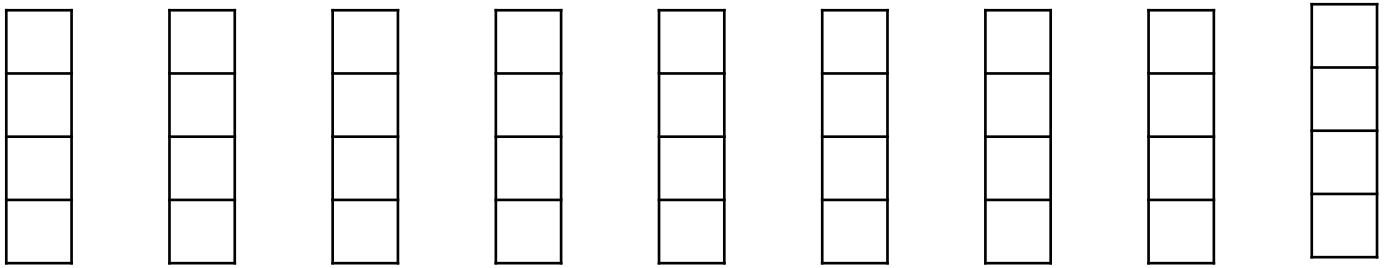
navAlgo

From sampling rate to content rate

Jan Chorowski
NavAlgo
AAAI SAS Workshop 28.02.2022

Representation learning

Unsupervised Representation Learner



Supervised Logistic Regression

b r ih k s aa r er n aa l t er n ih t ih v

Take raw waveforms

(sampled uniformly in time, e.g. 44kHz)

Transform into a sequence of latent feature vectors *(sampled uniformly e.g. every 10 ms)*

Train models on extracted features

Latent vectors and phoneme alignment?

Extract latent vectors every 10ms, regardless of content

✔ easy e.g. strided convolutions

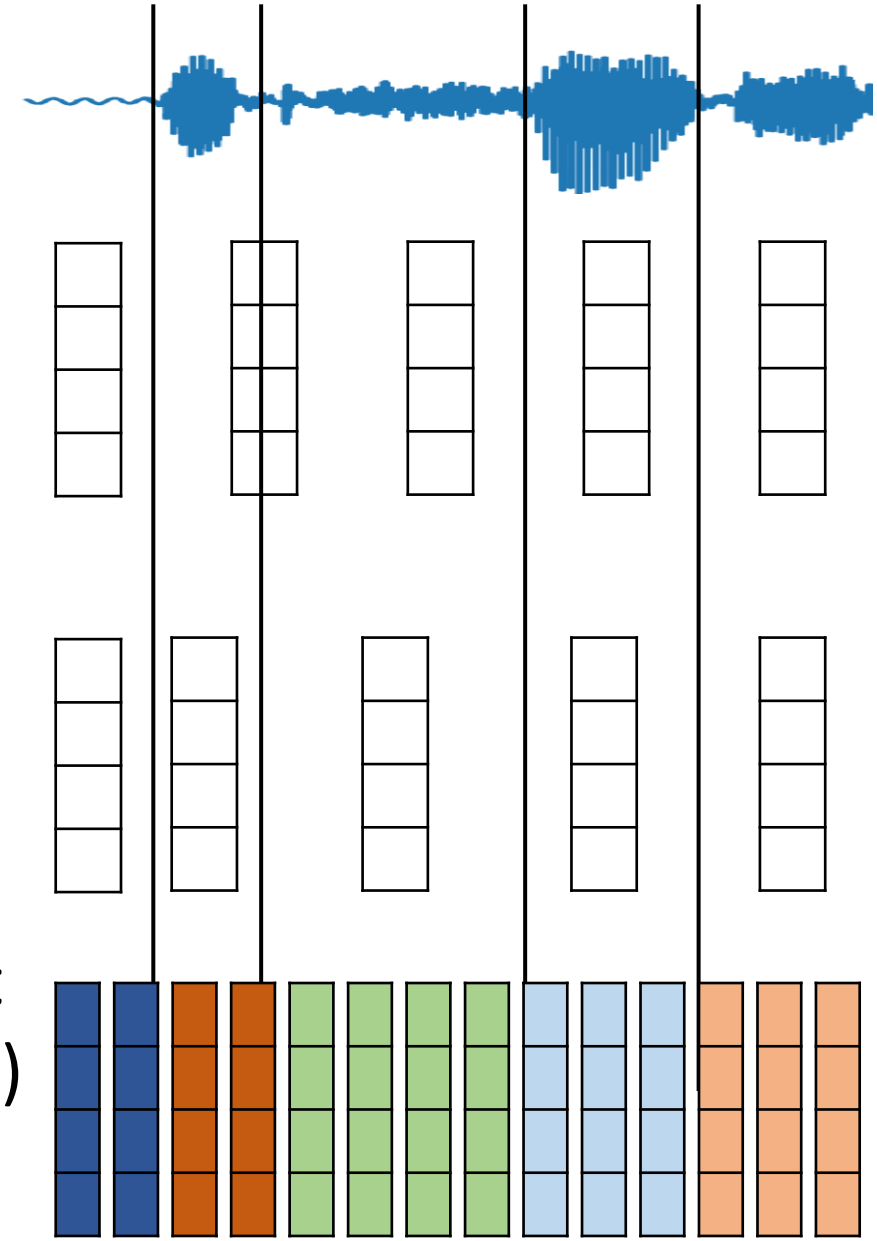
✕ phonemes span many vectors

Align representation to phonemes

How!?

✔ Detect phoneme boundaries

✔ Representation piecewise-constant (latent changes only at boundaries)



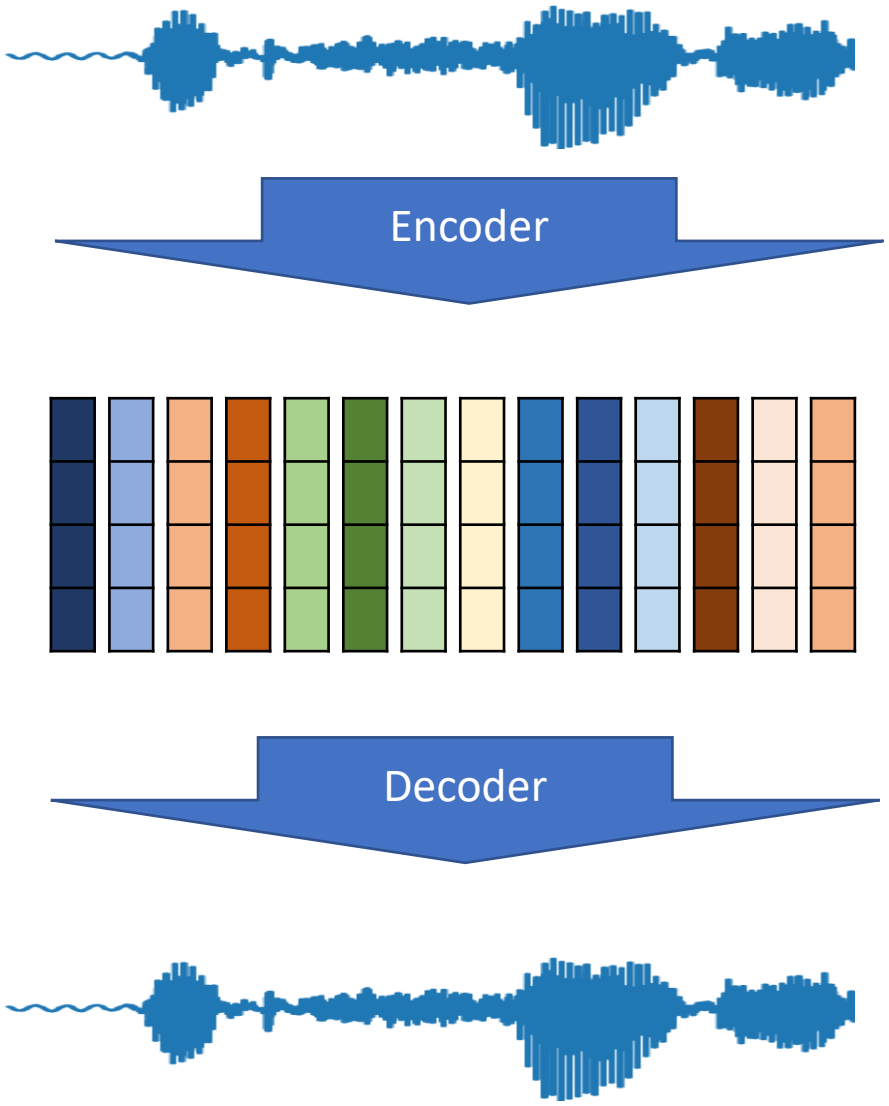
Many possibilities

1. Lossy encodings: autoencoders with piece-wise constant priors
2. Sequence-aware contrastive training losses
3. Post-processing extracted features

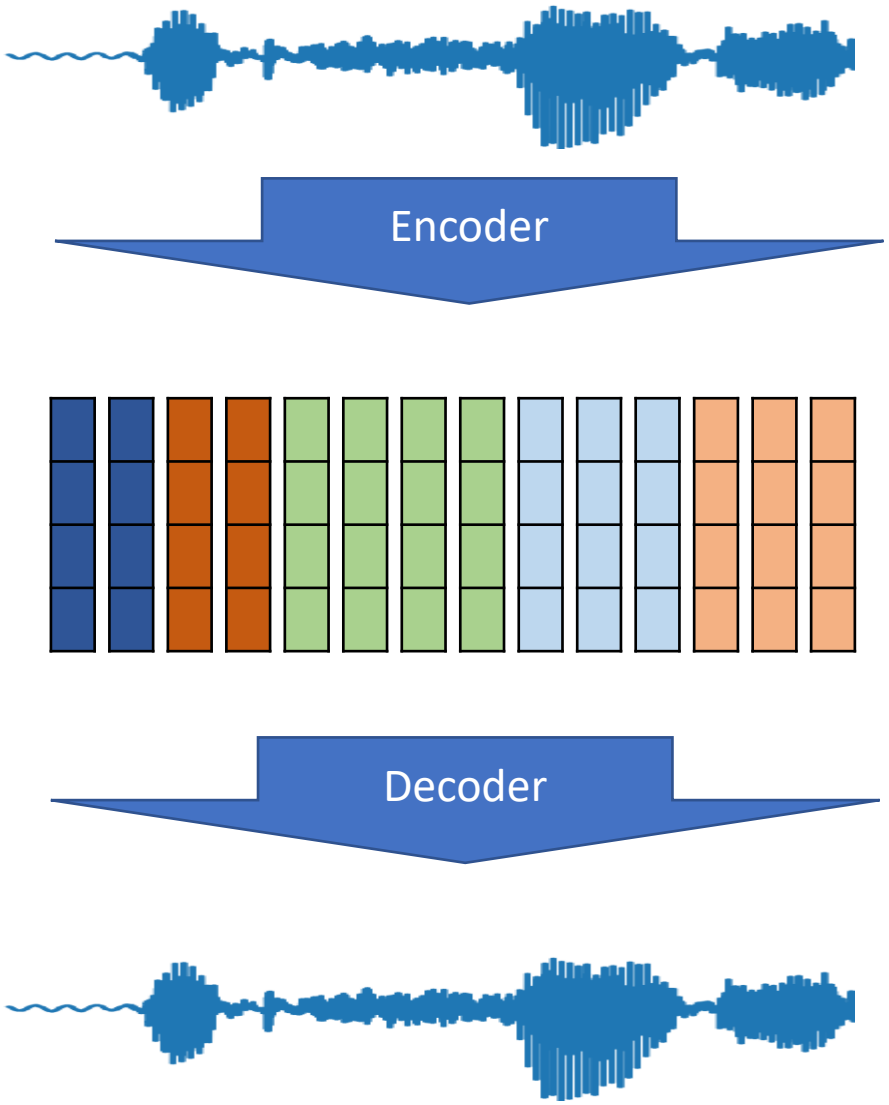
Approach 1: segmental auto-encoders

https://pgr-workshop.github.io/img/PGR009.pdf

Auto-encoders



Auto-encoders



Round representations such that:

1. The rounding error is small
2. There are only a few changes of coding vector

NB: This also work in the discrete case (e.g. VQVAE)

$$\min_{q_1 \dots q_T} \sum_{t=1}^{T} ||x_t - q_t|| \quad \text{subject to:} \sum_{t=1}^{T} [q_{t-1} \neq q_t] = K$$

encoder outputs

VQ token s

expected # of characters

Piecewise-constant results: Scribblelens

Dataset	*Model*	*Downstream CER*	*MLP Accuracy @Bottleneck*	*Token -> Char Accuracy*	*Rand score*	*Mutual information*
Single scribe Tasman	*Unsupervised base*	47%	54%	34%	0.15	0.15
	Piecewise-const VQ bottleneck	**30%**	**65%**	**57%**	**0.23**	**0.36**
All scribes	*Unsupervised base*	60%	42%	30%	0.13	0.10
	Piecewise-const VQ bottleneck	**51%**	**49%**	**39%**	**0.18**	**0.20**

Approach 2: Sequence-aware Aligned CPC

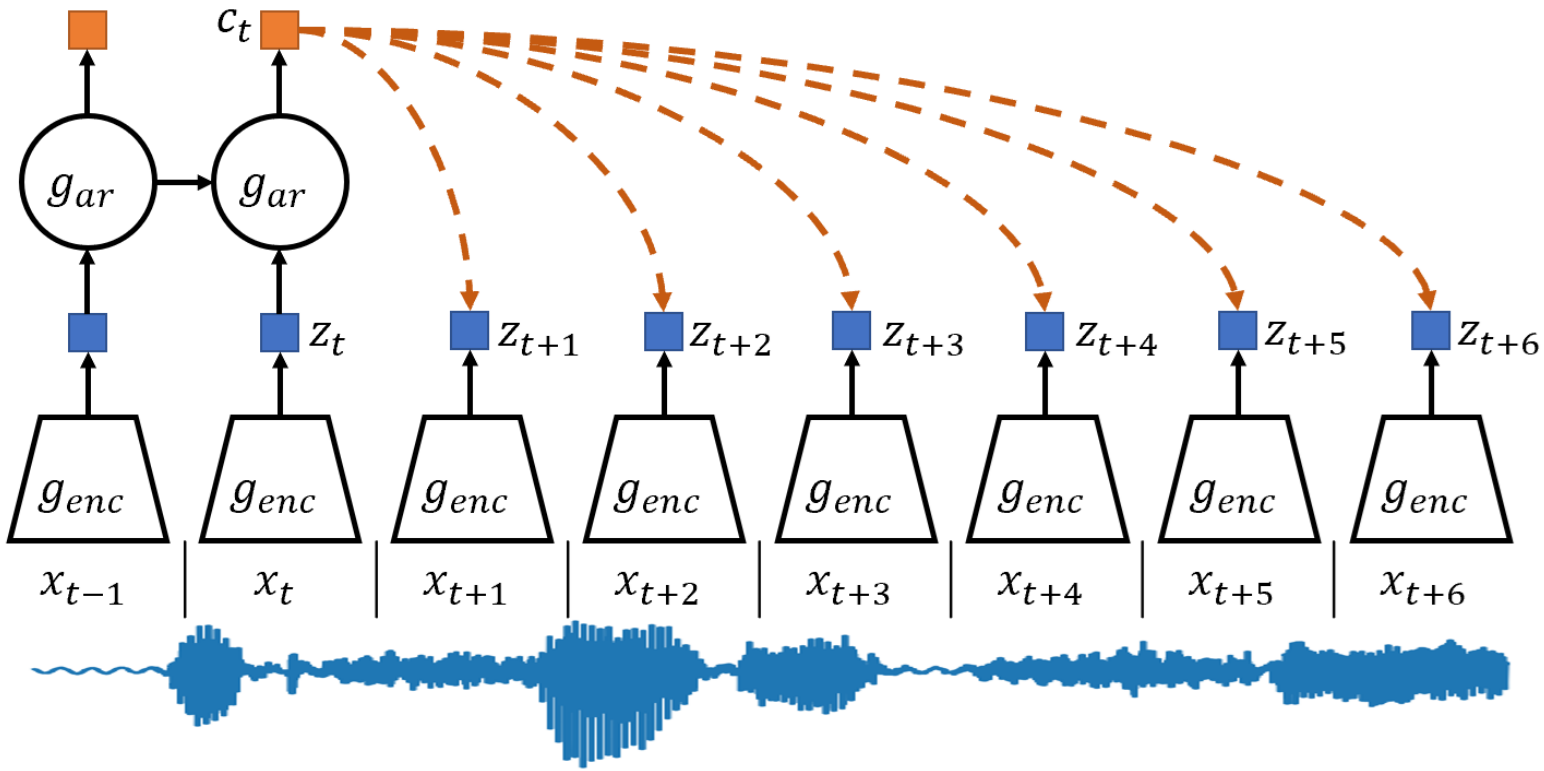
https://arxiv.org/abs/2104.11946 (Interspeech 21)

https://arxiv.org/abs/2110.15909 (ICASSP 22)

CPC: Excellent feature learner

CPC is frame-synchronous

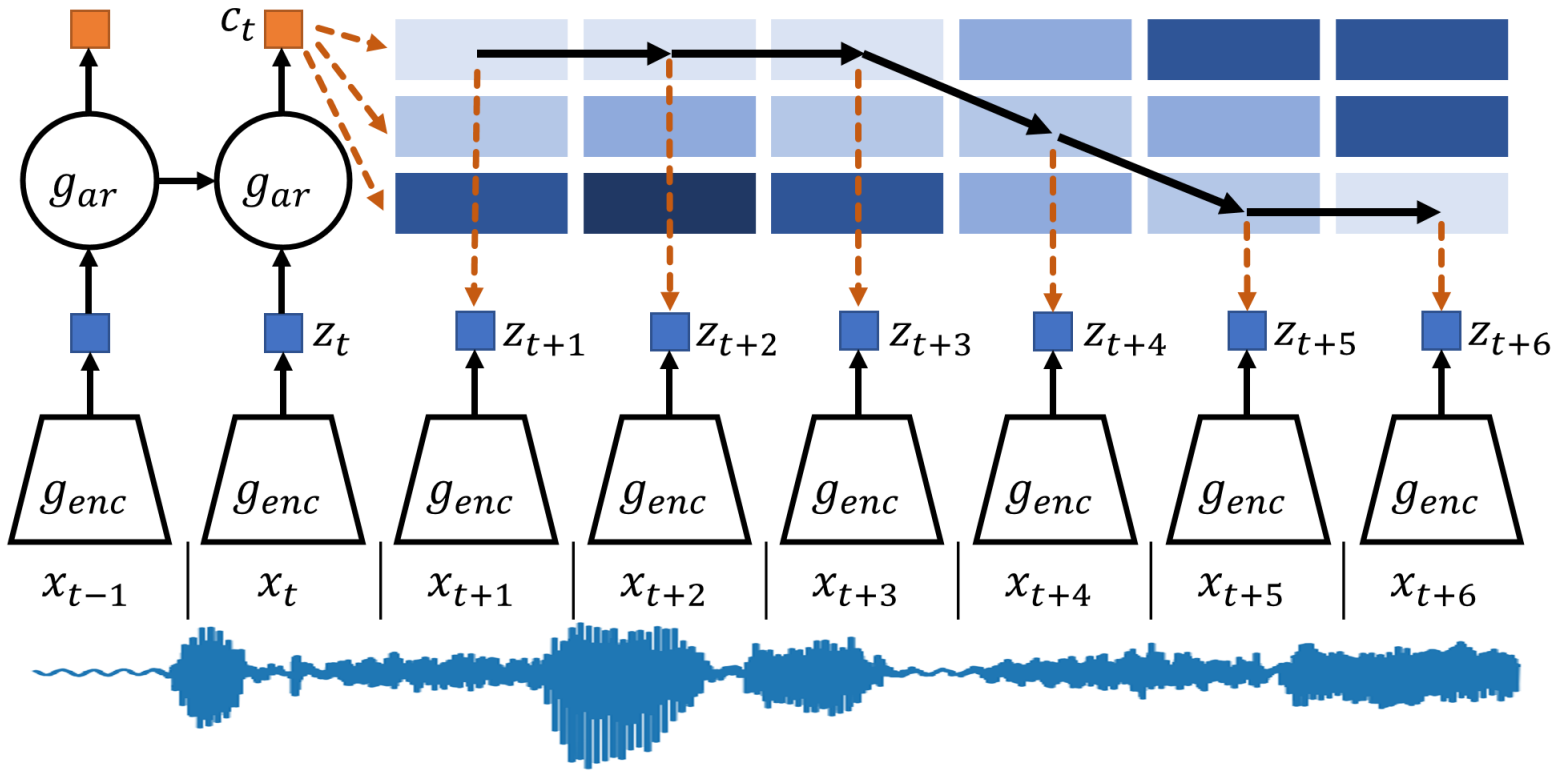
It makes **one** predictions **for each** predicted frame



Aligned CPC: Content-synchronous predictions

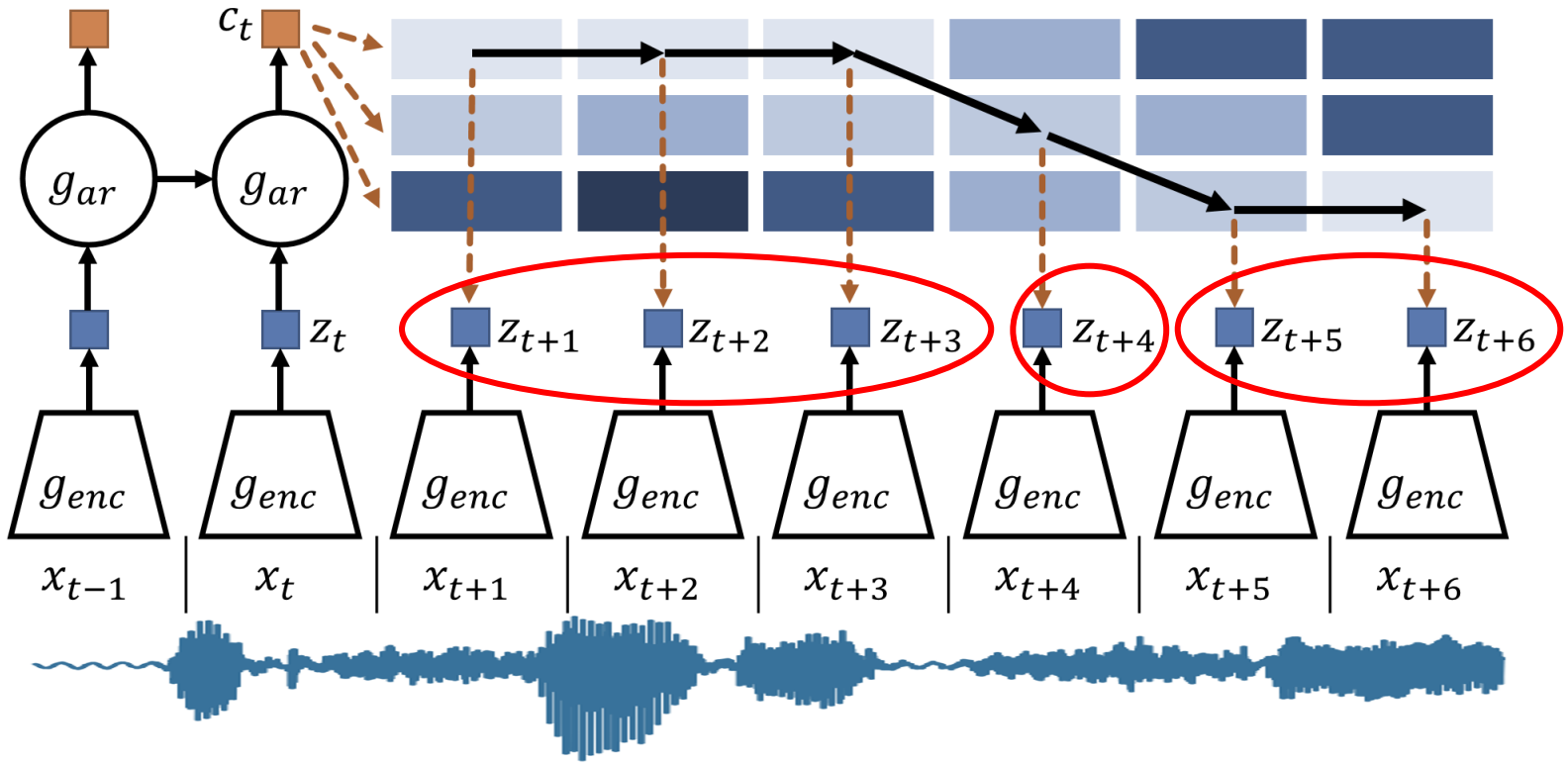
Make **few** predictions, align to **many** frames.

👉Predict exactly **what** will happen, but only **approximately when**.



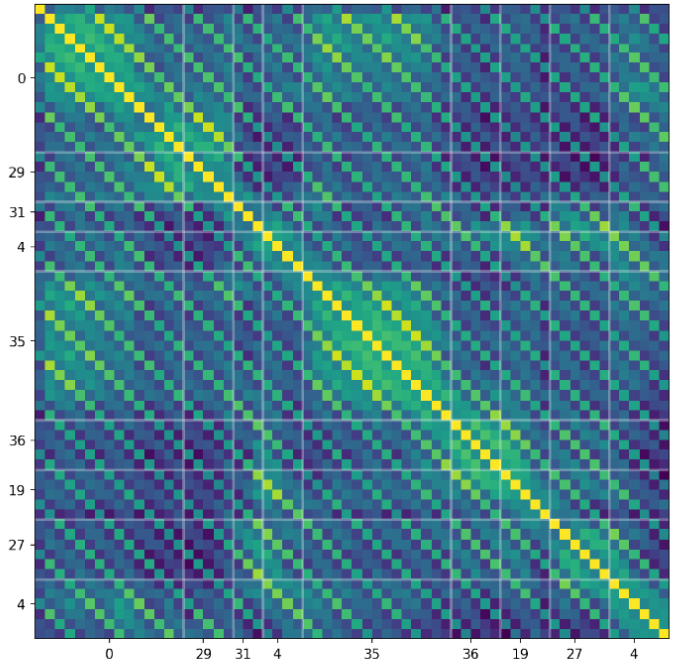
Aligned CPC: piecewise constant representation

- Frames aligned to the same prediction are trained **to be similar**
- Sharp transitions possible between frames aligned to different predictions

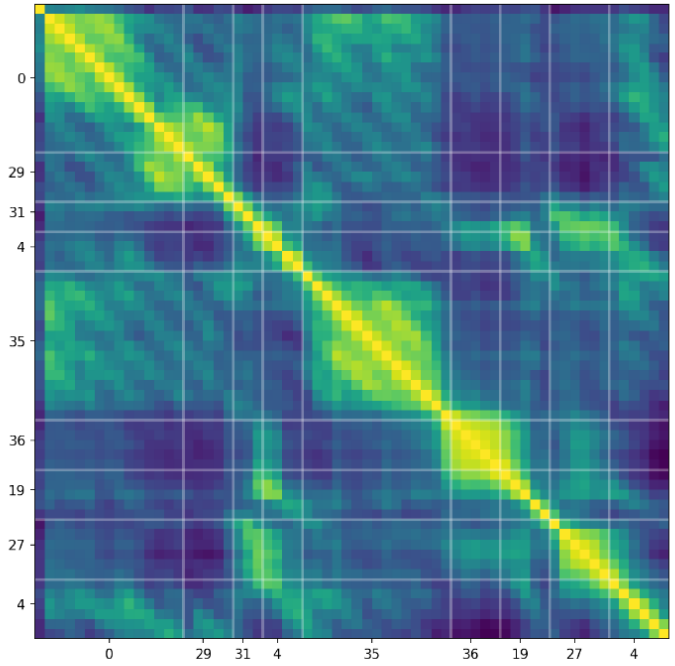


ACPC: Qualitative Results (learned representation self-similarity)

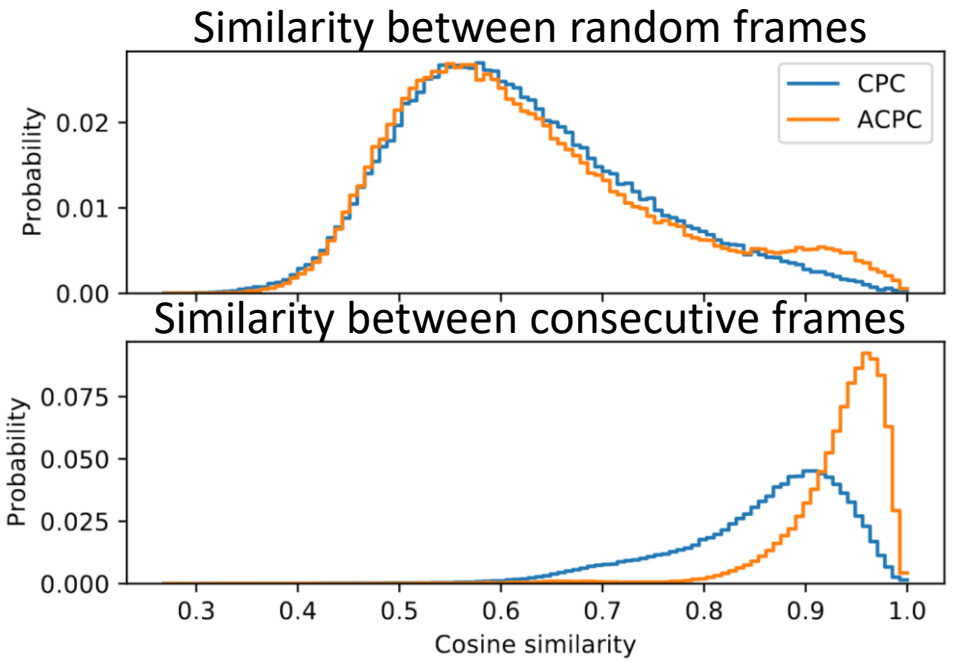
- 👉 Phoneme boundaries more clearly visible in ACPC
- 👉 ACPC prevents CPC artifacts (e.g. encoding relative position, as below)



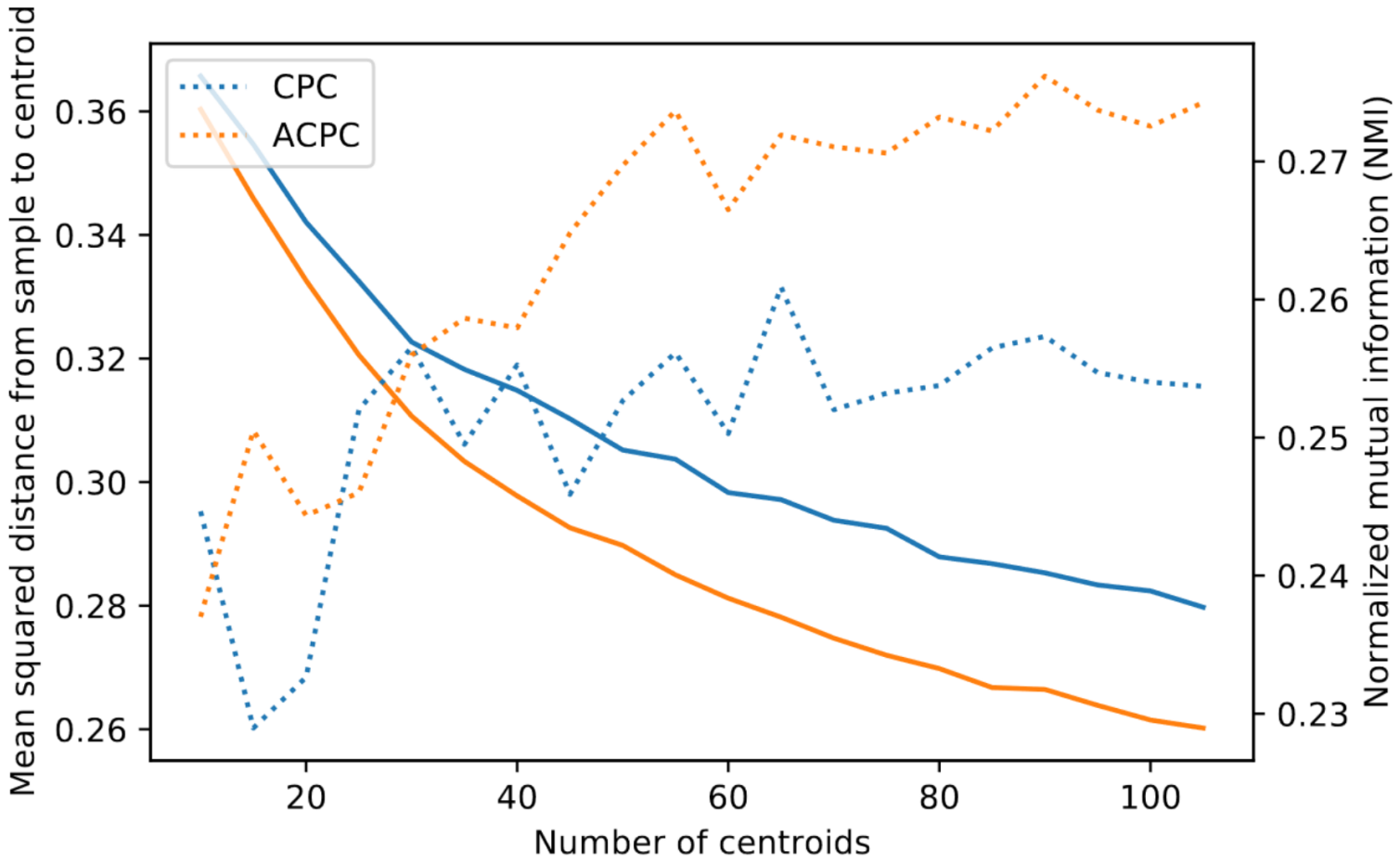
CPC



ACPC (K=8, M=12)



ACPC: Qualitative Results (better clusterability)



ACPC: Comparison with ZeroSpeech 2021 baselines

ABX Scores (lower is better)

	CPC (ZeroSpeech [10])		CPC (our baseline)		ACPC M=12 K=8	
Dev	Within	Across	Within	Across	Within	Across
clean	6.18%	8.02%	6.68%	8.39%	**5.37%**	**7.09%**
other	8.46%	13.59%	9.03%	13.87%	**7.46%**	**12.60%**

Framewise phoneme prediction (higher is better)

Model	Step time	Encoded z_t		Contextual c_t	
		Train	Val	Train	Val
CPC	2.6ms	51.5%	51.2%	67.8%	67.5%
ACPC M=12 K=10	2.4ms	51.8%	51.3%	68.8%	68.4%
ACPC M=12 K=8	2.1ms	52.0%	51.9%	**69.7%**	68.6%
ACPC M=12 K=6	1.8ms	**52.2%**	**52.1%**	69.2%	**68.8%**
ACPC M=12 K=4	**1.5ms**	52.1%	51.9%	69.0%	68.7%
ACPC M=16 K=10	2.4ms	52.1%	51.9%	69.1%	**68.8%**
ACPC M=20 K=12	2.6ms	52.1%	51.9%	68.0%	67.8%

The segmentation-classification performance trade-off

Phone segmentation performance on TIMIT test (top half) and Buckeye test (bottom half)

Model	Precision	Recall	F1	R-val
Kreuk et al.	84.80	**85.77**	85.27	87.35
SCPC	**85.31**	85.36	**85.31**	**87.38**
ACPC	83.41	83.15	83.26	85.64
Kreuk et al.	76.27	78.42	77.31	80.35
SCPC	**77.21**	**78.95**	**78.03**	**80.90**
ACPC	74.44	76.28	75.32	78.66

Phone classification performance on the test split of LibriSpeech train-clean-100

	On z vectors		On c vectors	
Model	Acc.	PER	Acc.	PER
Kreuk et al.	44.87	32.46	-	-
SCPC	43.79	31.62	-	-
ACPC	**47.62**	**24.34**	67.87	18.10

The best performing models on unsupervised phone segmentation perform badly on phone classification tasks, and vice versa.

[1] Kreuk et al. “Self-Supervised Contrastive Learning for Unsupervised Phoneme Segmentation” arxiv.org/abs/2007.13465
[2] Bhati et al. “Segmental Contrastive Predictive Coding for Unsupervised Word Segmentation” arxiv.org/pdf/2106.02170
[3] Chorowski et al. “Aligned Contrastive Predictive Coding” arxiv.org/abs/2104.11946

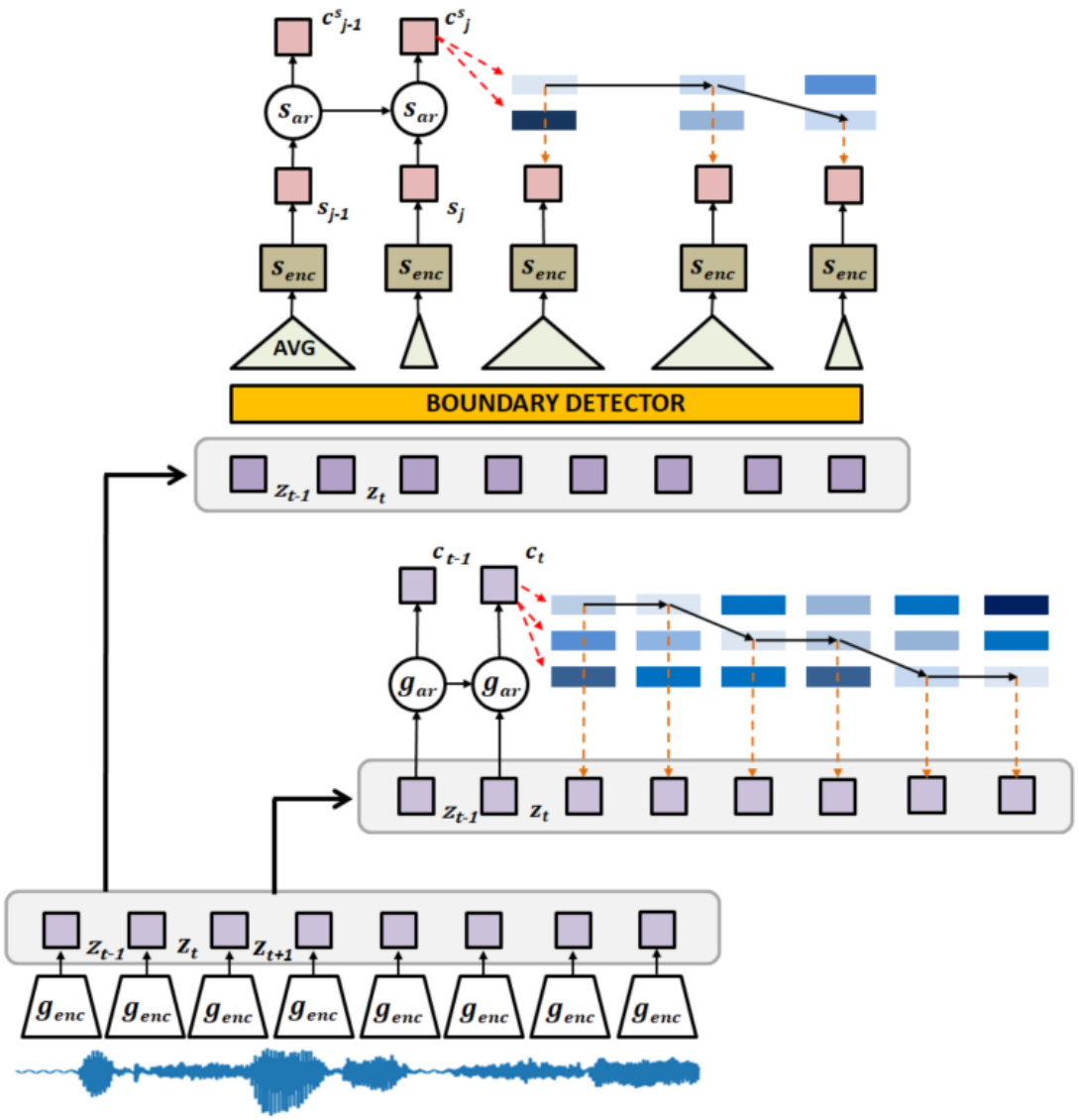
Studying the trade-off

Differences between CPC models for segmentation and classification.

Models for segmentation	**Models for classification**
• Do not use context builders at the frame-level. • Optimize a contrastive loss between adjacent representations, making the model sensitive to spectral changes • Multilevel modeling.	• Have context builders and prediction networks. • Predict many steps ahead.

Context builders cause a representation shift, hurting segmentation performance, but are necessary to capture phone class information.

Multi-level ACPC (mACPC): augmenting ACPC for segmentation



incorporate features from segmentation models into ACPC

Frame-level ACPC module extracts representations.

Its output is processed by a boundary detector, which predicts boundaries and averages latents within those boundaries to produce segment representations. Finally, the segment-level ACPC module learns to predict higher-level features.

An auxiliary contrastive loss between adjacent representations extracted by the frame-level module is applied to optimize for detection of spectral changes.

[4] *Cuervo et al. "Contrastive prediction strategies for unsupervised segmentation and categorization of phonemes and words"*

Hierarchical CPC Results

mACPC achieves competitive phone segmentation performance

Model	Precision	Recall	F1	R-val
Kreuk et al.	84.80	**85.77**	85.27	87.35
SCPC	**85.31**	85.36	**85.31**	**87.38**
ACPC	83.41	83.15	83.26	85.64
ACPC + Kreuk et. al. loss	83.68	84.74	84.69	86.86
mACPC	82.53	83.05	82.78	85.26
mACPC + Kreuk et. al. loss	84.63	84.79	84.70	86.86
Kreuk et al.	76.27	78.42	77.31	80.35
SCPC	**77.21**	**78.95**	**78.03**	**80.90**
ACPC	74.44	76.28	75.32	78.66
ACPC + Kreuk et. al. loss	74.68	76.59	75.59	78.88
mACPC	74.00	76.04	74.98	78.34
mACPC + Kreuk et. al. loss	74.70	76.81	75.72	78.97

mACPC improves word segmentation performance

Model	Precision	Recall	F1	R-val
SCPC	36.23	**32.75**	34.33	45.39
mACPC	**42.06**	30.32	**35.05**	**47.40**
mACPC + Kreuk et. al. loss	40.36	30.86	34.83	47.11

mACPC improves categorization performance

	On z vectors		On c vectors	
Model	Acc.	PER	Acc.	PER
Kreuk et al.	44.87	32.46	-	-
SCPC	43.79	31.62	-	-
ACPC	47.62	24.34	67.87	18.10
ACPC + Kreuk et. al. loss	47.82	25.93	67.99	18.15
mACPC	50.98	**21.15**	69.97	16.91
mACPC + Kreuk et. al. loss	**51.64**	21.69	**70.25**	**16.65**

Model	ABX within	ABX across
Kreuk et al.	10.93	19.11
SCPC	20.18	16.26
ACPC	5.78	7.93
ACPC + Kreuk et. al. loss	5.67	7.78
mACPC	5.28	7.13
mACPC + Kreuk et. al. loss	**5.13**	**6.84**

[1] Kreuk et al. “Self-Supervised Contrastive Learning for Unsupervised Phoneme Segmentation” arxiv.org/abs/2007.13465
[2] S. Bhati et al. “Segmental Contrastive Predictive Coding for Unsupervised Word Segmentation” arxiv.org/pdf/2106.02170
[3] Chorowski et al. “Aligned Contrastive Predictive Coding” arxiv.org/abs/2104.11946

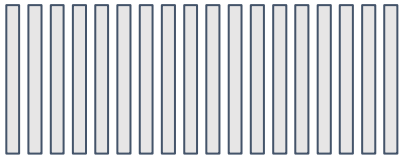
Approach 3:
Feature post-processing

(Preliminary, unpublished)

Intuitions

audio

CPC



k-means
pseudo-phones

heuristics

SentencePiece
pseudo-words

Viterbi decoding
cleaned pseudo-phones

Decoding to promote frequent pieces slightly improves:

1. The mapping of cluster IDs to phonemes *(cluster segm.)*
2. The mapping of sentecepiece IDs to phonemes *(piece segm.)*

Model	Cluster segm.	Piece segm.
Baseline	57.51%	-
+ STSLM 250	58.07%	58.13%
+ STSLM 500	58.34%	59.67%
+ STSLM 1000	**58.42%**	60.95%
+ STSLM 2000	58.33%	61.45%
+ STSLM 5000	58.35%	**62.59%**

Summary

1. Features improve when piecewise-constant segmentations are promoted.

   Aligned CPC (ACPC):

   1. Promotes piecewise-constant latent representations
   2. Improves learned representation clusterability
   3. Yields better results on downstream tasks

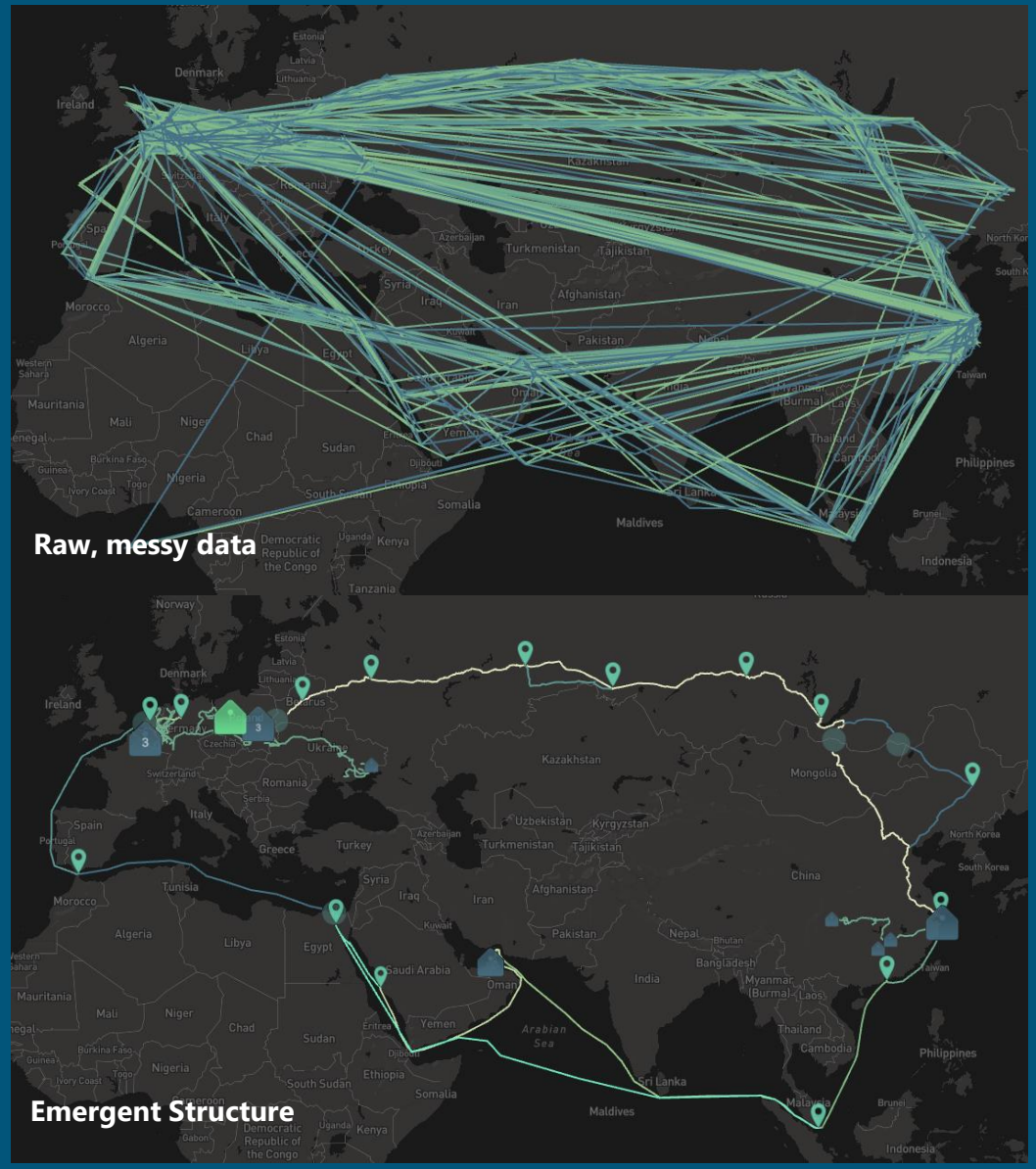
2. Detected boundaries are challenging to use in a hierarchical architecture

3. There is ongoing follow-up work to discover the best segmentations and have deep models benefit from them.

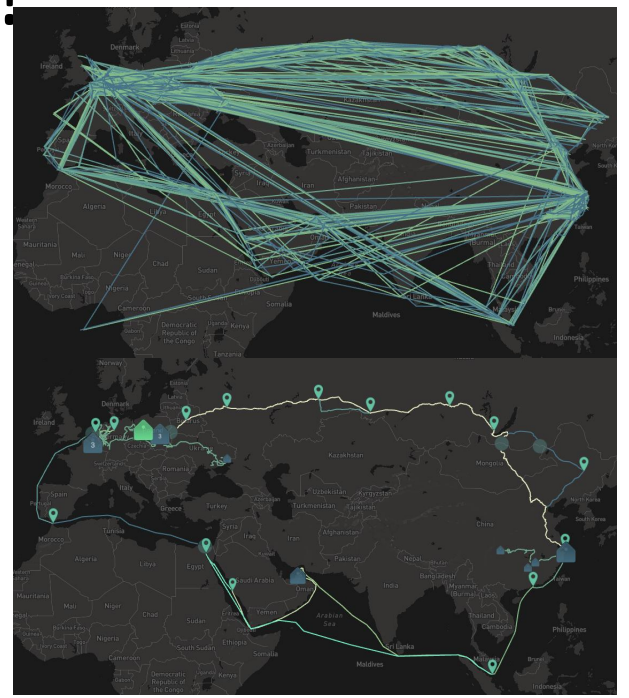
navAlgo

Mess in

Structure out



NavAlgo's methods link to ASR and NLP!



Input: streams of GPS location scans

Output: map with typical routes

Gist of approach:

1. Build an initial map (or take on from e.g. OSM)
2. Match all traces to the map
3. Improve the map
   (e.g. remove unused fragments, or move map edges to data)
4. Repeat and enjoy!

Thank you!

navAlgo

Papers

https://arxiv.org/abs/1901.08810

https://pgr-workshop.github.io/img/PGR009.pdf

https://arxiv.org/abs/2104.11946

https://arxiv.org/abs/2106.11603

https://arxiv.org/abs/2110.15909

Code:

https://github.com/distsup/DistSup

https://github.com/chorowski-lab/CPC_audio

Questions?

PS: NavAlgo is hiring!!!

PSPS:
Help Ukraine! Support Russians and Belarussians protesting war in their counties!