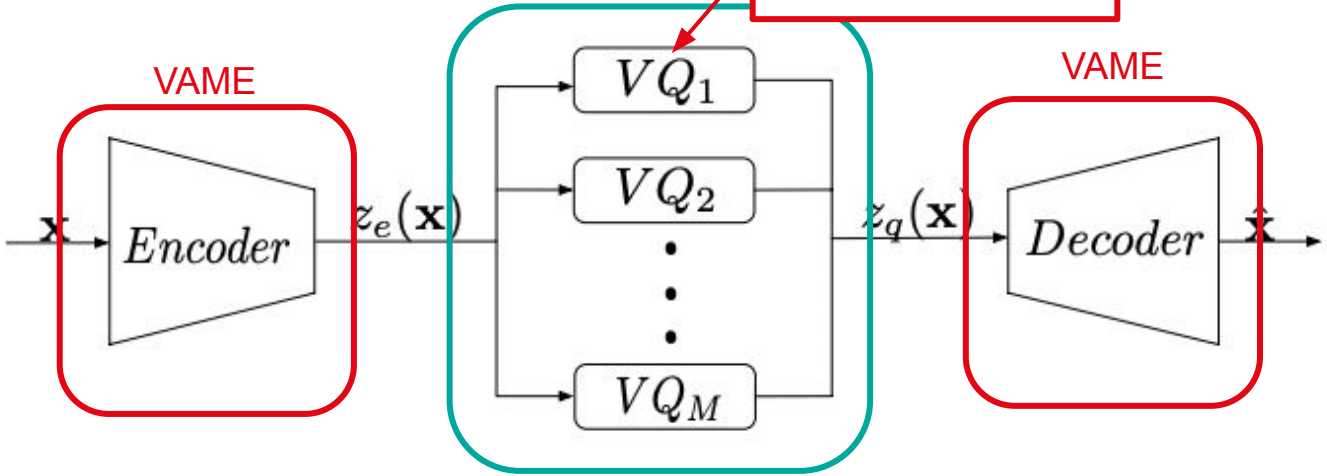


My model

- Encoder/Decoder from VAME (Luxem e
  - RNNs
- K-means Product Quantiser from vq-Wa et al. 2020)
  - Online differential version of K-means



vq-Wav2Ve



Metrics

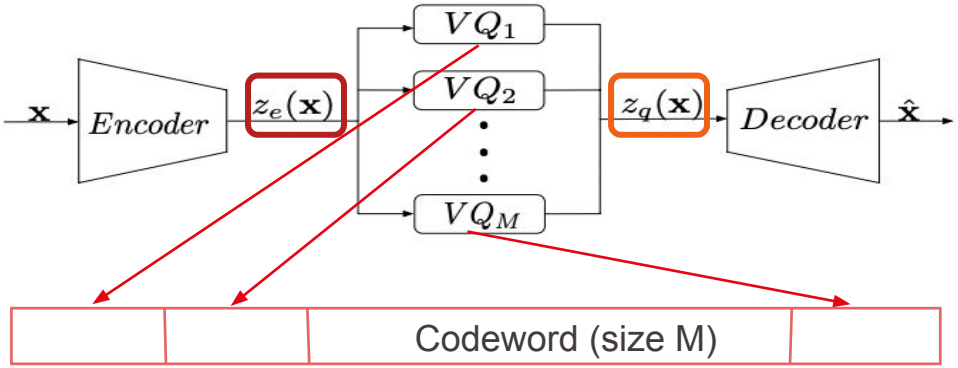
- R2
- RMSE

Evaluate reconstruction

Supervised learning for behavior estimation

Only at validation

- Classification accuracy from data before latent space (SVM) ⇒ ze(x)
- Classification accuracy from data after latent space (SVM) ⇒ zq(x)
- Classification accuracy from discrete representation of the data (DecisionTreeClassifier) ⇒ Codewords



Célia Benquet

Metrics

Accuracy = Mean of the diagonal of the confusion matrix

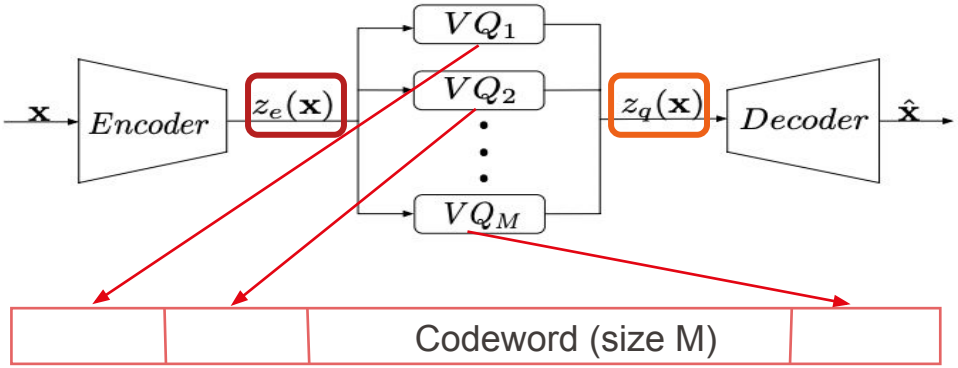
- R2
- RMSE

Evaluate reconstruction

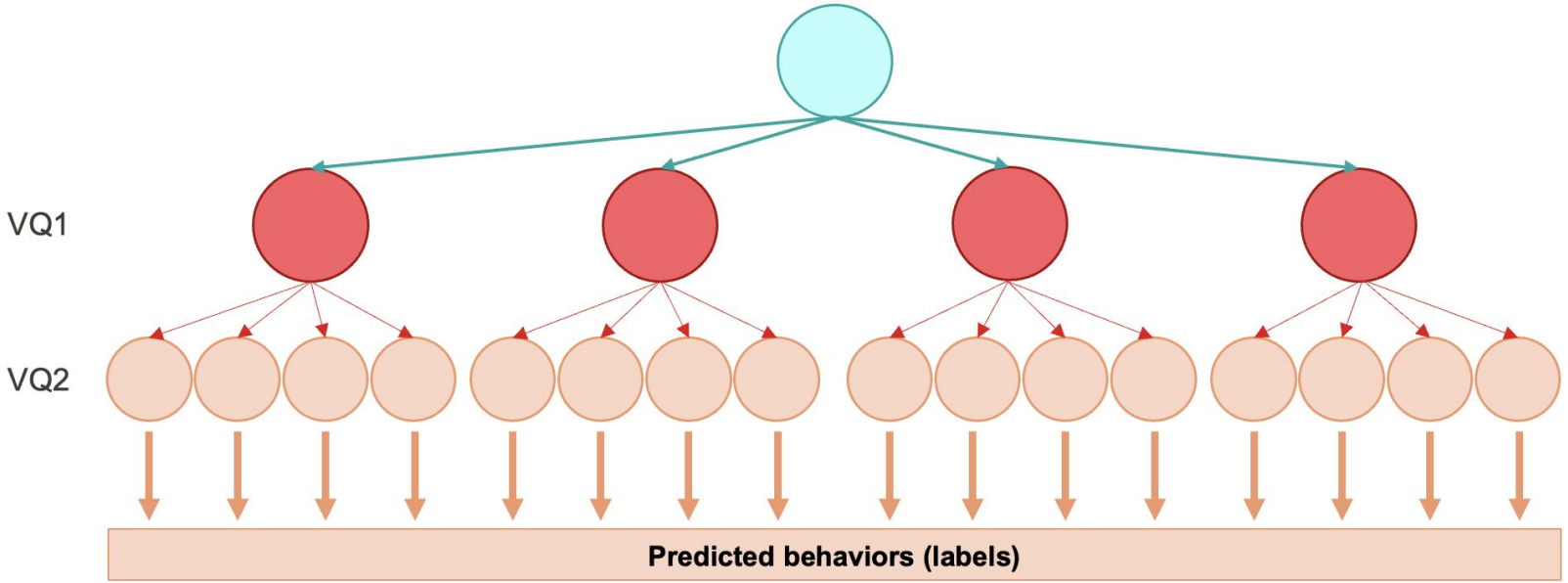
Supervised learning for behavior estimation

Only at validation

- Classification accuracy from data before latent space (SVM) ⇒ ze(x)
- Classification accuracy from data after latent space (SVM) ⇒ zq(x)
- Classification accuracy from discrete representation of the data (DecisionTreeClassifier) ⇒ Codewords



Rationale: Discrete Tree Classification



- Toy example for 2 groups / 4 variables

Célia Benquet

Losses

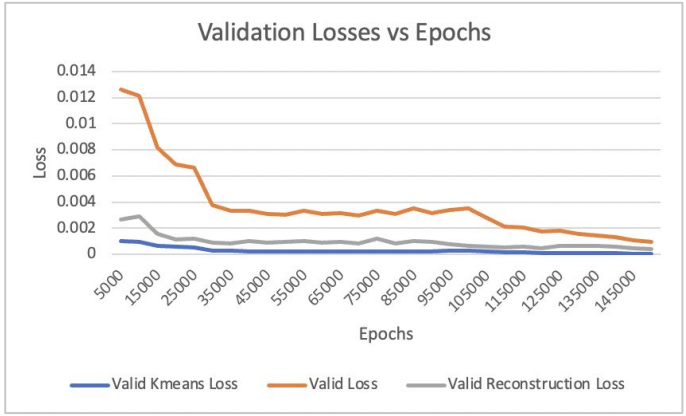
*Loss = Reconstruction Loss + Eta * kMeans Loss*

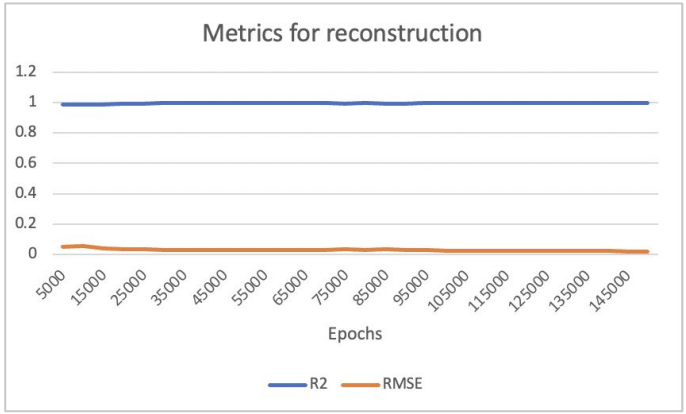
- ***Reconstruction loss:*** *nn.MSELoss() ⇒ optimize encoder/decoder*
- ***kMeans loss:***
  - *kMean loss = Latent loss + gamma * commitment loss*
  - *Latent loss ⇒ st embedding vectors move towards encoder outputs*
  - *Commitment loss ⇒ st encoder commits to an embedding / its output does not grow*

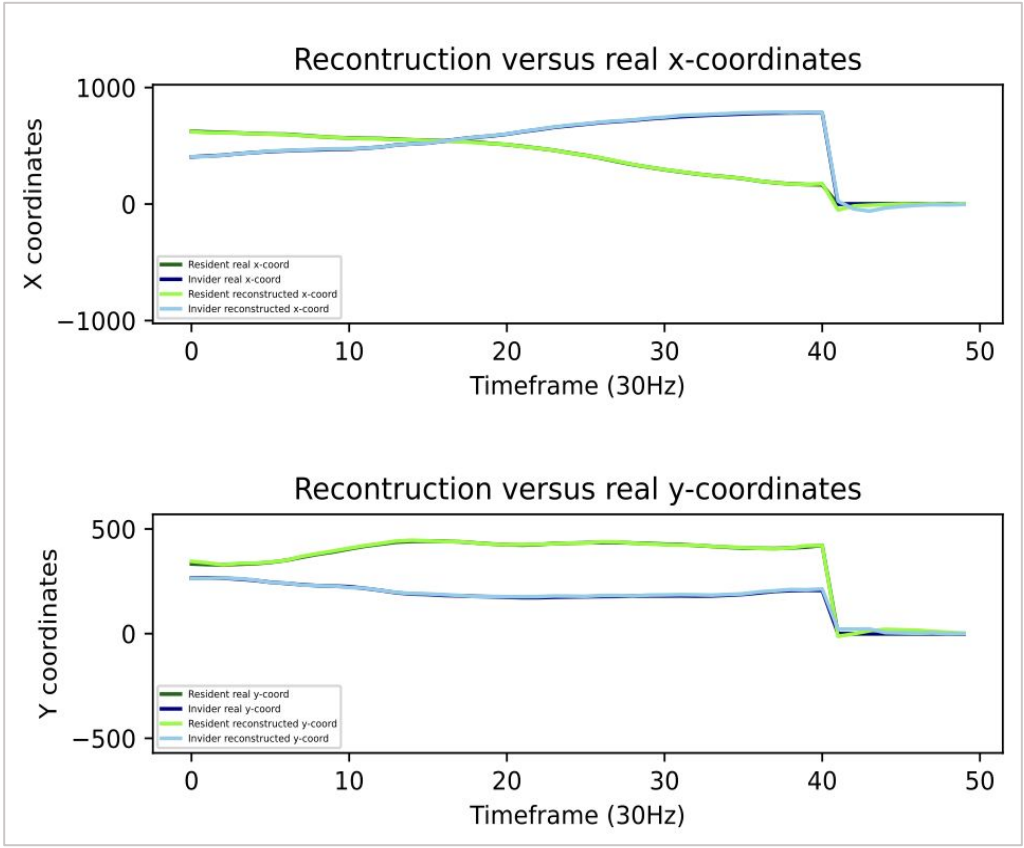
Egocentric Representation

Pipeline: *Data ⇒ Normalization ⇒ Alignment (centered, aligned on y-axis) ⇒ Model ⇒ Reconstruction ⇒ Realignment ⇒ De-normalization*

Egocentric Representation (32G/120V)



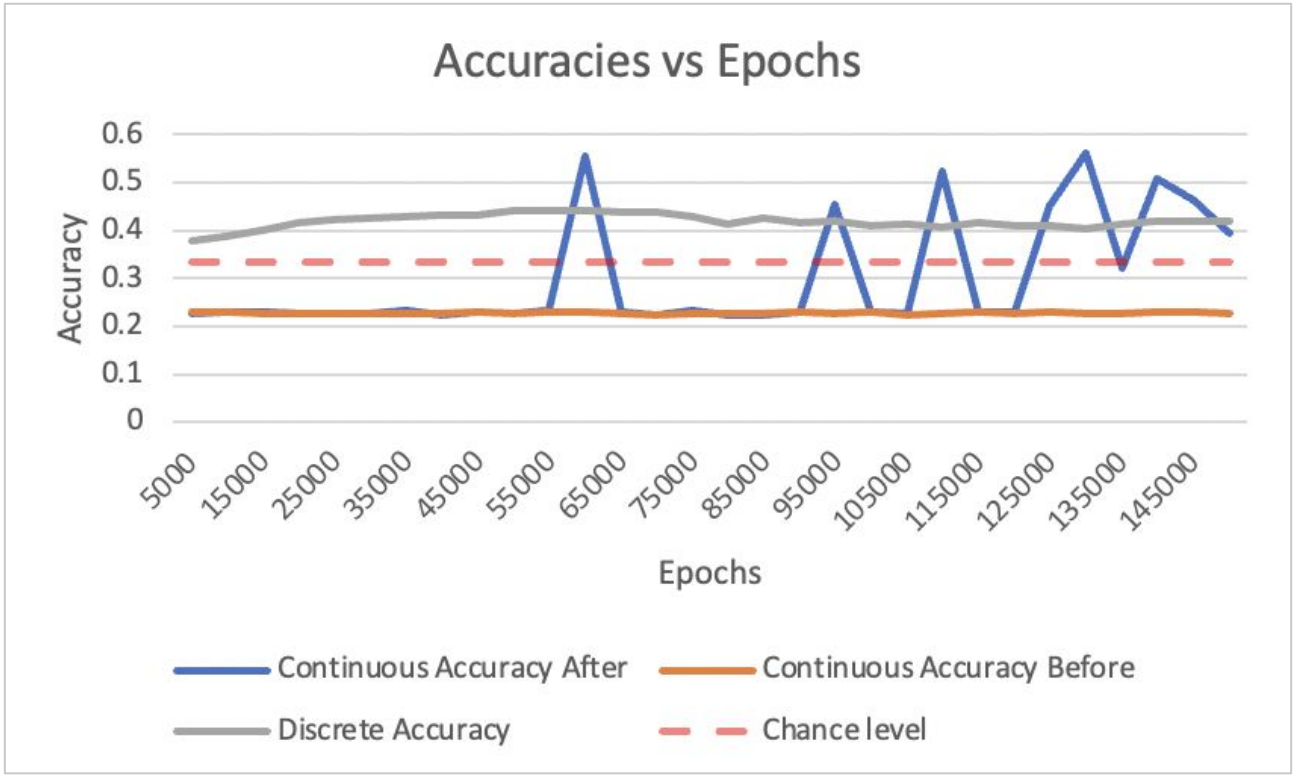




Célia Benquet

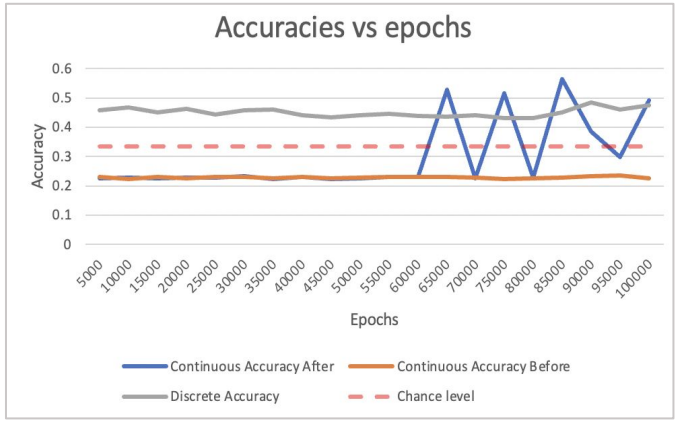
Célia Benquet

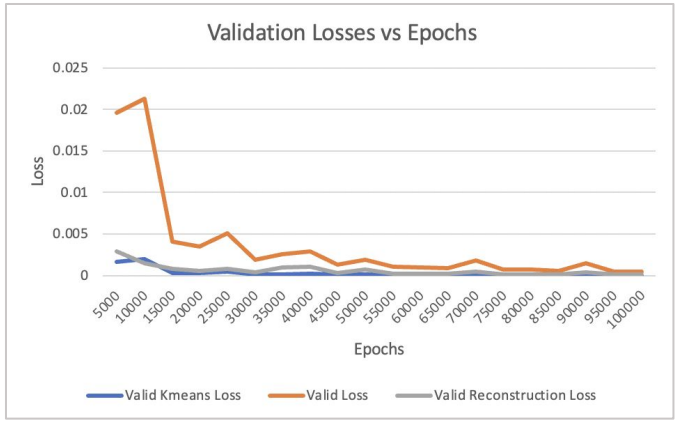
Egocentric Representation



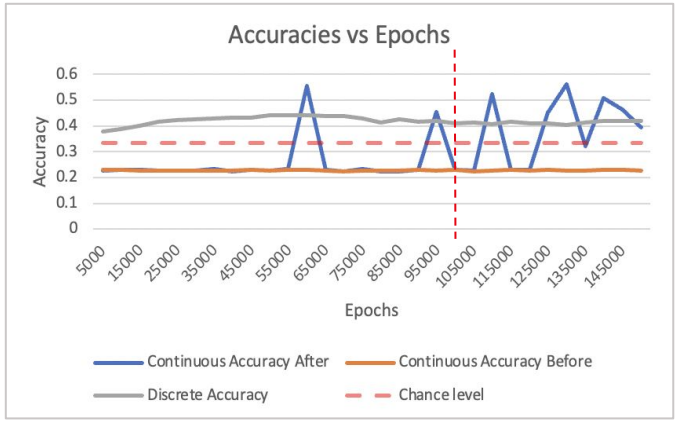
Comparison

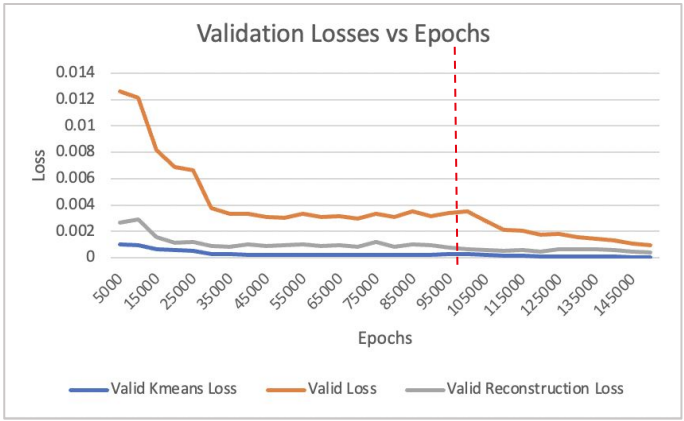
Baseline





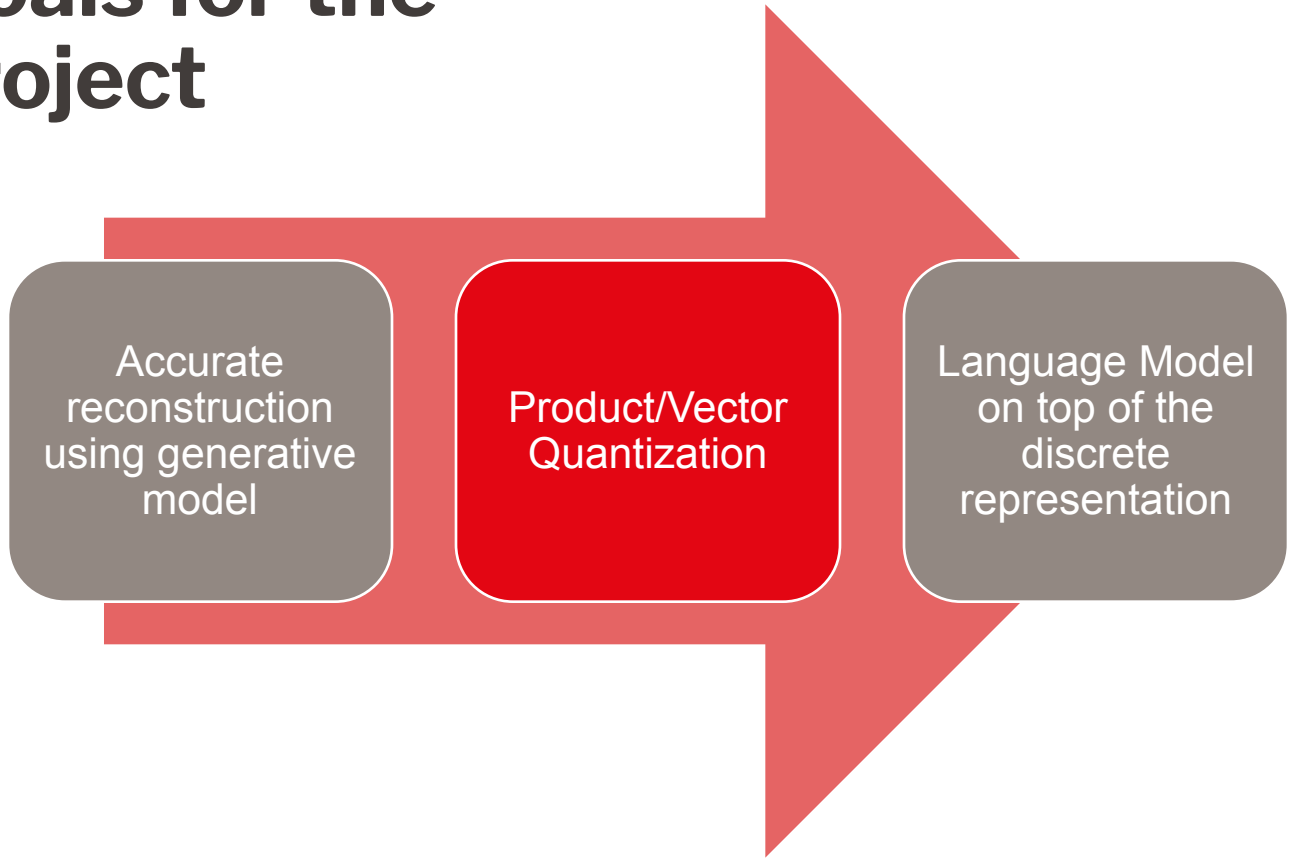
Egocentric





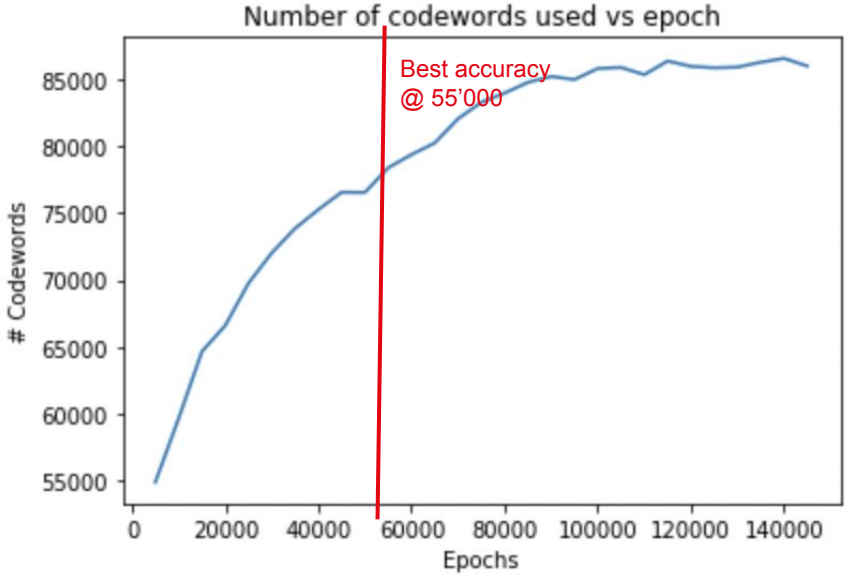
Célia Benquet

Goals for the project



Célia Benquet

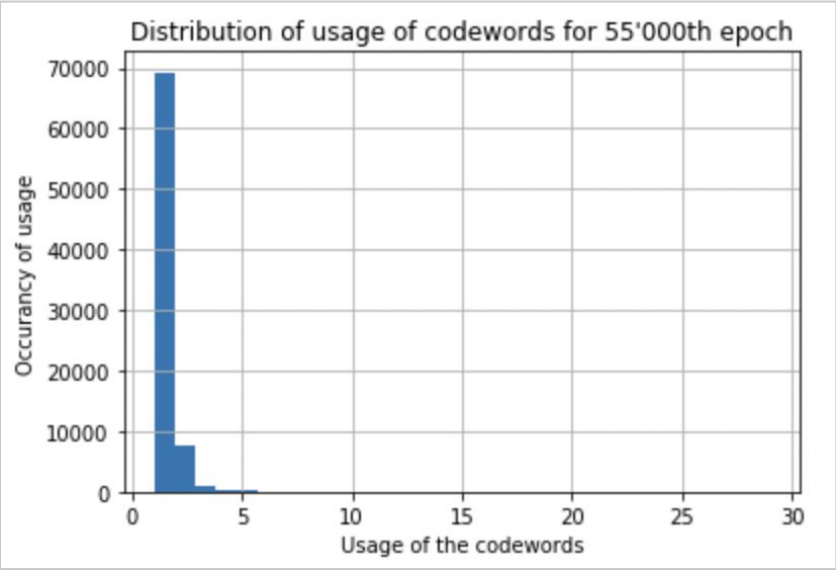
#Used Codewords increases with training time

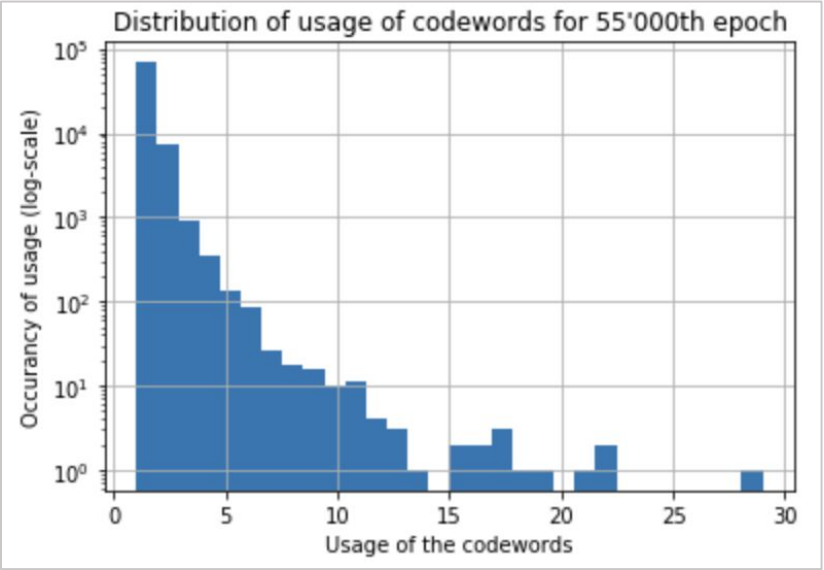


- 32 groups / 120 variables
- #Epochs ⇒ increase #codewords used especially #codewords used once
- Accuracy not linked to #codewords

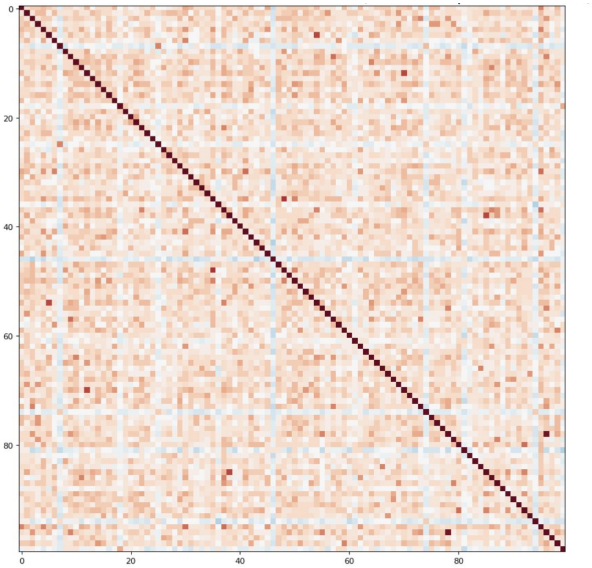
Célia Benquet

Most of the codewords used once

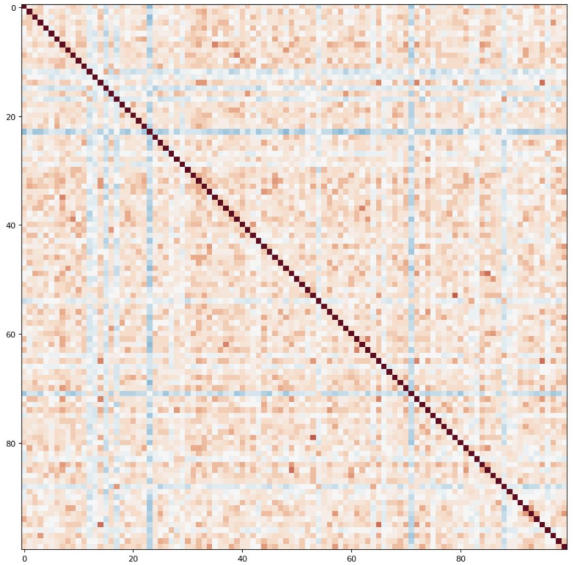




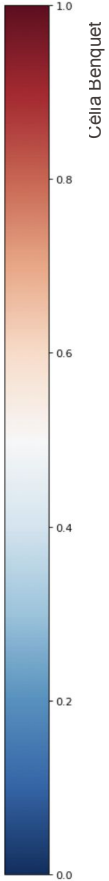
High similarity of variables used in the codewords



Codewords used more than once (100 out of 9111)



Codewords used once only (100 out of 69291)

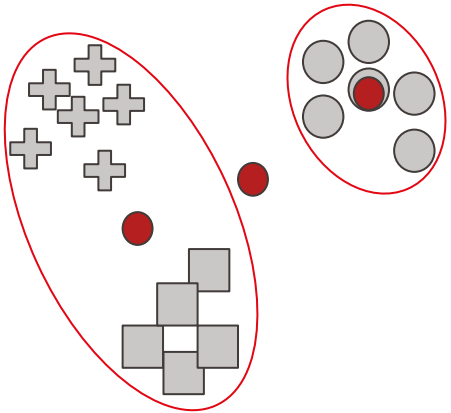


- Avg: ~0.55 (unique usage) and ~0.56 (multiple usages)
- Between unique usage codewords, used for the same target: avg ~0.54

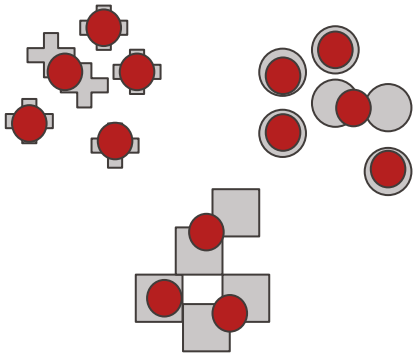
Hyp: Trade-off Tokens/Groups

- Investigation for a good trade-off → efficient utilisation of latent variables
- $\#Codewords = V^G$

Few groups/variables
⇒ mode collapse

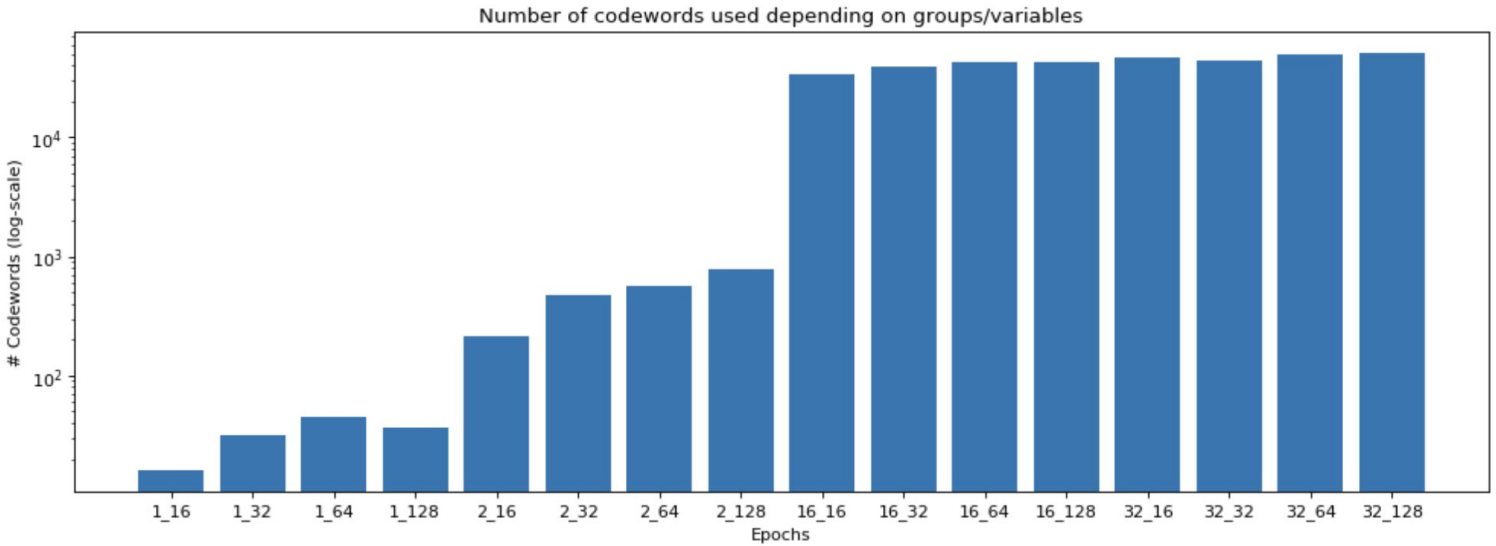


Lots of groups/lots of variables
⇒ poor discretisation



Célia Benquet

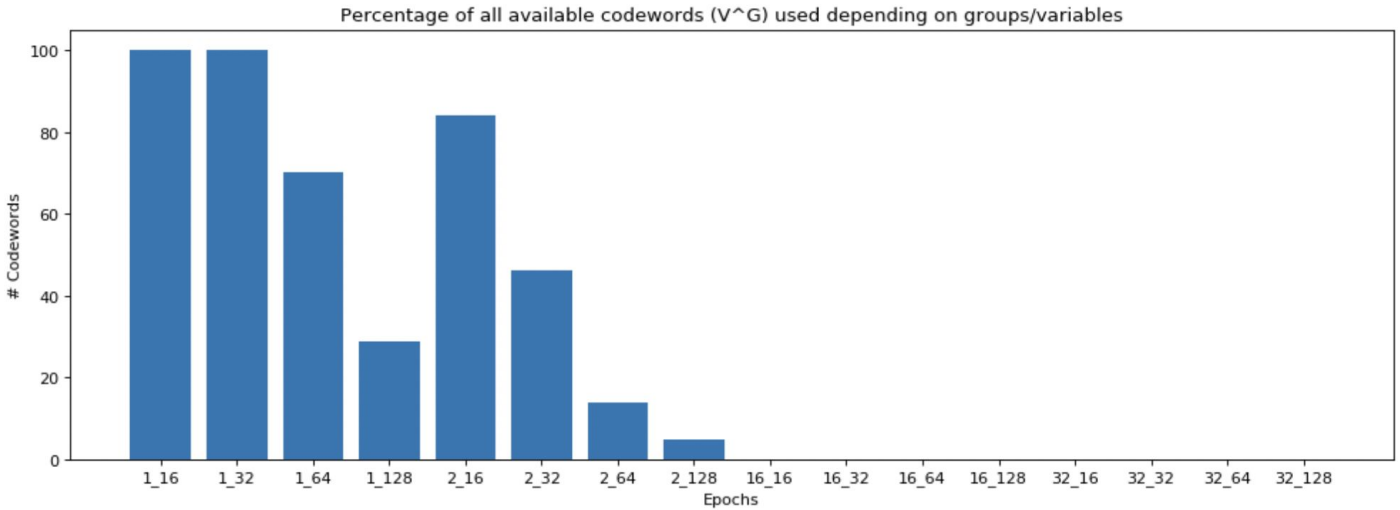
Higher #Codewords for more groups



	#Codewords
1_16	16
1_32	32
1_64	45
1_128	37
2_16	215
2_32	474
2_64	561
2_128	785
16_16	33654
16_32	39516
16_64	42326
16_128	42309
32_16	46744
32_32	44379
32_64	49165
32_128	51758

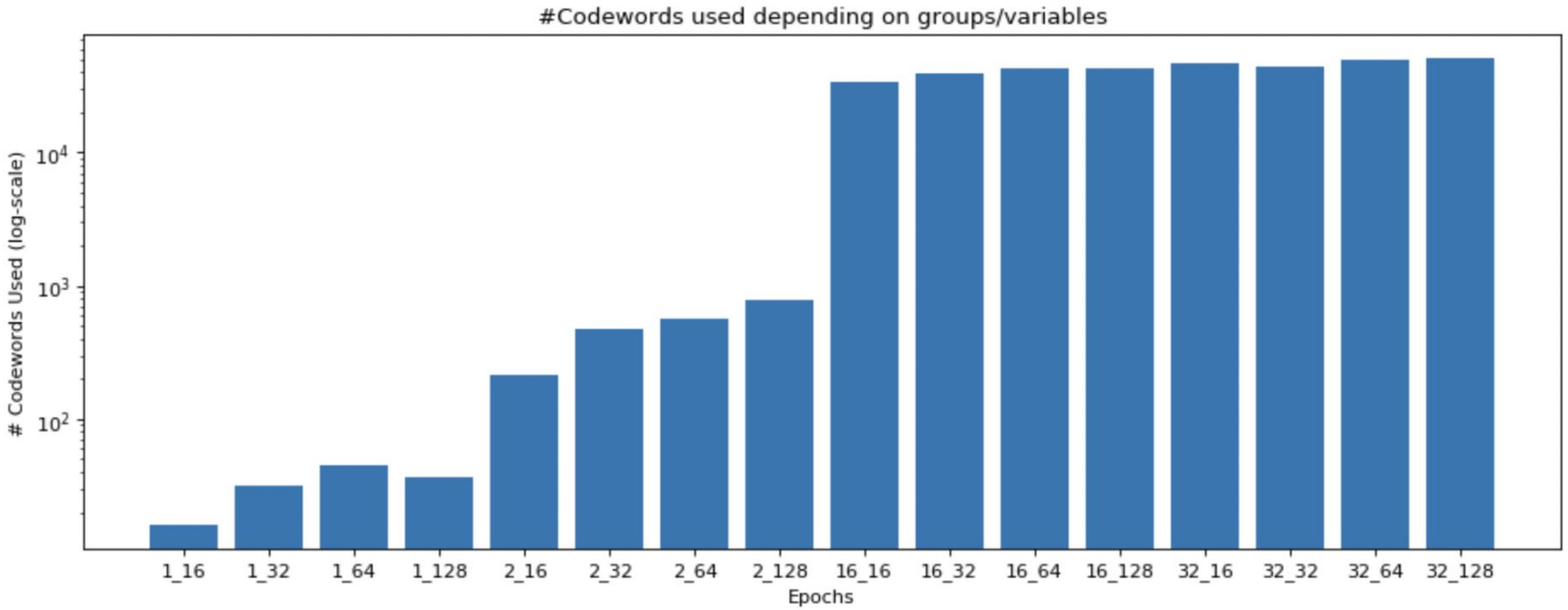
Higher usage of available codewords for less groups

- Total #Codewords = V^G



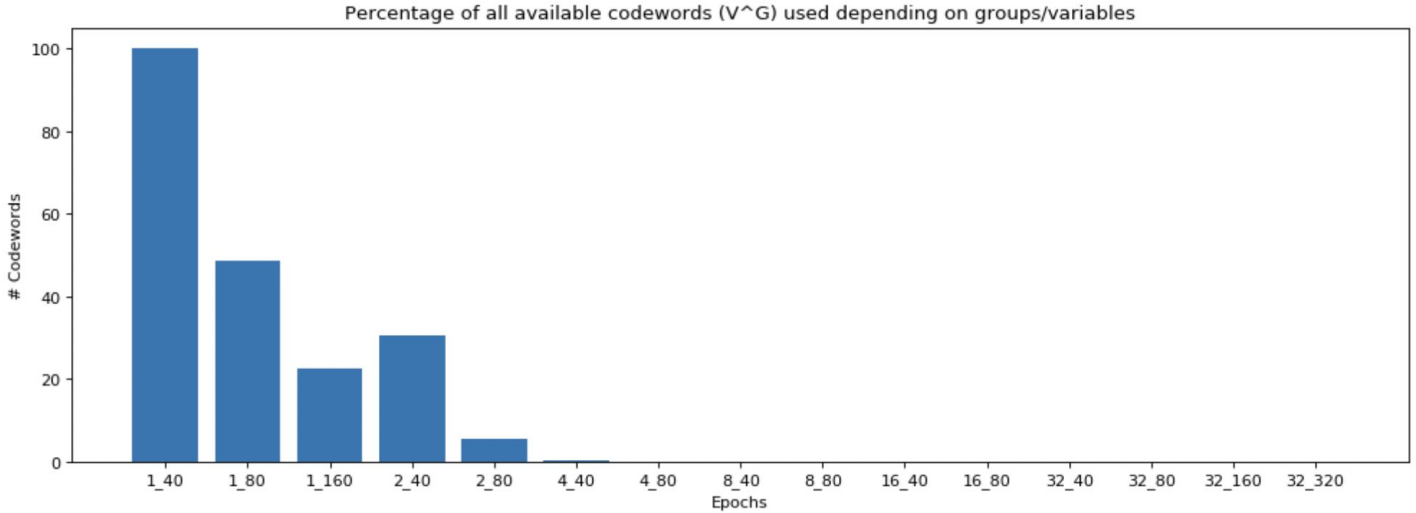
	Percentage(%)
1_16	100.00000
1_32	100.00000
1_64	70.31250
1_128	28.90625
2_16	83.98438
2_32	46.28906
2_64	13.69629
2_128	4.79126
16_16	0.00000
16_32	0.00000
16_64	0.00000
16_128	0.00000
32_16	0.00000
32_32	0.00000
32_64	0.00000
32_128	0.00000

Comparison to Schneider et al. (2020)



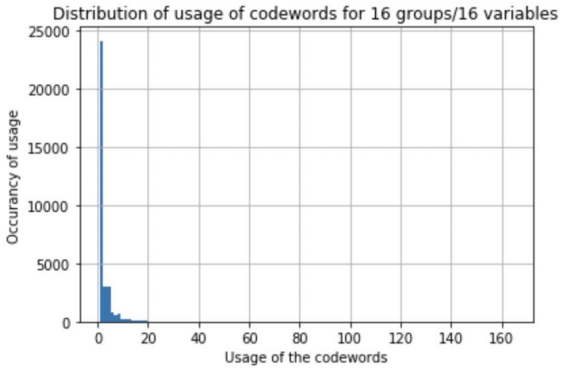
Comparison to Schneider et al. (2020)

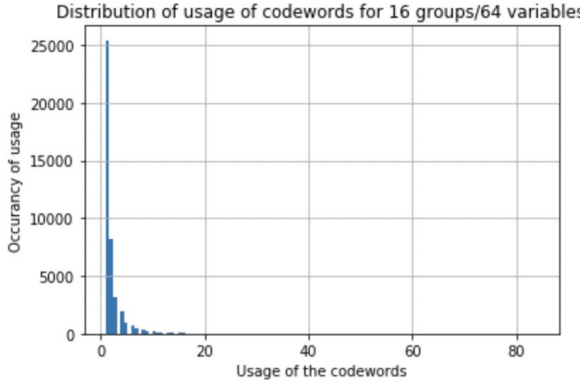
V	1 group	2 groups	4 groups	8 groups	16 groups	32 groups
40	100 % (40)	95.3 % (1.6k)	27.4 % (2.56M)	74.8 % (39.9M)	99.6 % (39.9M)	99.9 % (39.9M)
80	92.5 % (80)	78.5 % (6.4k)	11.8 % (39.9M)	91.5 % (39.9M)	99.3 % (39.9M)	100 % (39.9M)
160	95 % (160)	57.2 % (25.6k)	35.2 % (39.9M)	97.6 % (39.9M)	99.8 % (39.9M)	100 % (39.9M)
320	33.8 % (320)	24.6 % (102.4k)	57.3 % (39.9M)	98.7 % (39.9M)	99.9 % (39.9M)	100 % (39.9M)
640	24.6 % (640)	10 % (409.6k)	60.2 % (39.9M)	99.3 % (39.9M)	99.9 % (39.9M)	100 % (39.9M)
1280	7.2 % (1.28k)	4.9 % (1.63M)	67.9 % (39.9M)	99.5 % (39.9M)	99.9 % (39.9M)	100 % (39.9M)

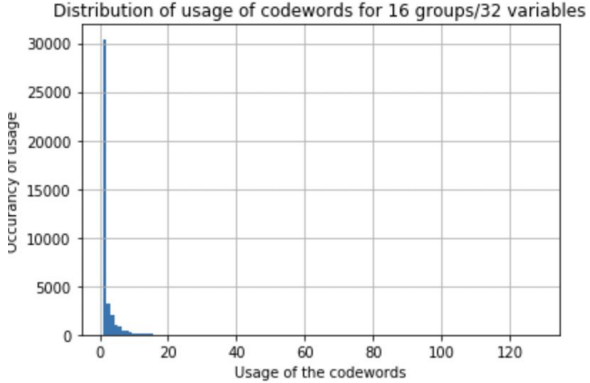


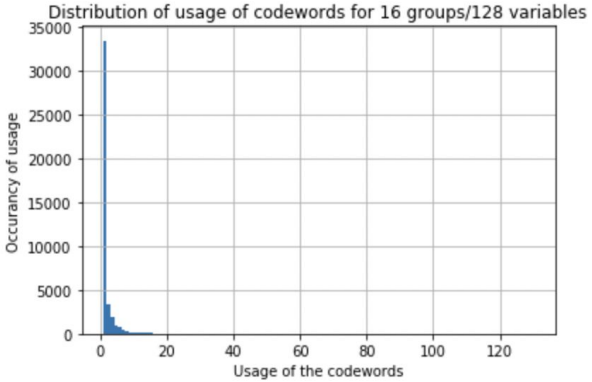
	Percentage(%)
1_40	100.00000
1_80	48.75000
1_160	22.50000
2_40	30.50000
2_80	5.64062
4_40	0.38742
4_80	0.00942
8_40	0.00000
8_80	0.00000
16_40	0.00000
16_80	0.00000
32_40	0.00000
32_80	0.00000
32_160	0.00000
32_320	0.00000

Variation of latent variables doesn't infer to much on skewness



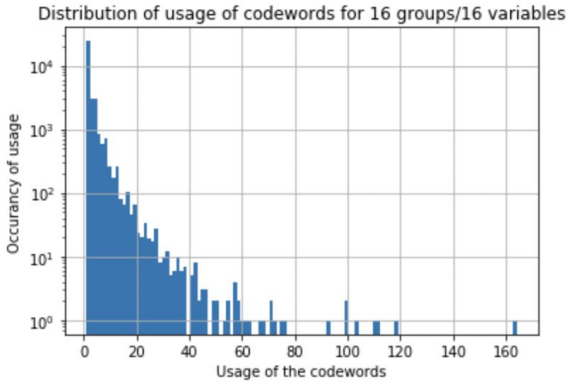


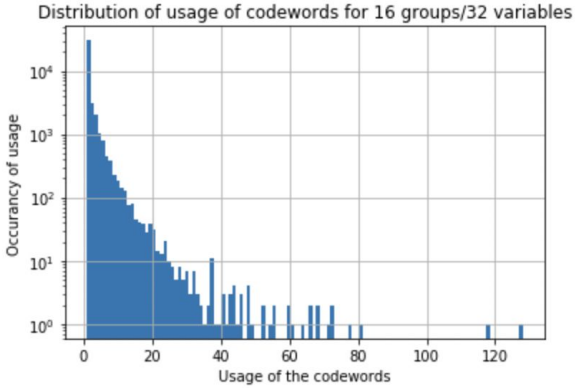


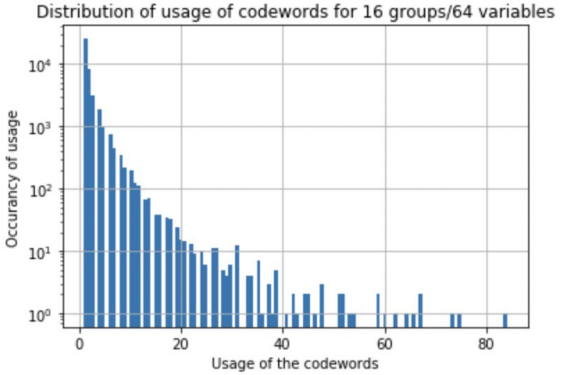


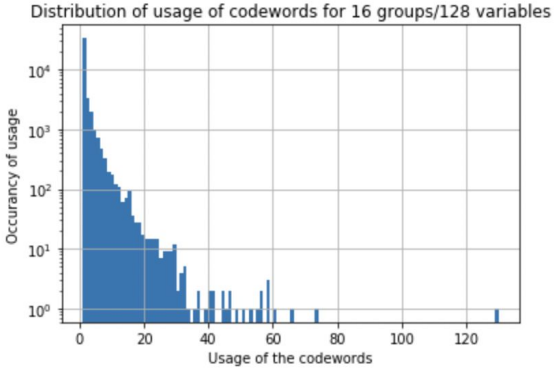
- Slightly more codewords (V^16)
- Skewness is variable

Variation of latent variables doesn't infer to much on skewness



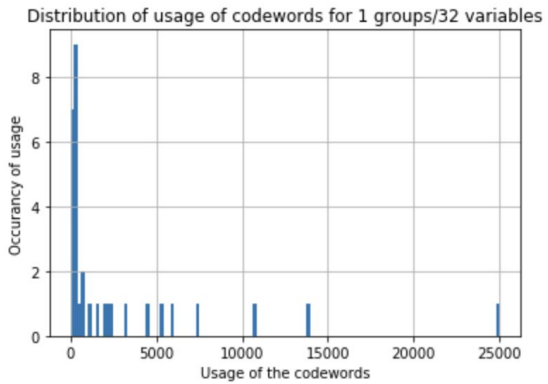


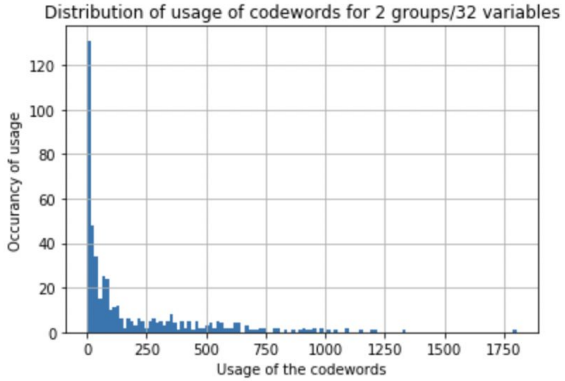


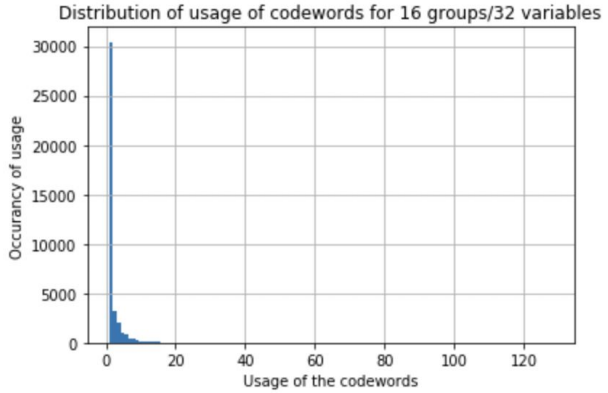


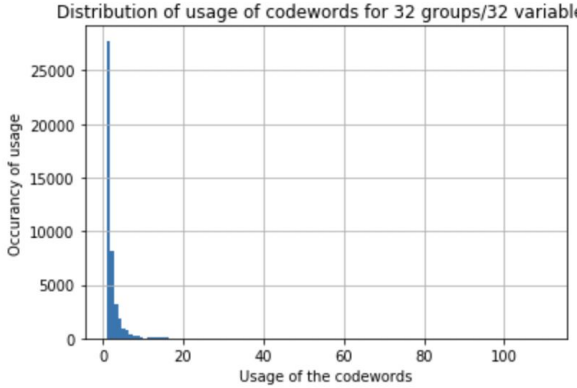
- Slightly more codewords (V^16)
- Skewness is variable

Variation of groups means more tokens used only once



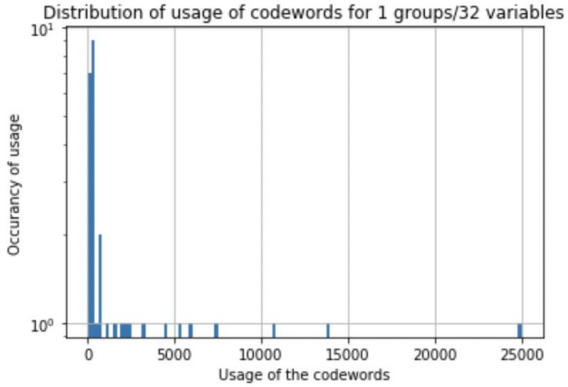


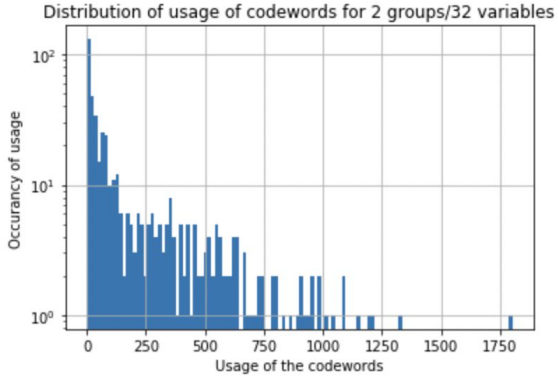


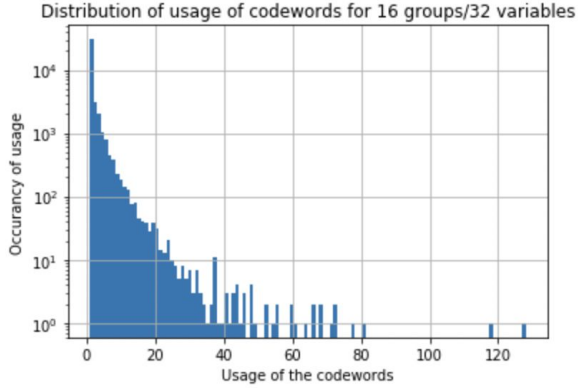


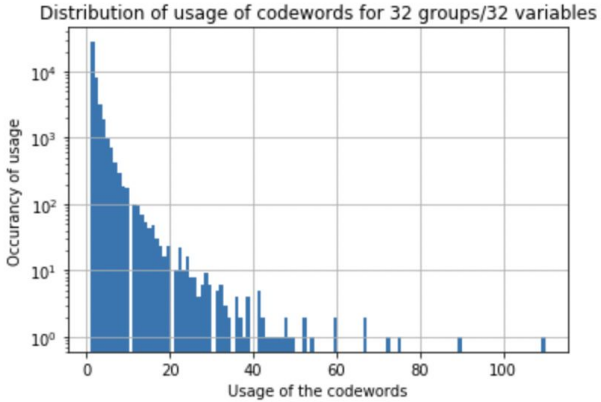
- More codewords of course (32^G)
- More tokens used once!

Variation of groups means more tokens used only once









- More codewords of course (32^G)
- More tokens used once!

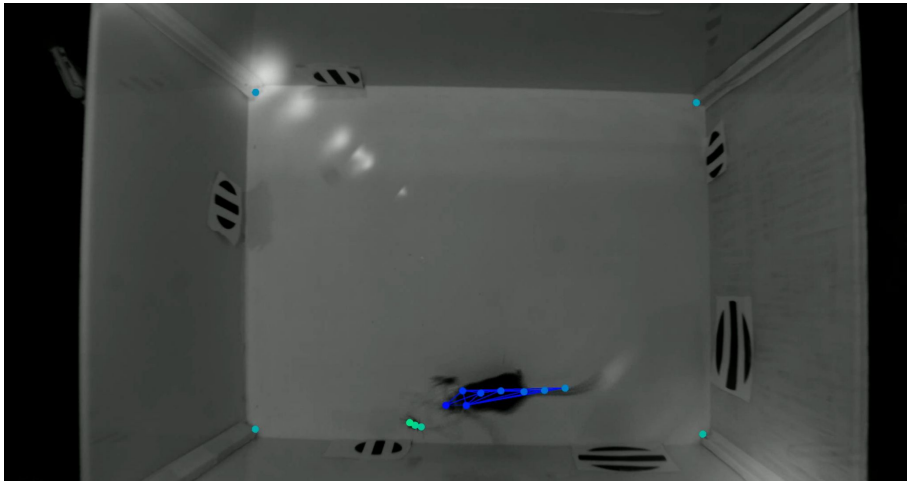
Possible directions of investigation

- Using a different/more diverse(!) dataset

  ⇒ only 3 classes which one is “Other”

  ⇒ Cricket hunting dataset

- ConvNet as Encoder/Decoder
  ⇒ comparison to VAME

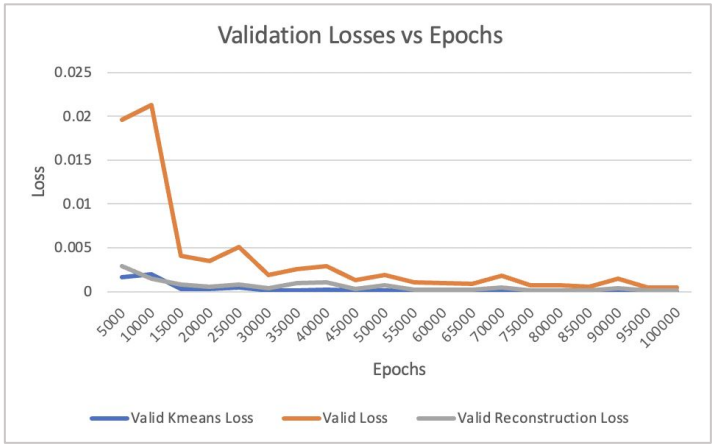


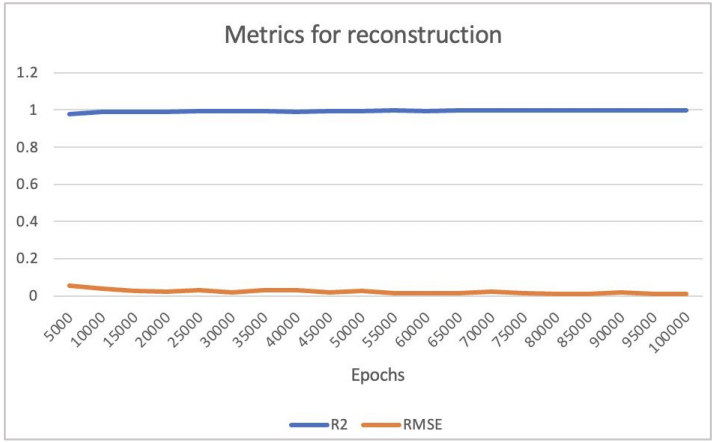
Thank you for your attention

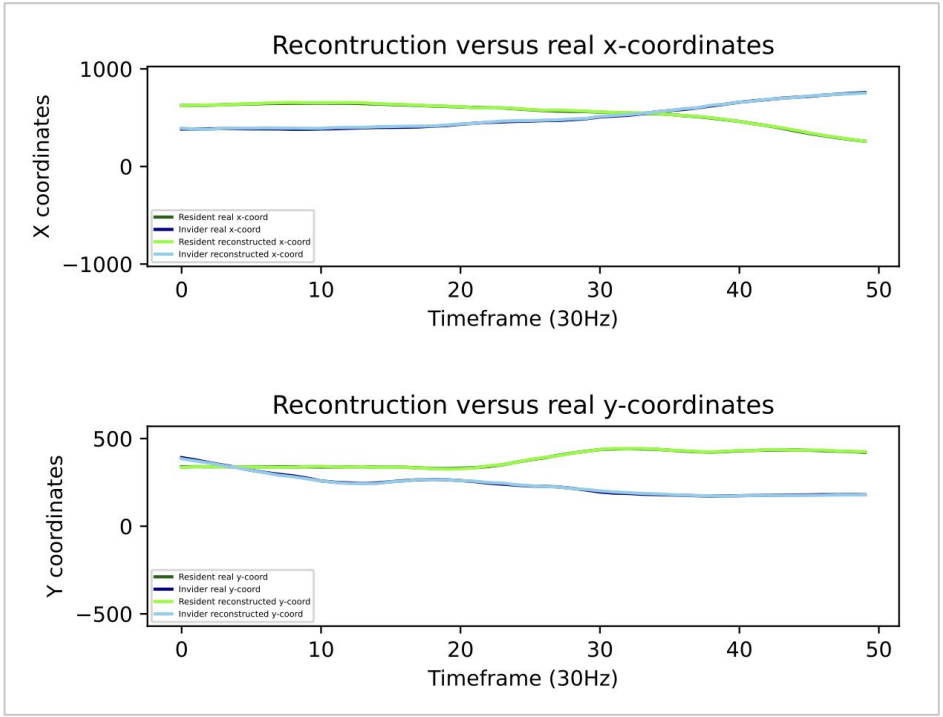
Any thoughts?

Extra Slides

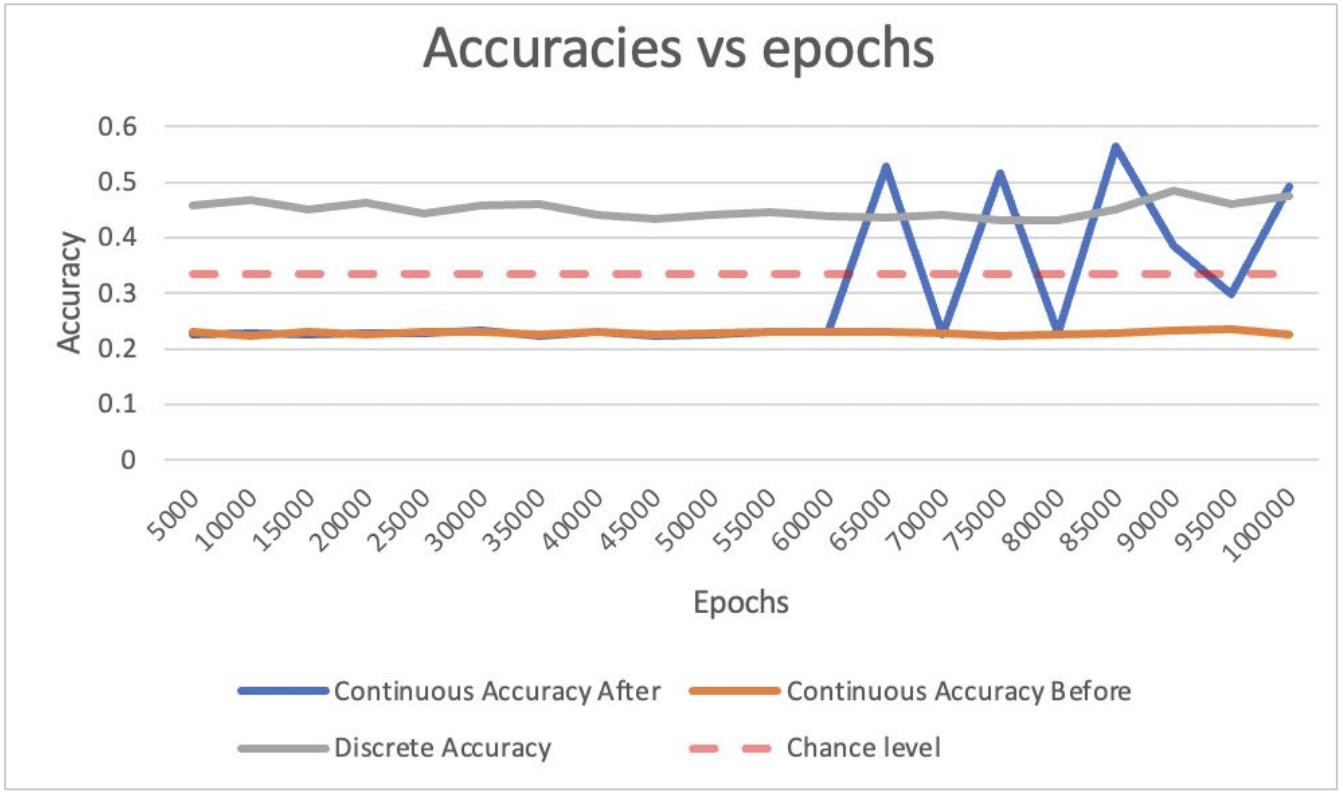
Baseline Representation (no alignment)



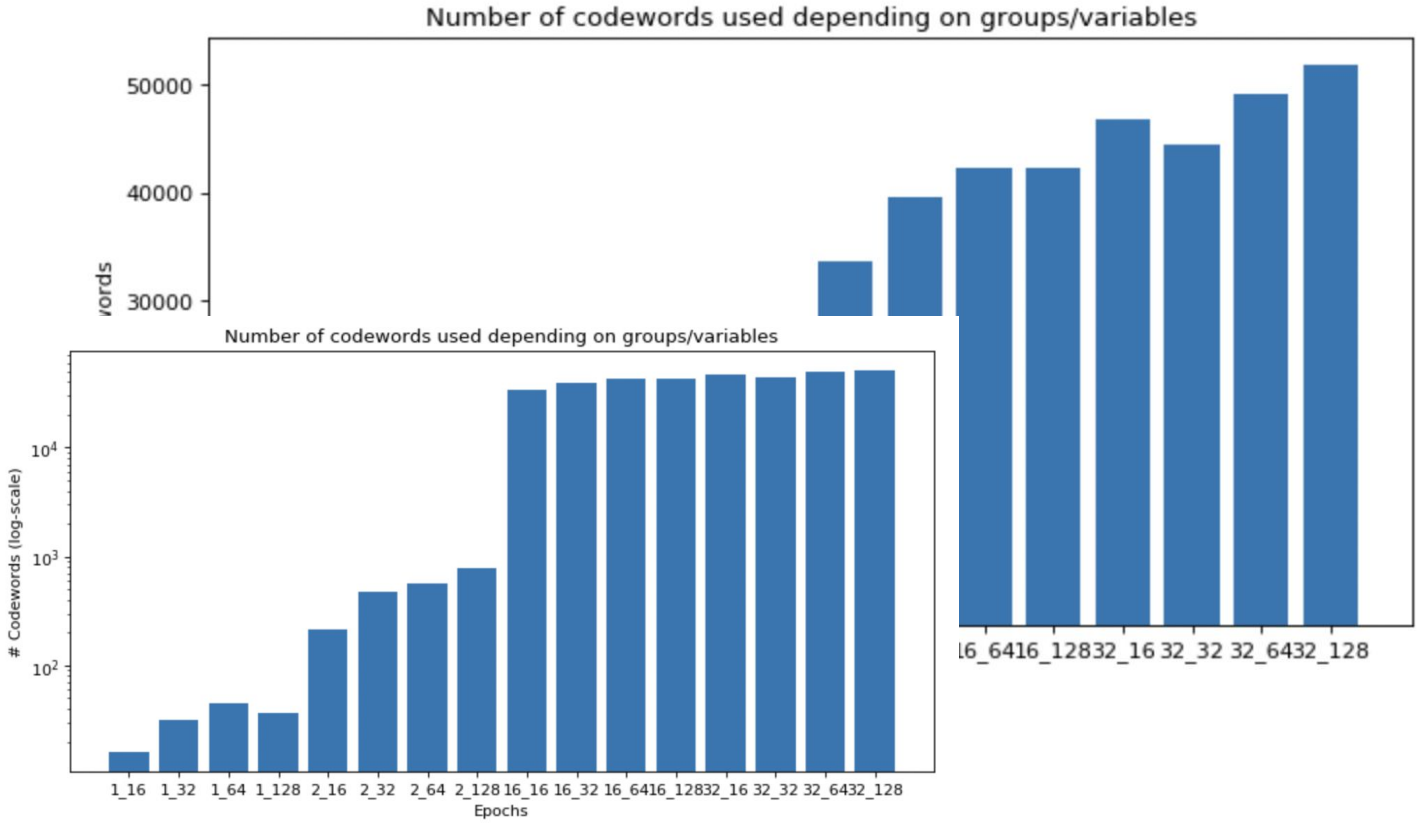




Baseline Representation (no alignment)



Usage of tokens



	#Codewords
1_16	16
1_32	32
1_64	45
1_128	37
2_16	215
2_32	474
2_64	561
2_128	785
16_16	33654
16_32	39516
16_64	42326
16_128	42309
32_16	46744
32_32	44379
32_64	49165
32_128	51758