Multimodal self-supervised learning for real-world signals

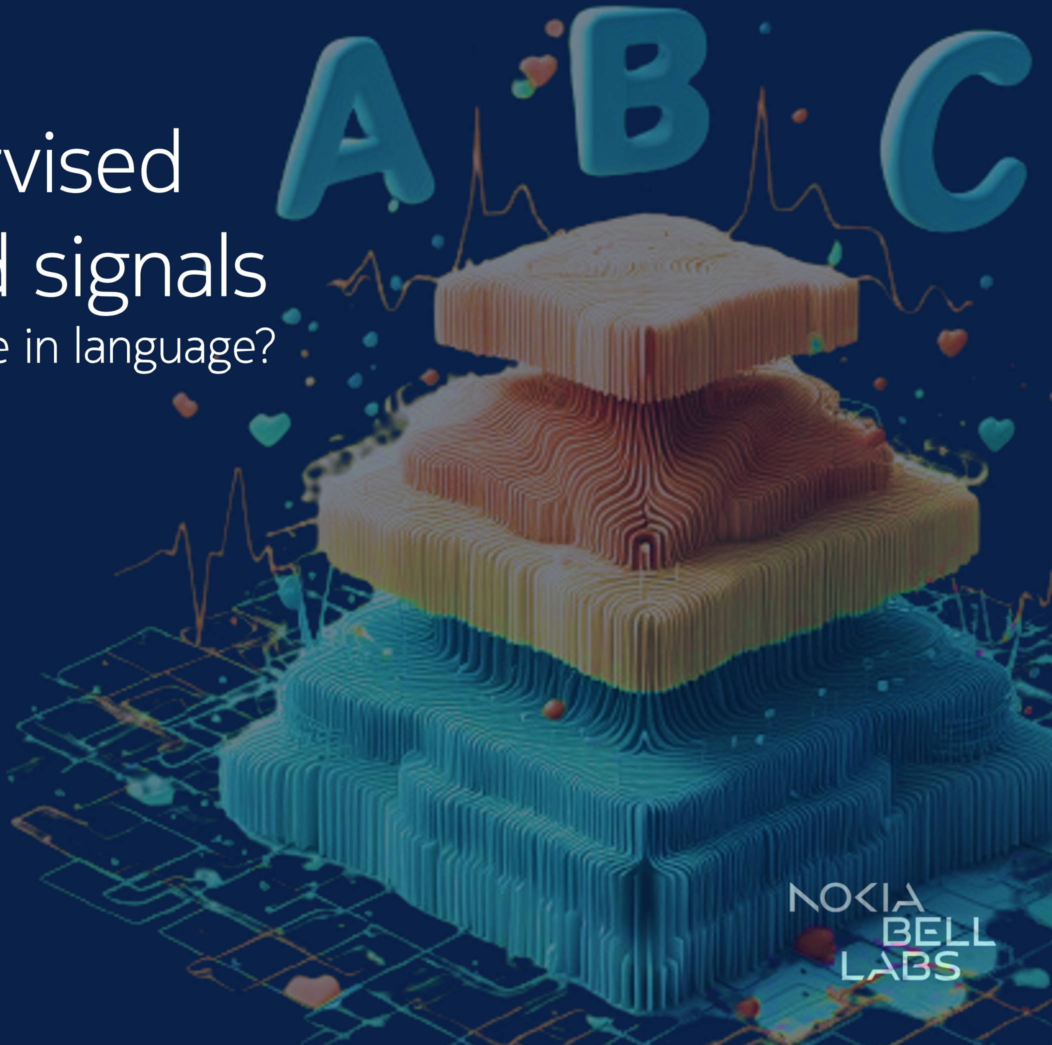
Does the key to specialized models lie in language?

Dimitris Spathis

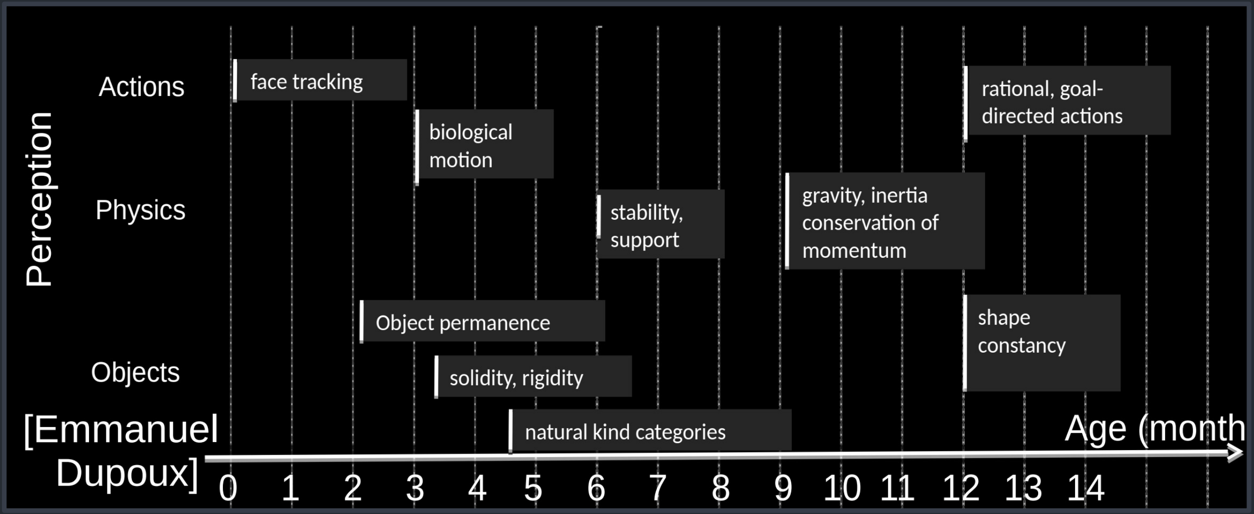
Sr Researcher
Nokia Bell Labs
Cambridge, UK

dispathis.com

Microsoft AI & Pizza talk • November 2023



Why don't models learn like humans or animals?

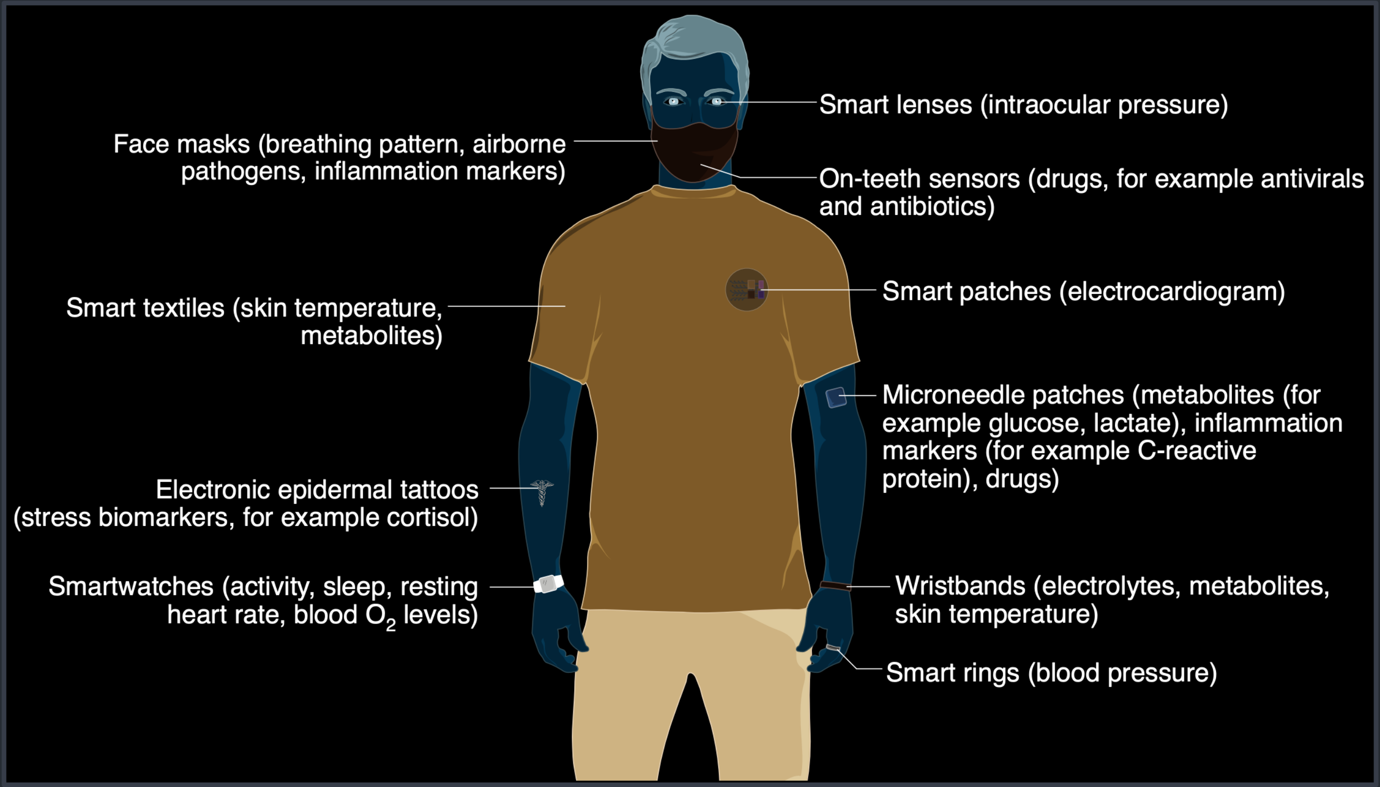


How do babies learn to interact with the world in a few months?

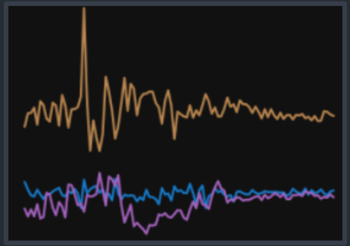
How do teenagers learn to drive with only a few hours of training?

Image credit: Yann Lecun

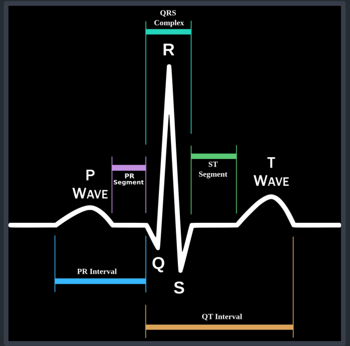
Multimodal data → structurally different signals



Accelerometers



Heart sensors



NOKIA BELL LABS

0.05/label

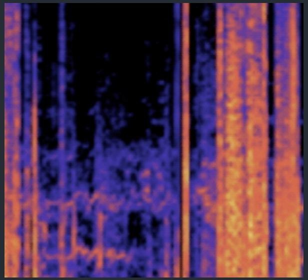
DOG

easy

💰 0.05/label

DOG

easy



?

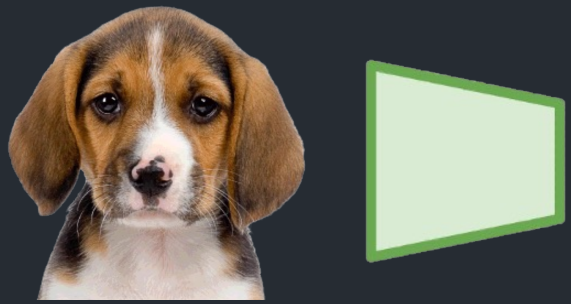
hard

NOKIA BELL LABS

>80x

💰0.05/label → 💰4.00+*/label

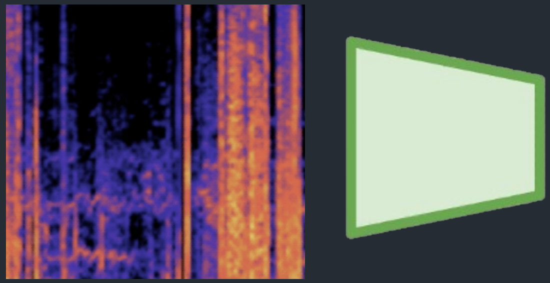
*assuming a sleep technician charging $50/h
and 90-120 sleep stage transitions per 8 hours of sleep



DOG

easy

IMAGE RECOGNITION



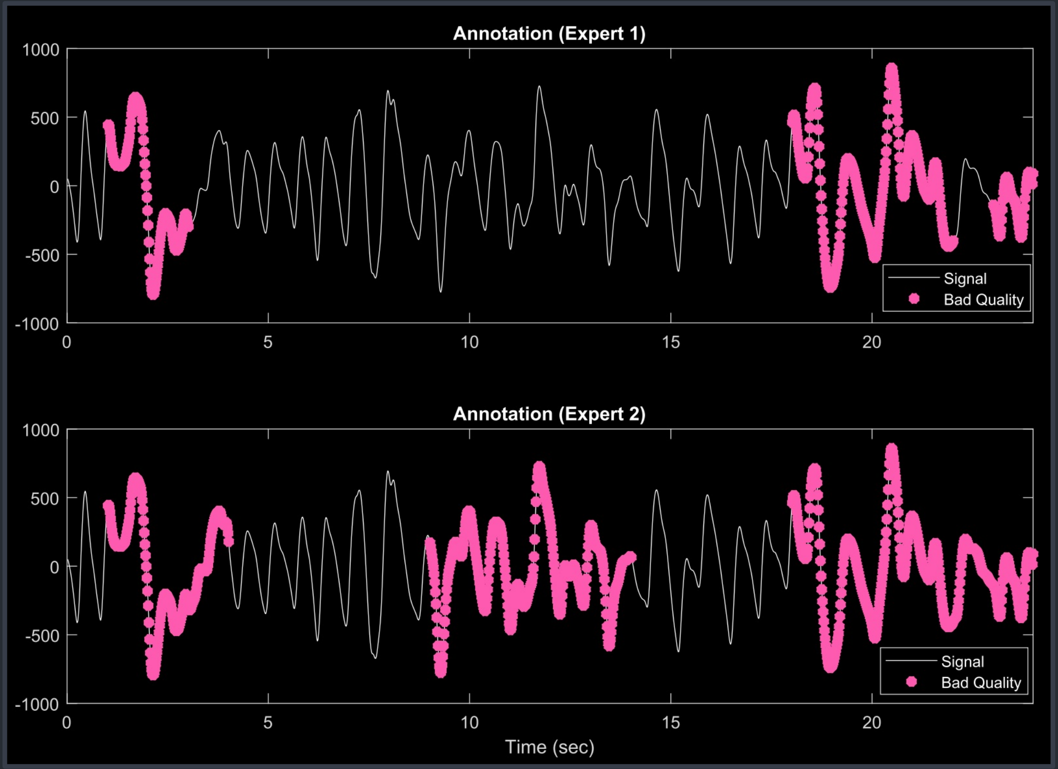
?

hard

SENSOR-BASED SLEEP TRACKING

NOKIA BELL LABS

Signal annotation is not straightforward and sometimes infeasible



Annotation is supposed to be the golden standard in collecting ground-truth but rater (dis)agreement introduces further confusion to the models

For some tasks such as sensor-based activity recognition, an additional video recording is required for annotation, which cannot scale to real-world settings

Sabetti et al., 2019, Informatics in Medicine Unlocked

NOKIA BELL LABS

Self-supervised learning uses existing data as prediction targets

- Predict any part of the input from any other part.
- Predict the future from the past.
- Predict the future from the recent past.
- Predict the past from the present.
- Predict the top from the bottom.
- Predict the occluded from the visible
- Pretend there is a part of the input you don’t know and predict that.

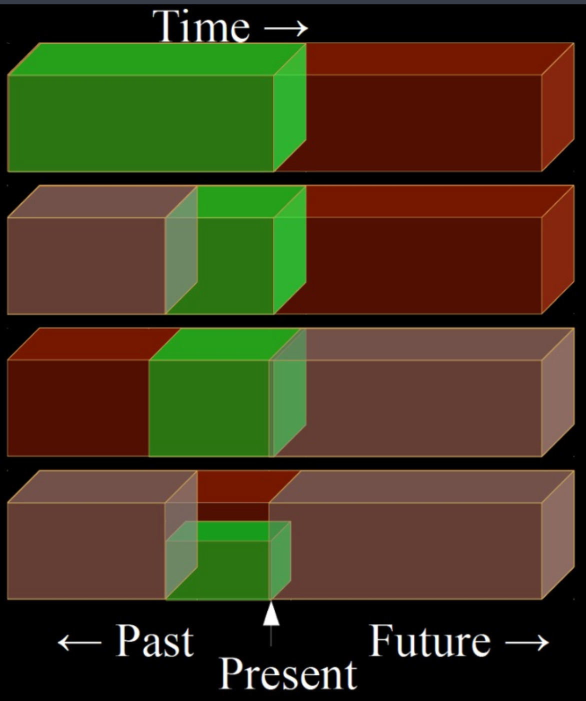
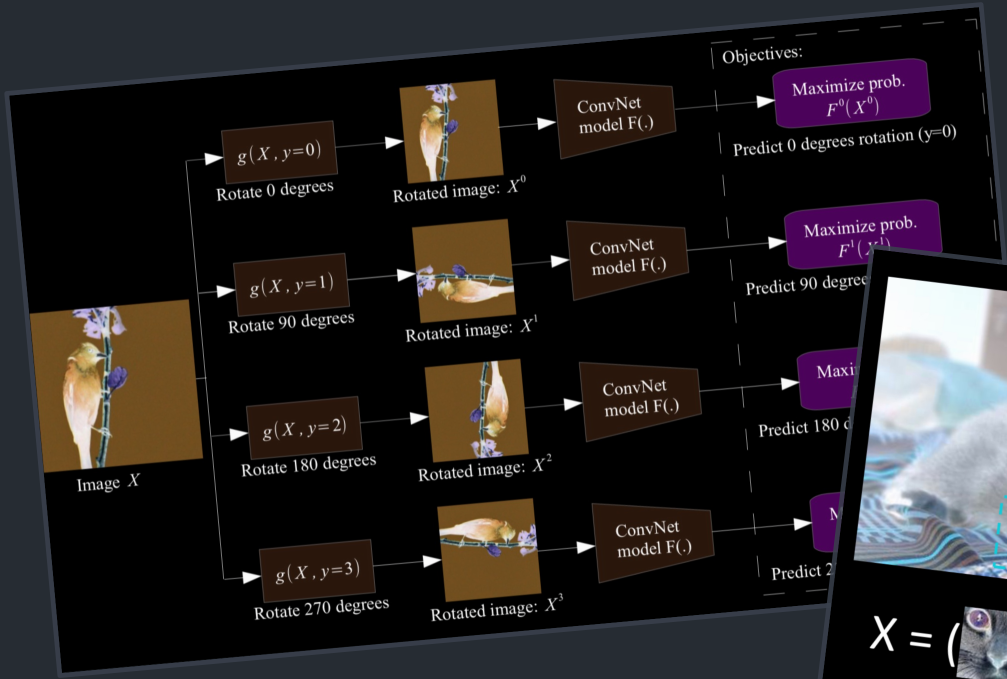
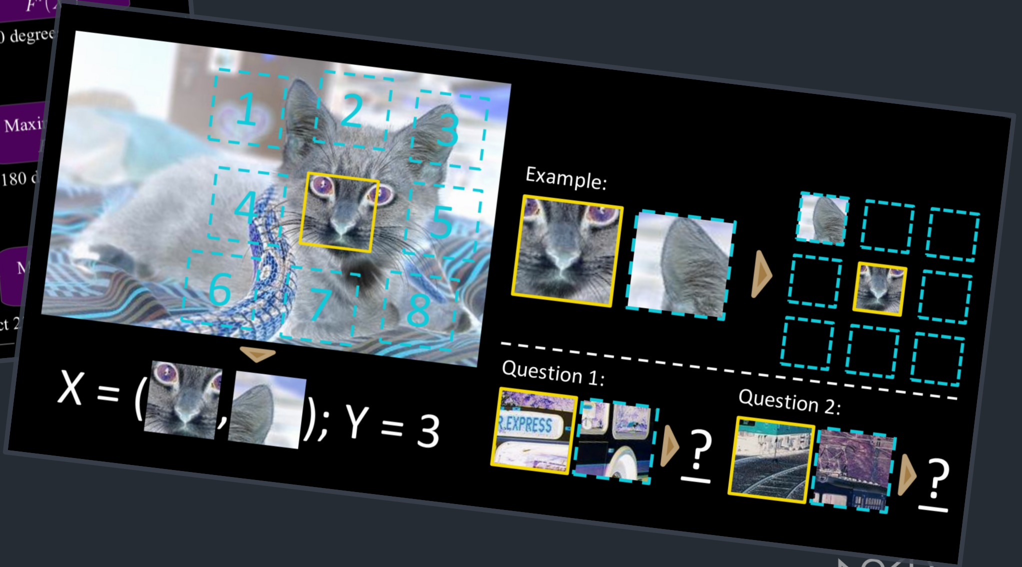


Image credit: Yann Lecun

NOKIA BELL LABS

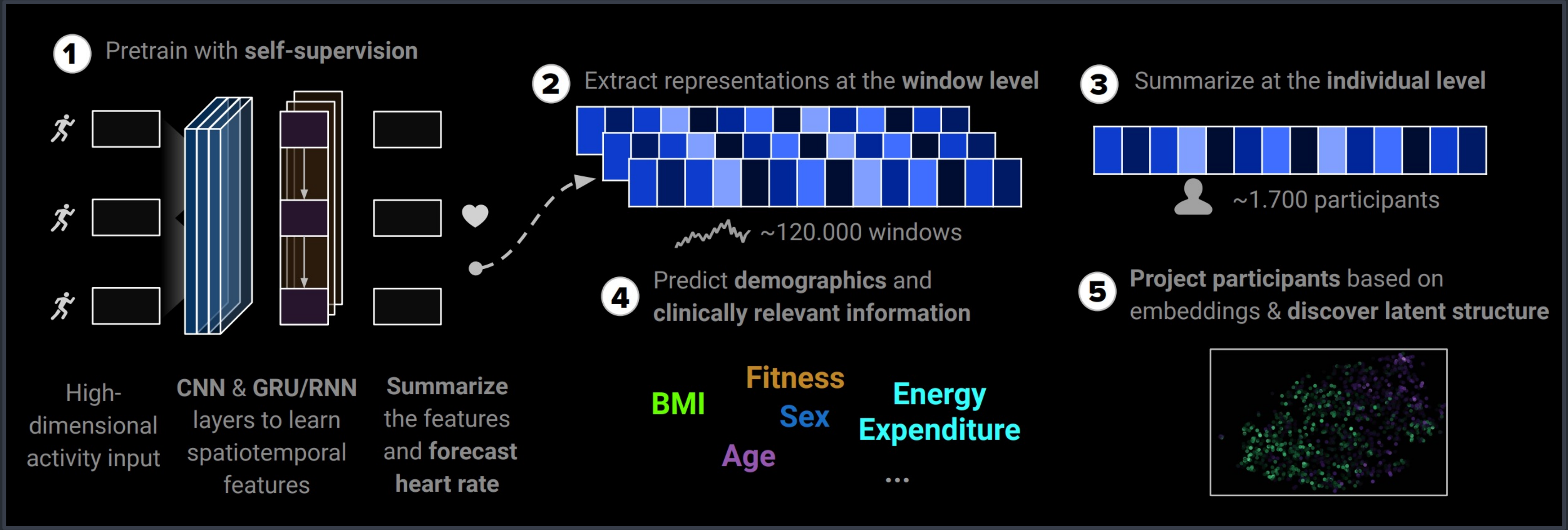
The first models were based on heuristic (pretext) tasks





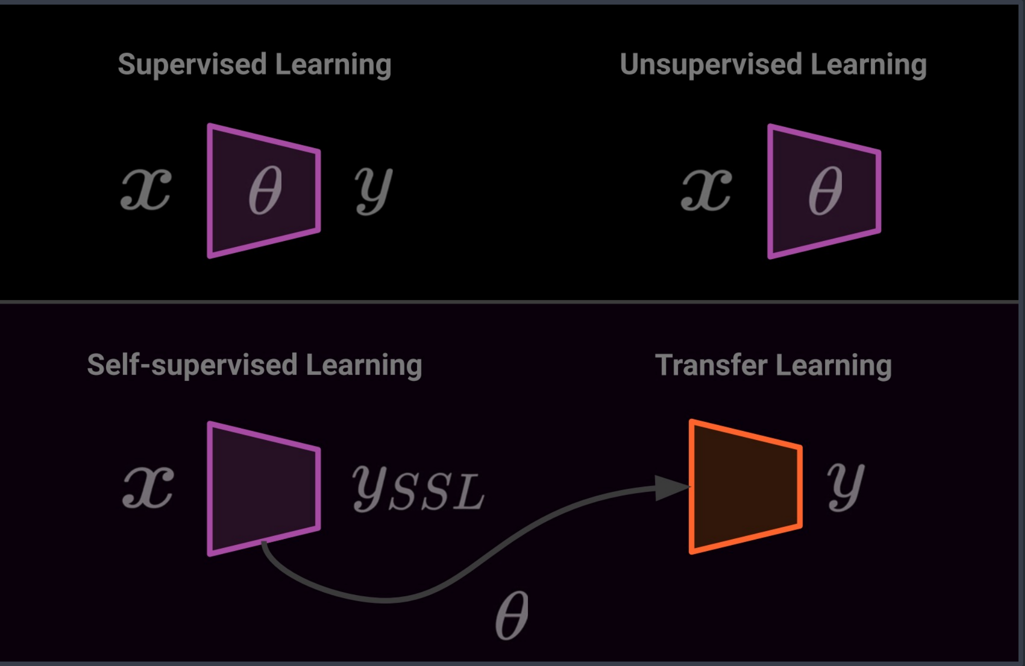
https://lilianweng.github.io/

We can even create pretext tasks across different modalities



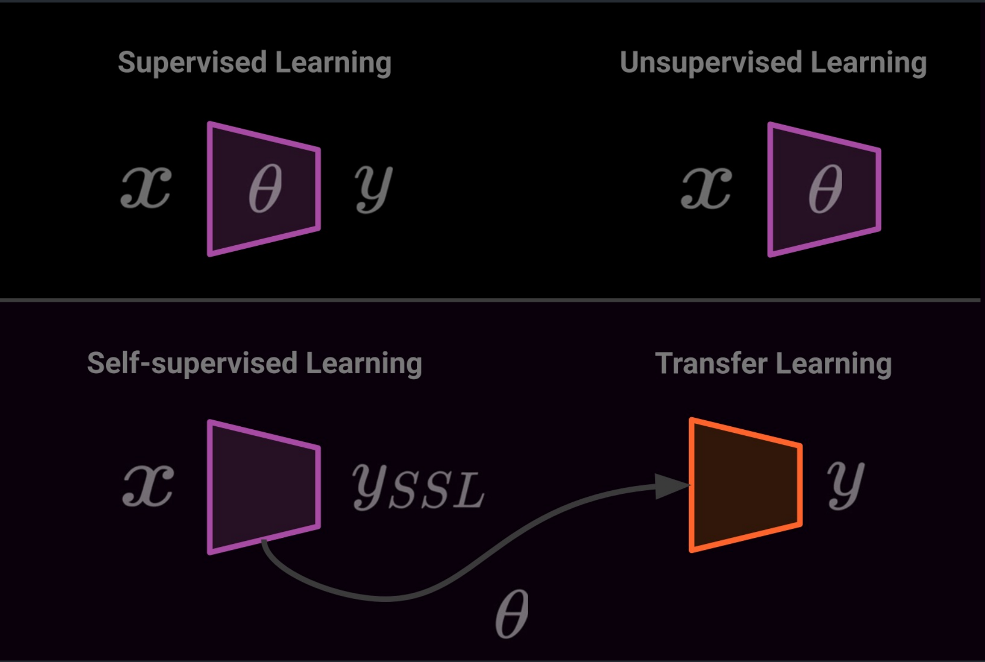
Spathis et al., CHIL 2021

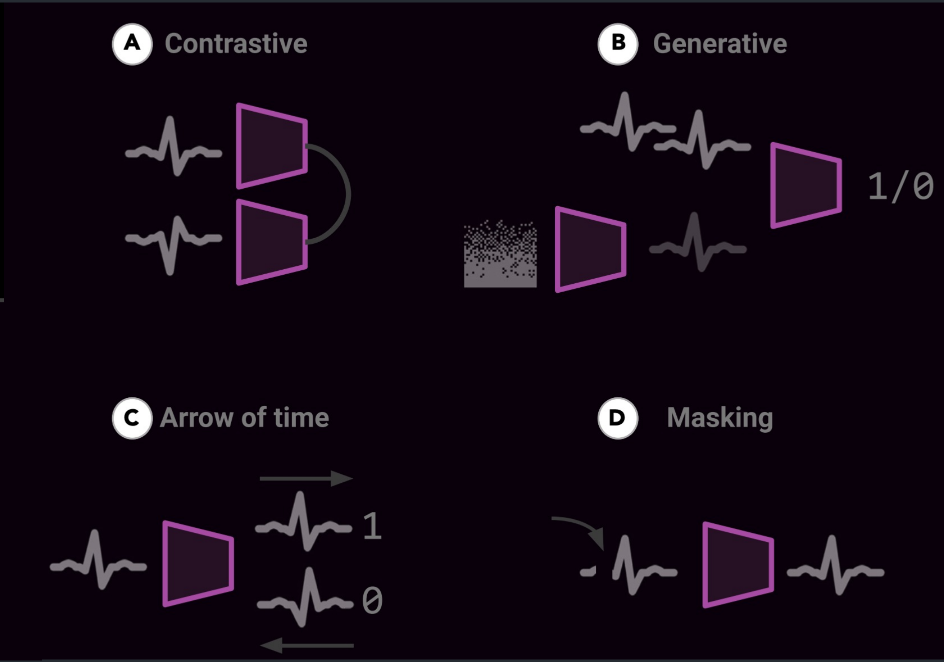
Self-supervised learning enables transferrable models



Spathis et al., Cell Patterns 2022

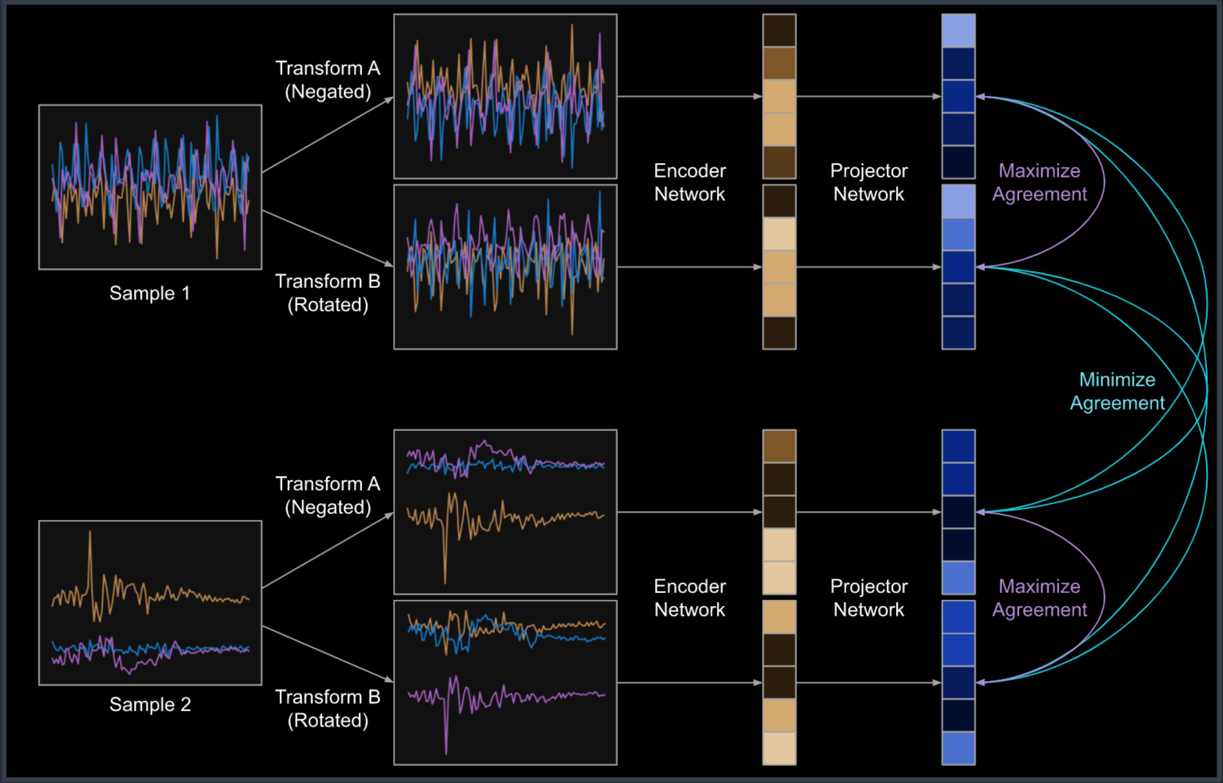
And it comes in different variants





Spathis et al., Cell Patterns 2022

Current approaches focus on pre-processing & unimodal data

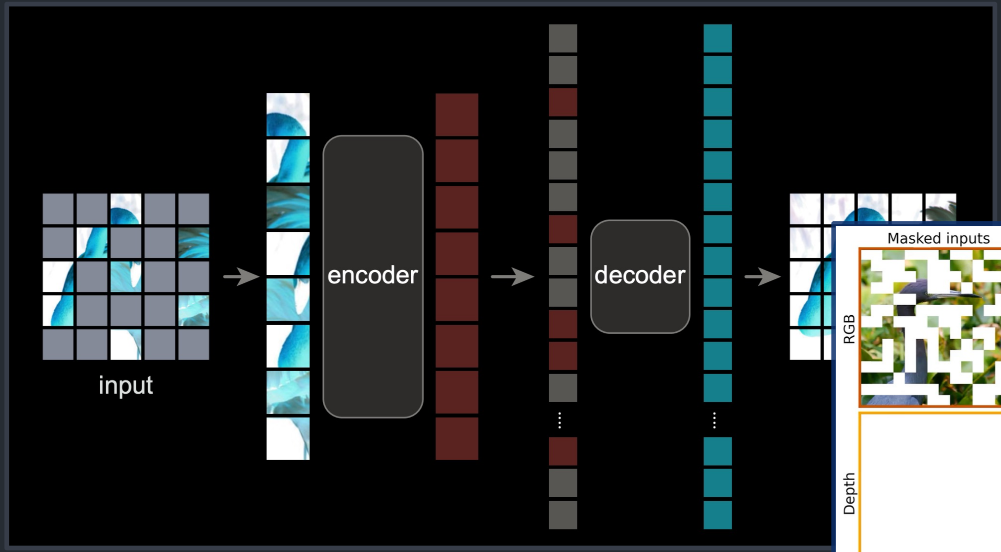


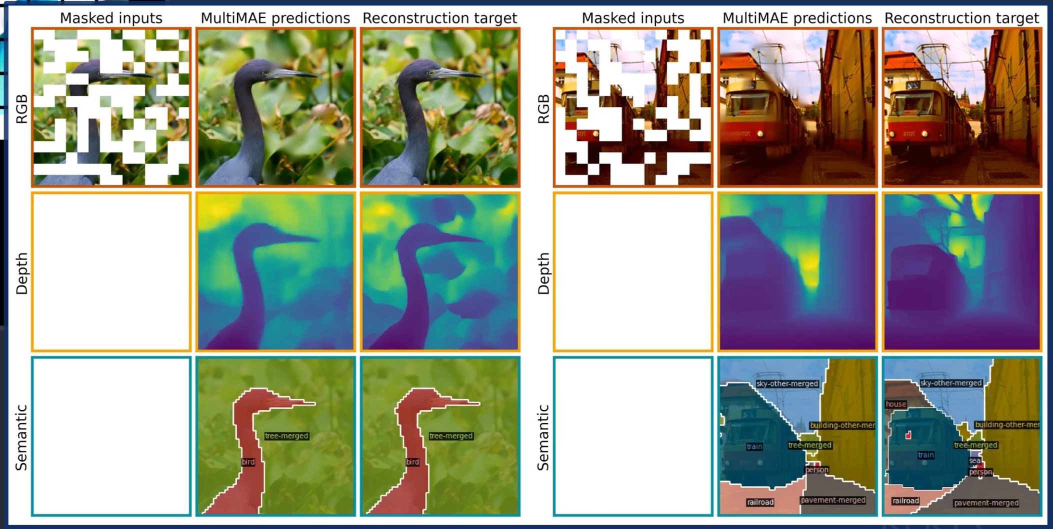
We have to create both positive and negative pairs (not straightforward to pick)

There are no considerations for learning both within and across different modalities

https://github.com/iantangc/ContrastiveLearningHAR

Masked Autoencoders offer a simpler architecture based on masking

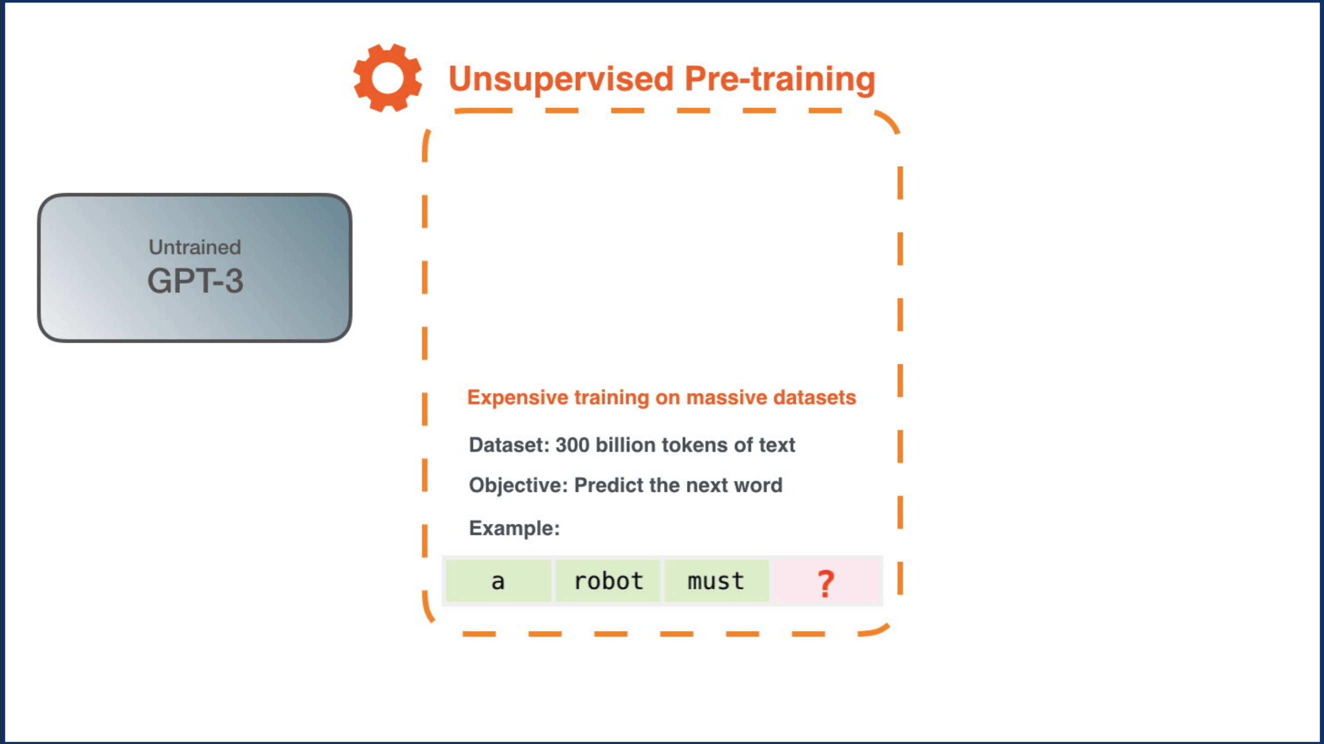




https://multimae.epfl.ch/

NOKIA BELL LABS

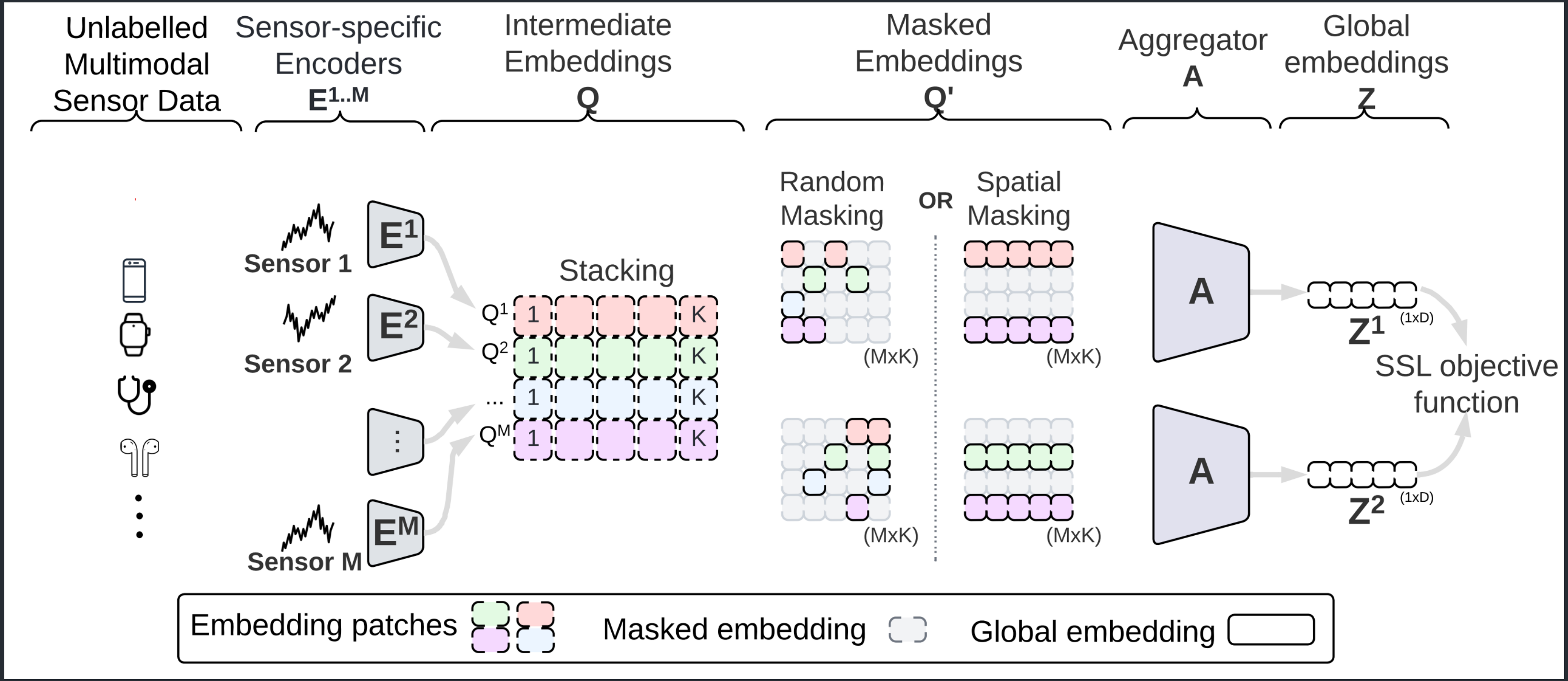
Masking has been wildly successful in training (Chat)GPT



http://jalammar.github.io/how-gpt3-works-visualizations-animations/

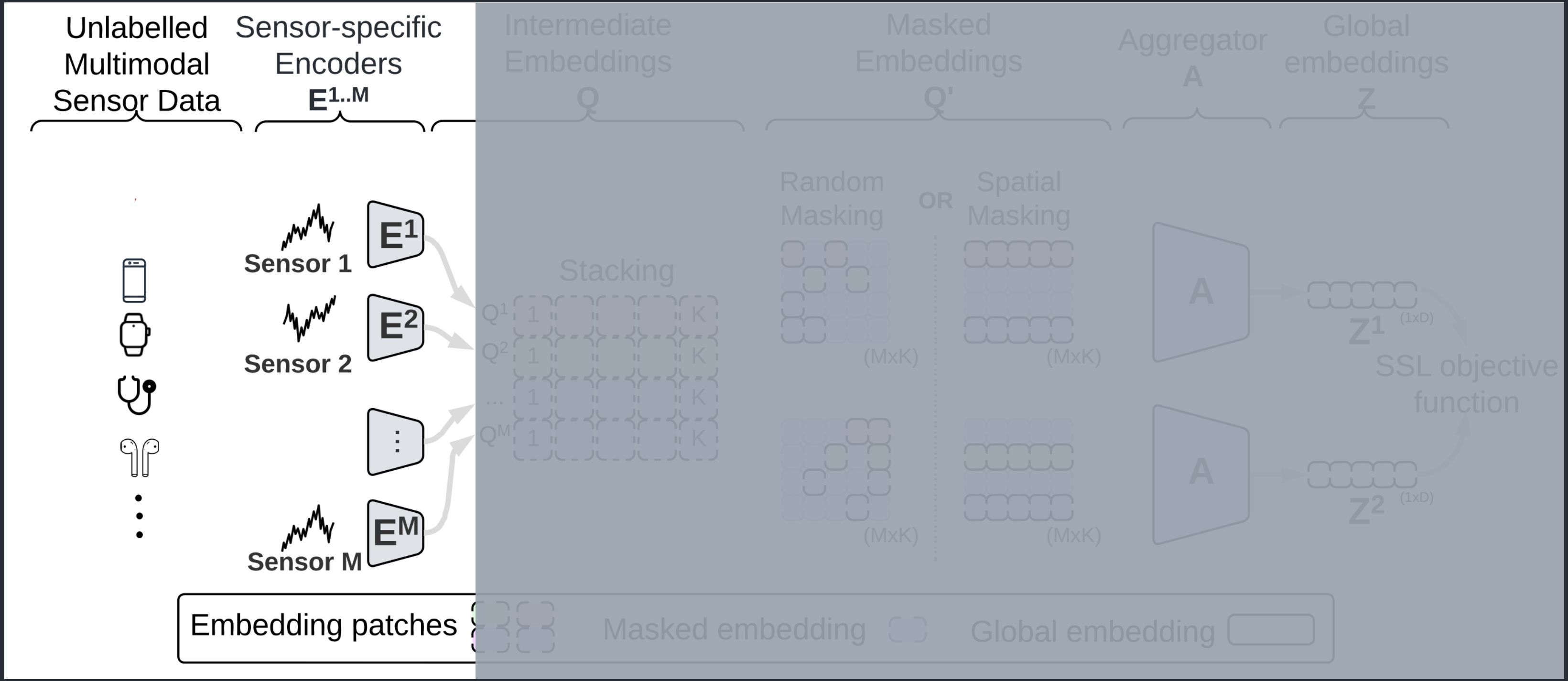
NOKIA BELL LABS

Core idea: contrasting masked latent embeddings



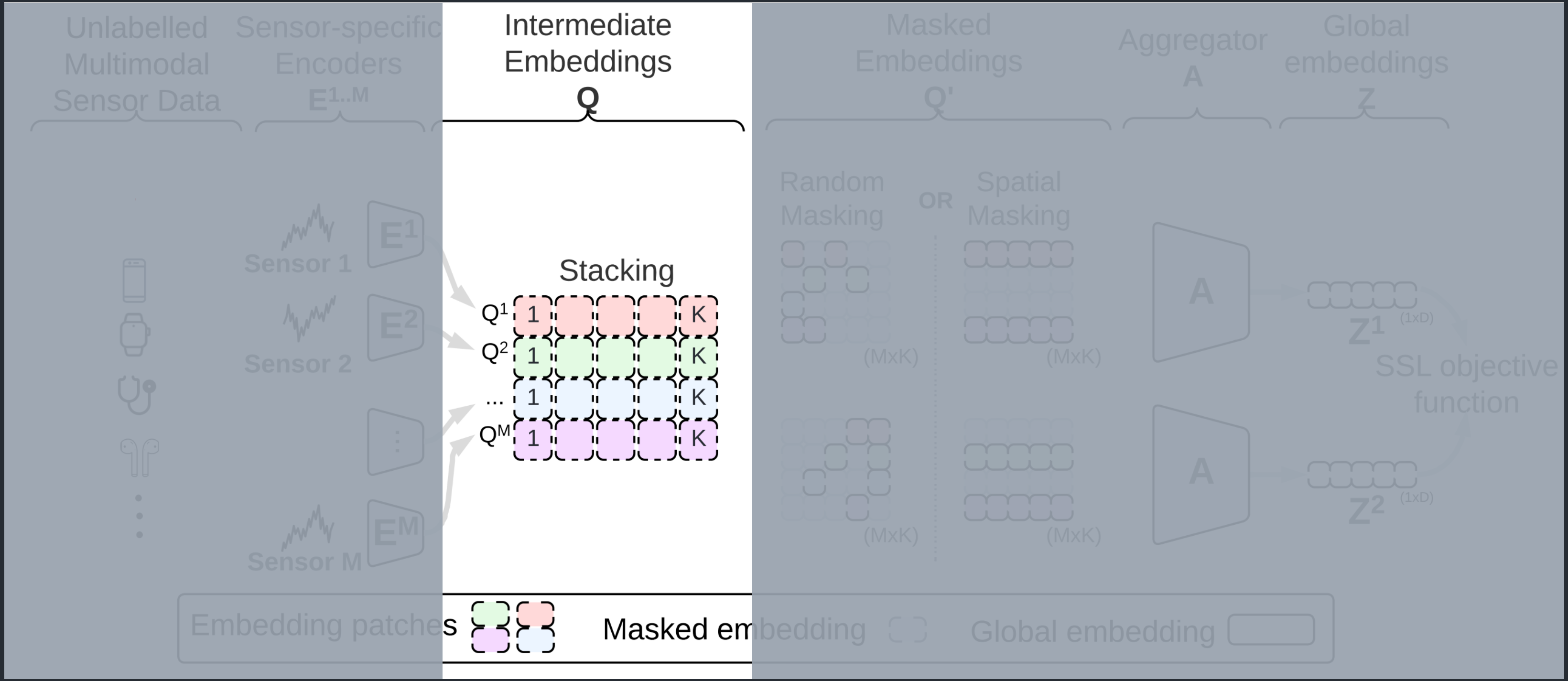
Deldari et al., WSDM 2024 & MLMHD @ ICML 2023

1. Use a separate encoder for each modality/sensor



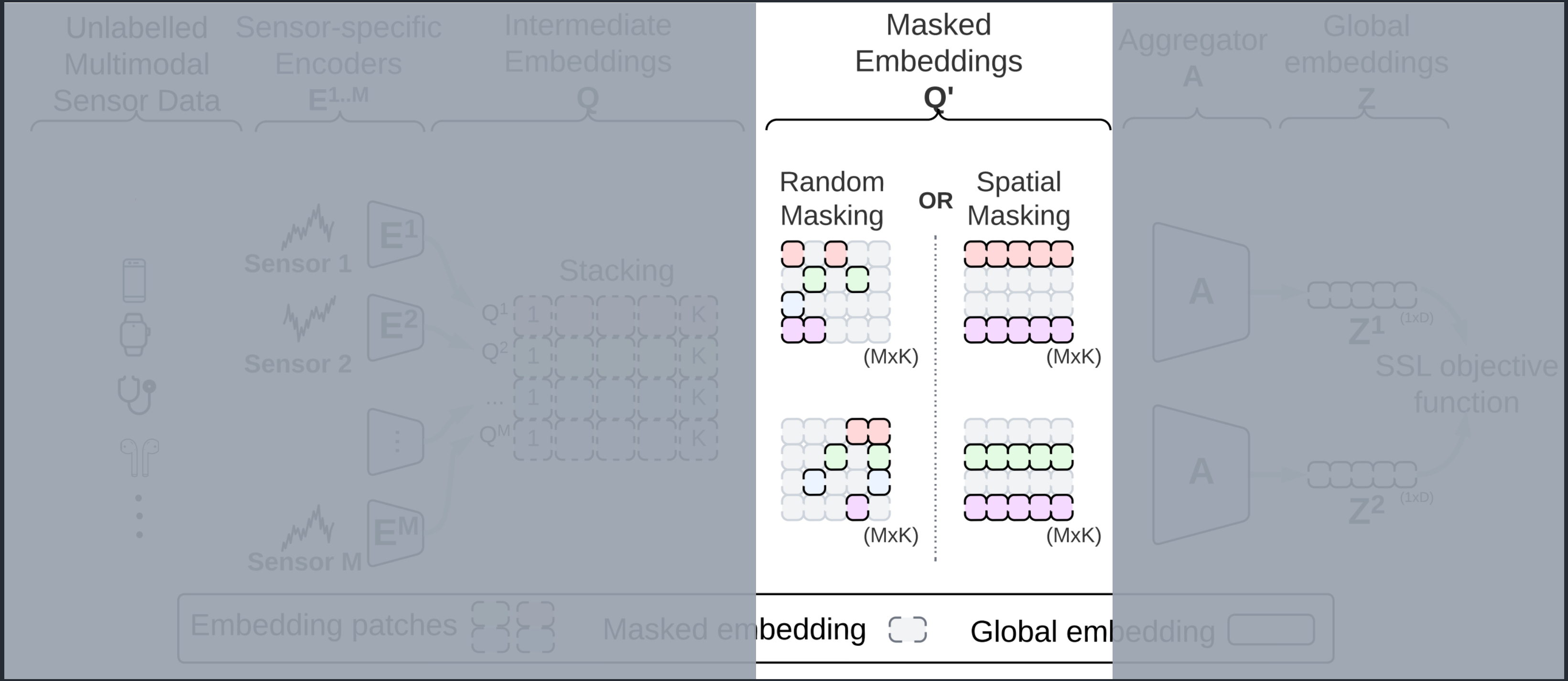
Deldari et al., WSDM 2024 & MLMHD @ ICML 2023

2. Merge all embeddings to a joint representation

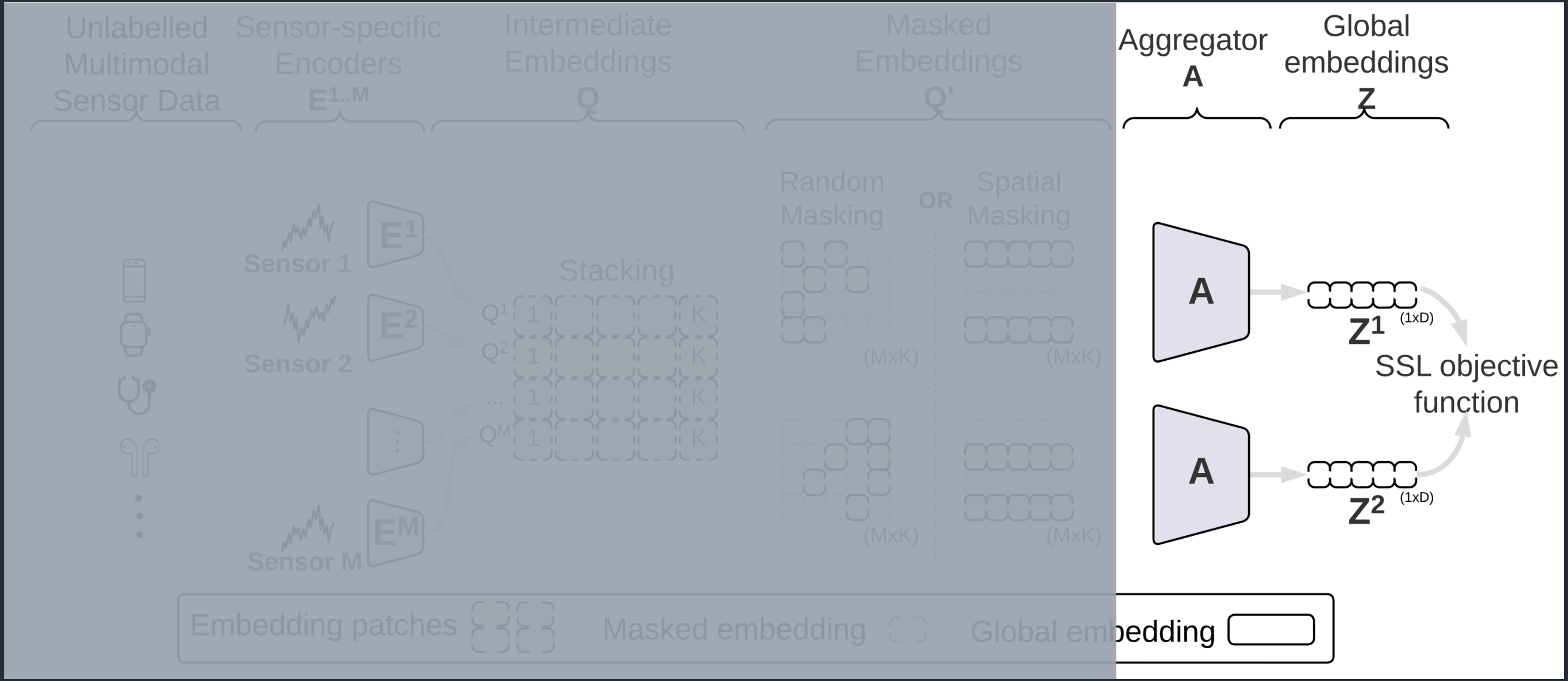


Deldari et al., WSDM 2024 & MLMHD @ ICML 2023

3. Mask the representations in the latent space

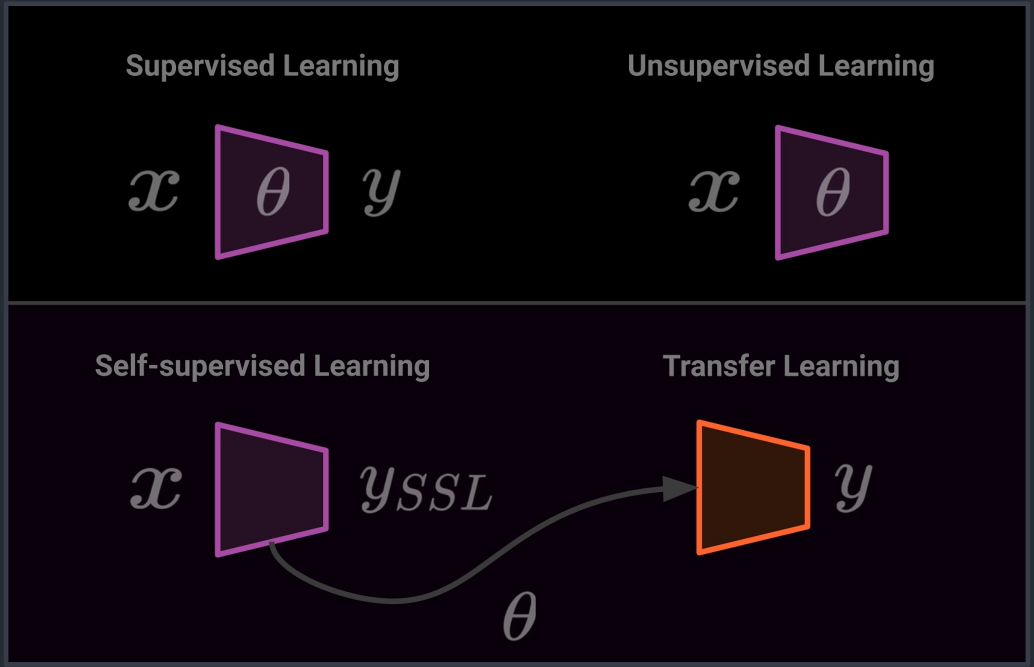


4. Train the network to contrast the two views



Deldari et al., WSDM 2024 & MLMHD @ ICML 2023

5. After pre-training is done, we fine-tune the model with labels



Spathis et al., Cell Patterns 2022

Results

Spatial masking + fine-tuning outperforms other methods

Technique		Dataset		
Training	Method	SleepEDF	PAMAP2	WESAD
End-2-End	Supervised	0.717 (.03)	0.879 (.12)	0.884 (.02)
	DeepConvLSTM	0.601 (.02)	0.718 (.18)	0.791 (.04)
SSL (Fixed enc.)	COCOA	0.628 (.02)	0.839 (.11)	0.669 (.01)
	CroSSL (random)	0.628 (.00)	0.802 (.15)	0.642 (.02)
	CroSSL (spatial)	0.722 (.02)	0.822 (.13)	0.667 (.02)
SSL (Fine-tuned enc.)	COCOA	0.678 (.01)	0.882 (.11)	0.913(.03)
	CroSSL (random)	0.726 (.00)	0.871 (.11)	0.894 (.02)
	CroSSL (spatial)	**0.741 (.00)**	**0.892 (.10)**	**0.939 (.03)**

CroSSL: we test our method in two modes

Masking

- Random
- Spatial

Transfer learning

- Fixed (frozen)
- Fine-tuned (re-training)

CroSSL is robust to missing modalities in prediction time

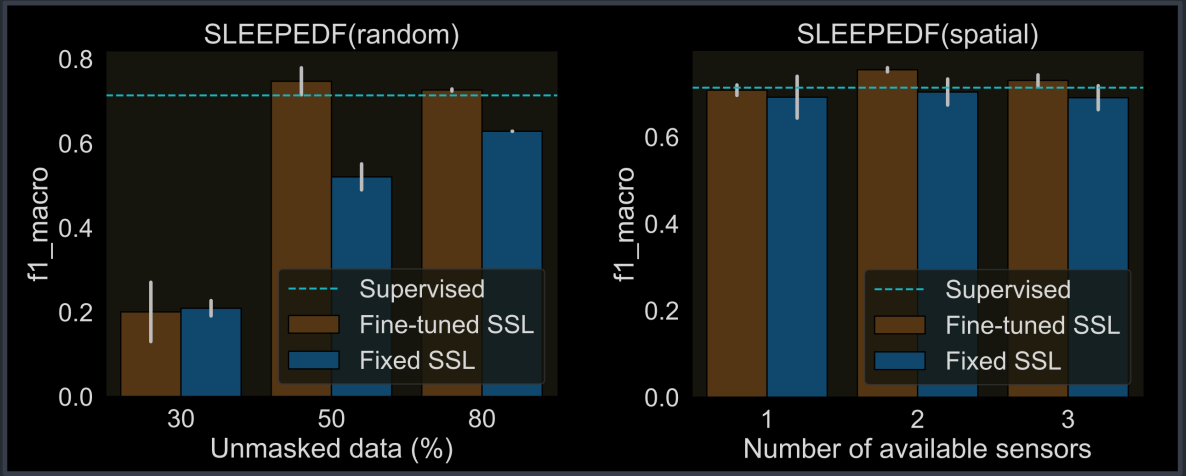
Missing data at: Fine-tuning	Missing data at: Inference	Masking	Method	SleepEDF	PAMAP2	WESAD
No	No	-	Supervised	0.717 (.03)	0.879 (.12)	0.884 (.02)
		random	Fixed SSL	0.628 (.00)	0.709 (.18)	0.629 (.02)
			Fine-tuned SSL	0.726 (.00)	0.825 (.13)	0.890 (.01)
		spatial	Fixed SSL	0.722 (.02)	0.822 (.14)	0.715 (.06)
			Fine-tuned SSL	**0.741 (.00)**	**0.892 (.11)**	**0.925 (.03)**
No	Yes	-	Supervised	0.703 (.03)	0.897 (.11)	0.894 (.02)
		random	Fixed SSL	0.602 (.03)	0.742 (.18)	0.622 (.03)
			Fine-tuned SSL	0.738 (.03)	0.859 (.13)	0.899 (.02)
		spatial	Fixed SSL	0.694 (.01)	0.805 (.16)	0.655 (.02)
			Fine-tuned SSL	0.739 (.02)	**0.899 (.09)**	**0.923 (.03)**
Yes	Yes	-	Supervised	0.202 (.17)	0.469 (.36)	0.304 (.37)
		random	Fixed SSL	0.206 (.35)	0.331 (.19)	0.186 (.16)
			Fine-tuned SSL	0.200 (0)	0.440 (.28)	0.139 (.18)
		spatial	Fixed SSL	0.667 (.13)	0.646 (.21)	0.278 (.14)
			Fine-tuned SSL	0.581 (.24)	0.495 (.35)	0.234 (.17)

Spatial masking is more robust in missing modalities on inference time

Fixed/base models outperform in data-scarce fine-tuning

while supervised models are heavily impacted by missing data

Spatial > Random, masking more effective in larger datasets

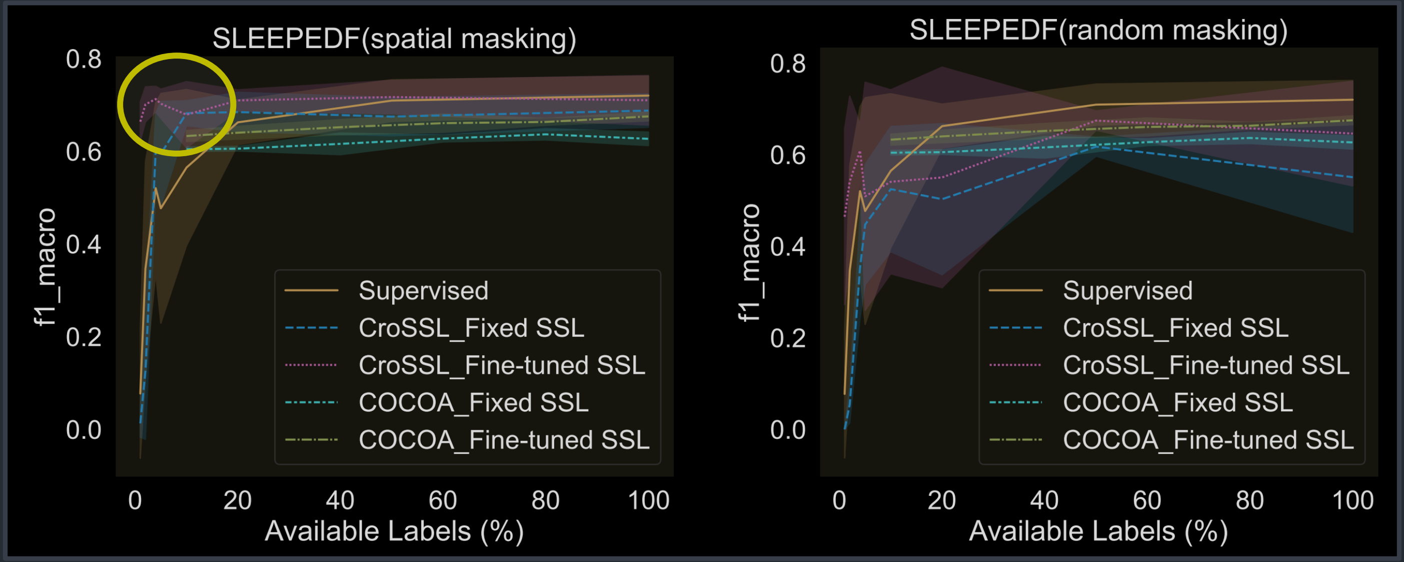


High latent masking ratios do not result in high performance, unlike in vision/MAE papers. Performance drop is more visible in random masking.

Fine-tuned CroSSL outperforms the fixed variant in most cases.

Deldari et al., WSDM 2024 & MLMHD @ ICML 2023

Fine-tuned models are label-efficient, fixed ones need warmup



Fine-tuning is as good as supervised models that have access to labeled data, but it is particularly effective in the low-data regime (1-10% of labels)

Deldari et al., WSDM 2024 & MLMHD @ ICML 2023

NOKIA BELL LABS

Takeaways

1

Achieves state of art performance in multimodal signal ML tasks

2

Handles missing data/sensors in an elegant manner

3

Is data & label-efficient with performance on par or better to supervised models

4

Requires no data pre-processing such as negative pair mining or hiding inputs

Problem solved?

Self-supervision needs large unlabeled data: where to find them?

	PPG	ECG
Number of participants	141,207	106,643
Number of segments	19,854,101	3,743,679
Average number of calendar days per participant	92.54	23.27
Total dataset time span (days)	890	1,240

Apple Heart and Movement Study

Dataset	#Subjects	#Samples	#Classes	Environment	References
UK-Biobank	∼100K	6 B	Unlabelled	Free-living	Doherty et al. (2017)
Capture-24	152	573K	4	Free-living	Willetts et al. (2018)
Rowlands	55	36K	13	Lab	Esliger et al. (2011)
WISDM	46	28K	18	Semi free-living	Weiss et al. (2019)
REALWORLD	14	12K	8	Lab	Sztyler and Stuckenschmidt (2016)
Opportunity	4	3.9K	4	Semi free-living	Roggen et al. (2010)
PAMAP2	8	2.9K	8	Lab	Reiss and Stricker (2012)
ADL	7	0.6K	5	Lab	Bruno et al. (2013)

UK Biobank (wristband) compared to benchmark HAR data

Large unlabeled data of that kind is hard to collect

Not publicly available on the web, unlike images or text

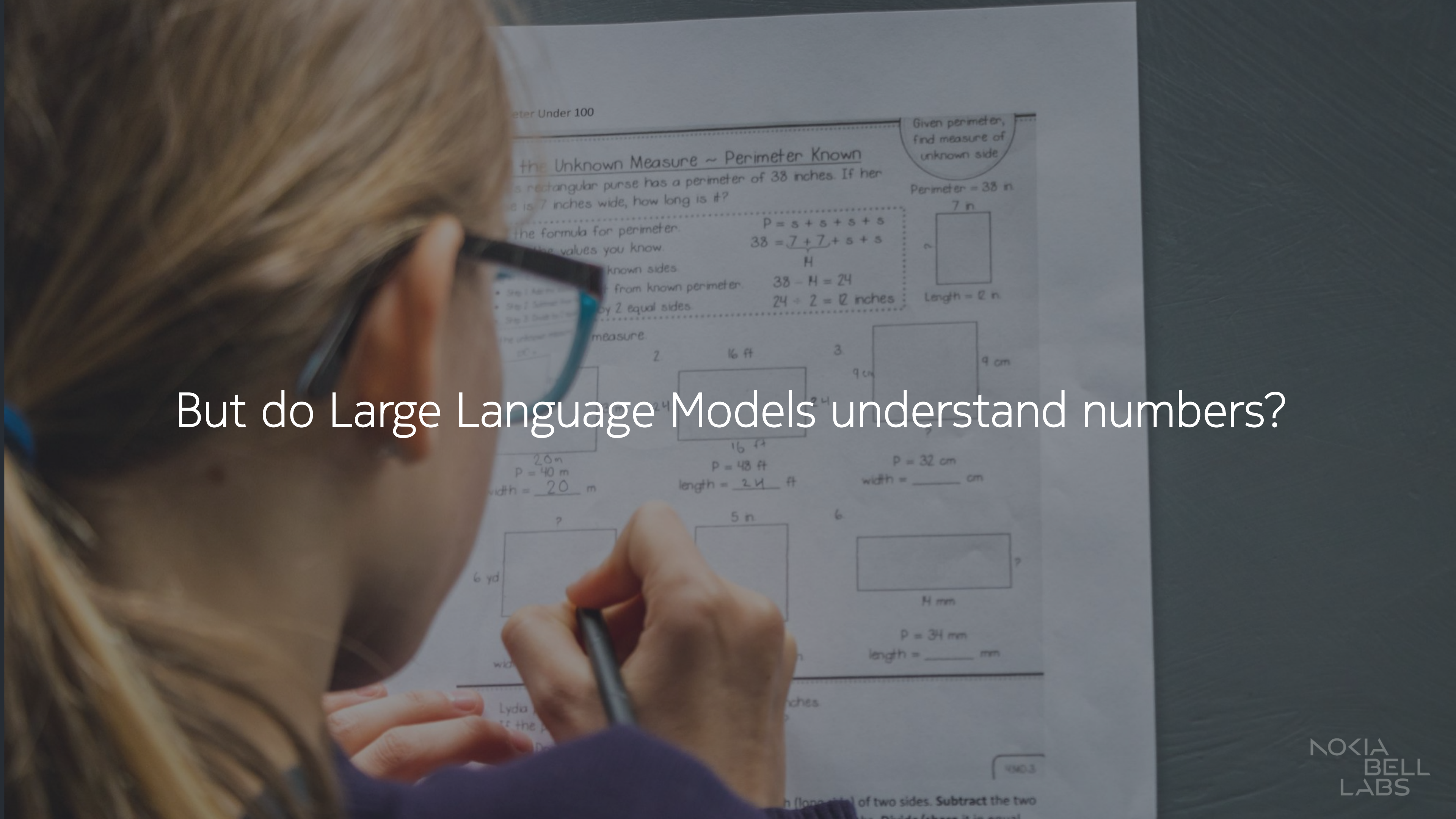
Number of potential modalities hampers progress because it requires aligned/paired data

Available pre-trained models are limited in size and generalization capabilities

https://openreview.net/forum?id=pC3WJHf51j & Yuan et al., ArXiv 2023

From text

to signals



But do Large Language Models understand numbers?

LLM tokenizers are not designed for numbers

Consecutive digit chunking

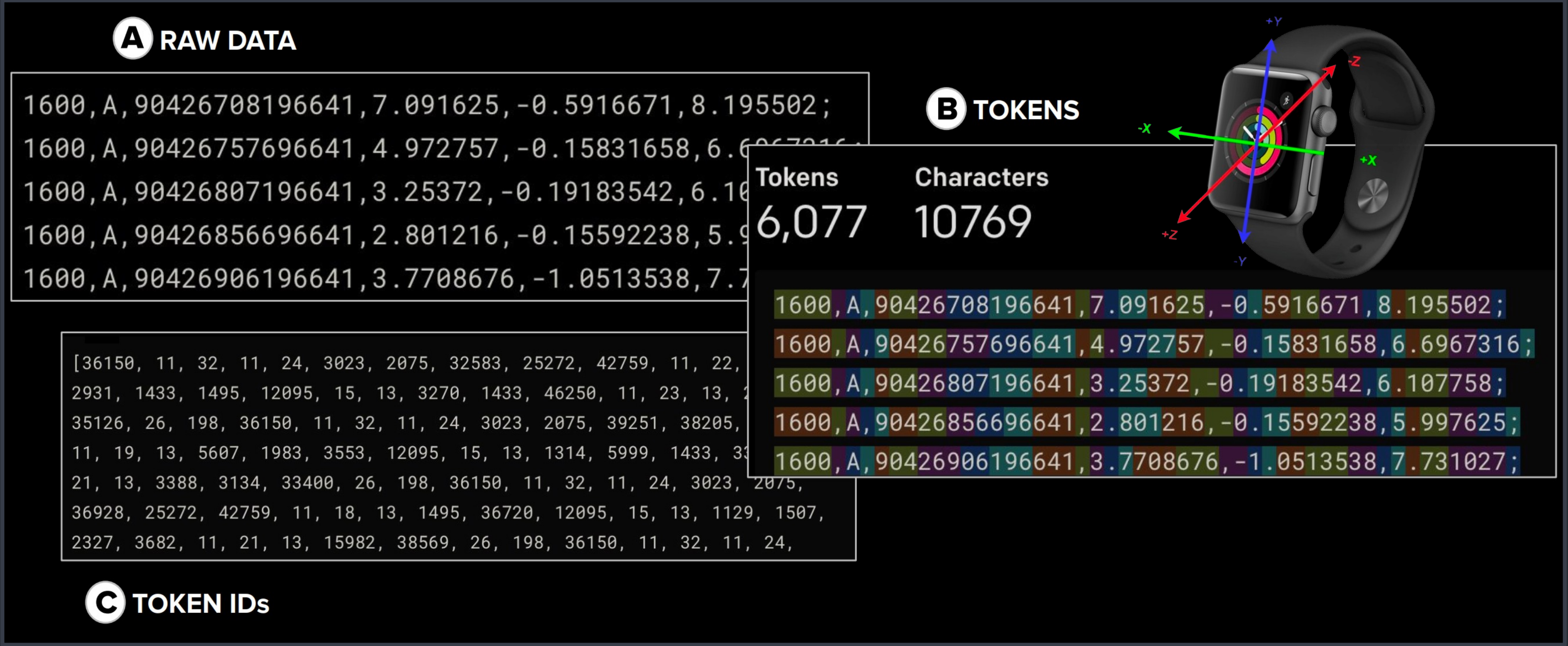
```
Input         → Token IDs
480, 481, 482 → 22148, 11, 4764, 16, 11, 4764, 17
```

Floats

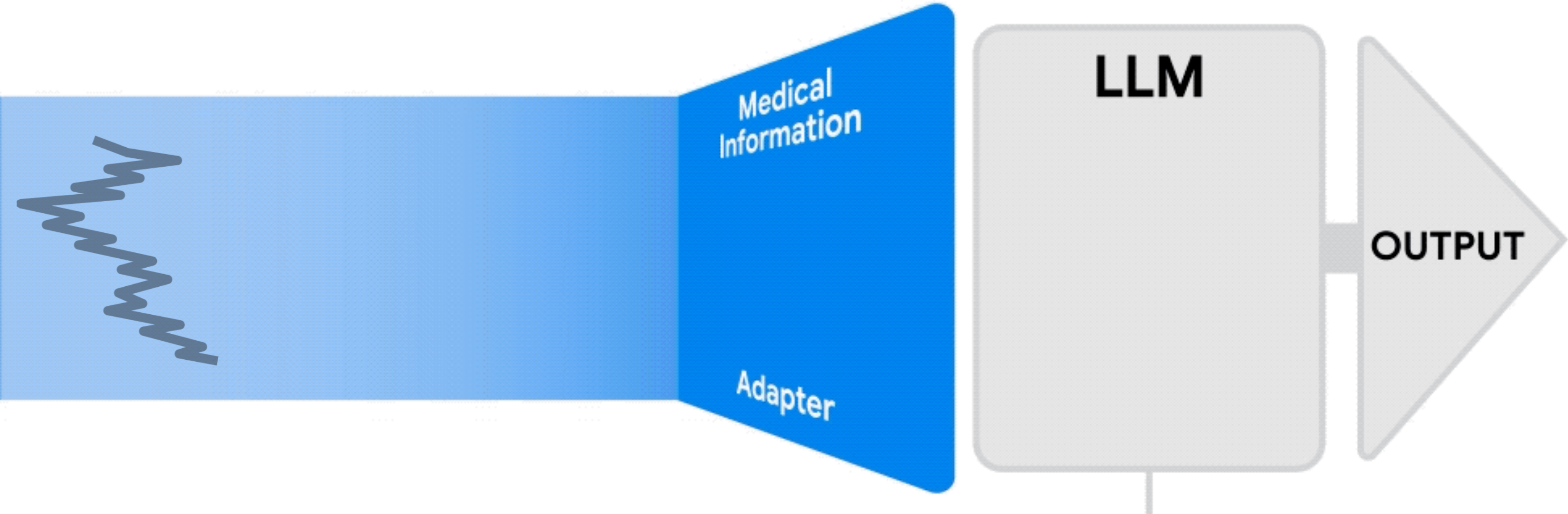
```
Input   → Token IDs
3.14159 → 18, 13, 1415, 19707
```

Case sensitive, trailing whitespaces, arbitrary integer grouping, inconsistent long integer chunking, model-specific behaviours, ...

A case study with activity timeseries data and the GPT tokenizer



Bridging the modality gap with adapters & prompt-tuning



https://blog.research.google/2023/08/multimodal-medical-ai.html

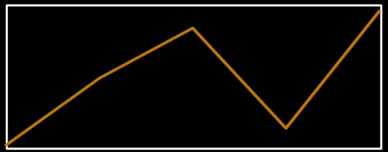
Prompting with **numbers** in addition to **text**

Activity Recognition

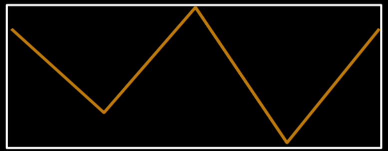
Prompt:

Classify the accelerometer data in meters per second squared as either walking or running.

Response:



Walking.



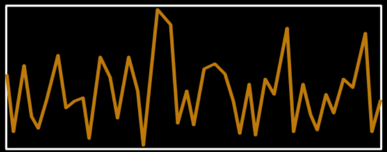
Running.

Atrial Fibrillation Classification

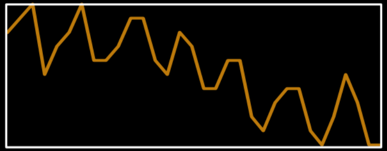
Prompt:

Classify the given Interbeat Interval sequence in ms as either Atrial Fibrillation or Normal Sinus.

Response:



Atrial Fibrillation.



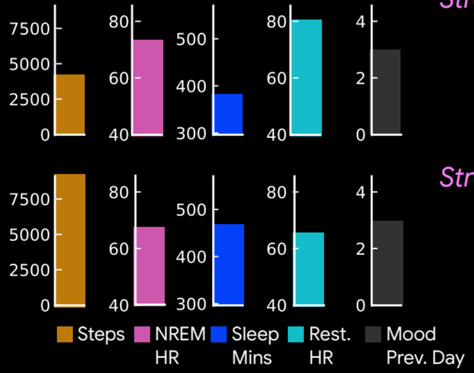
Normal Sinus.

Stress

Prompt:

Steps [Steps] , resting heart rate: [RHR] beats/min, sleep duration: [SleepMinutes] mins, non-rem heart rate: [NREMHR] beats/min, mood last day [Mood] out of 5. What will my stress level be?

Response:

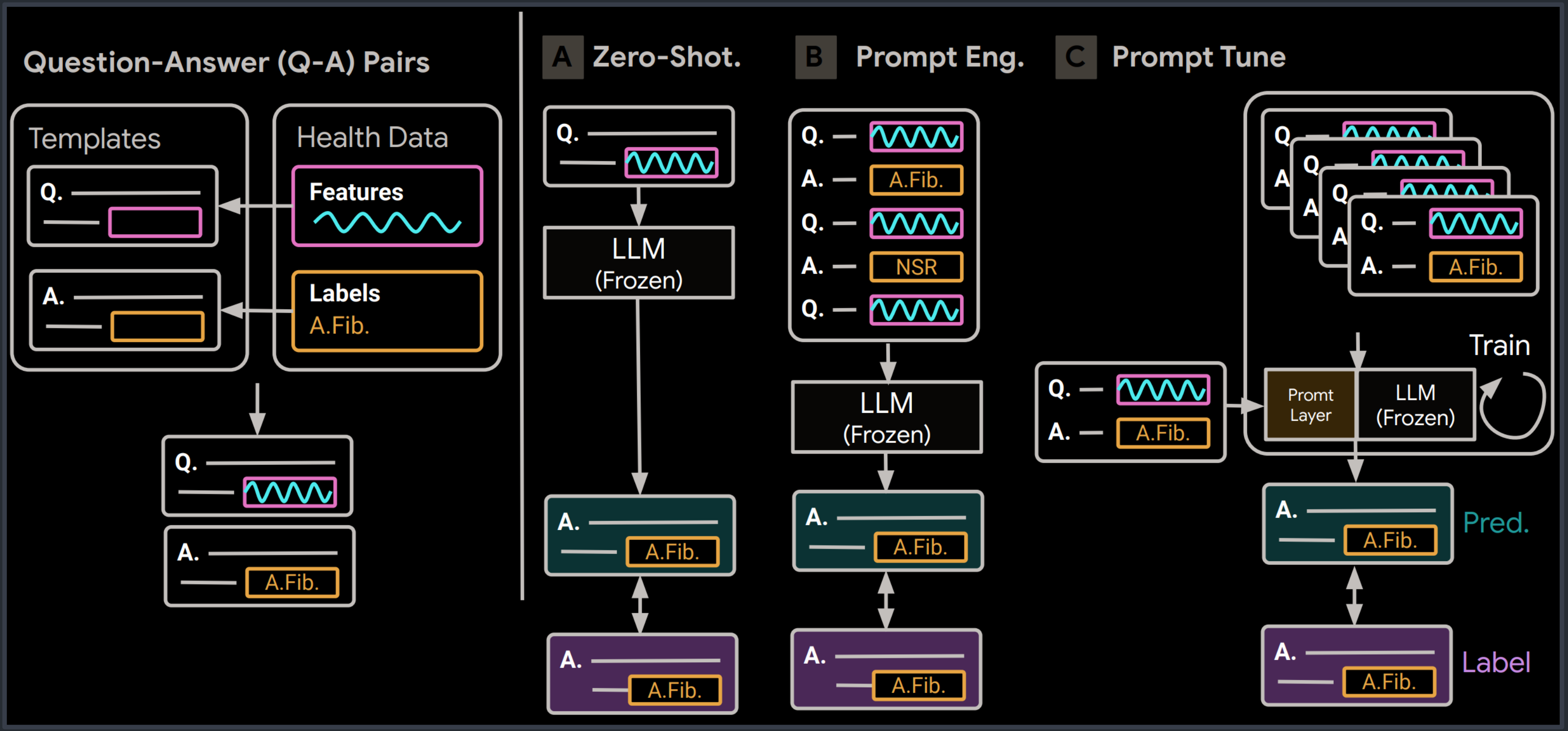


Stress: 5 out of 5.

Stress: 1 out of 5.

NOKIA BELL LABS

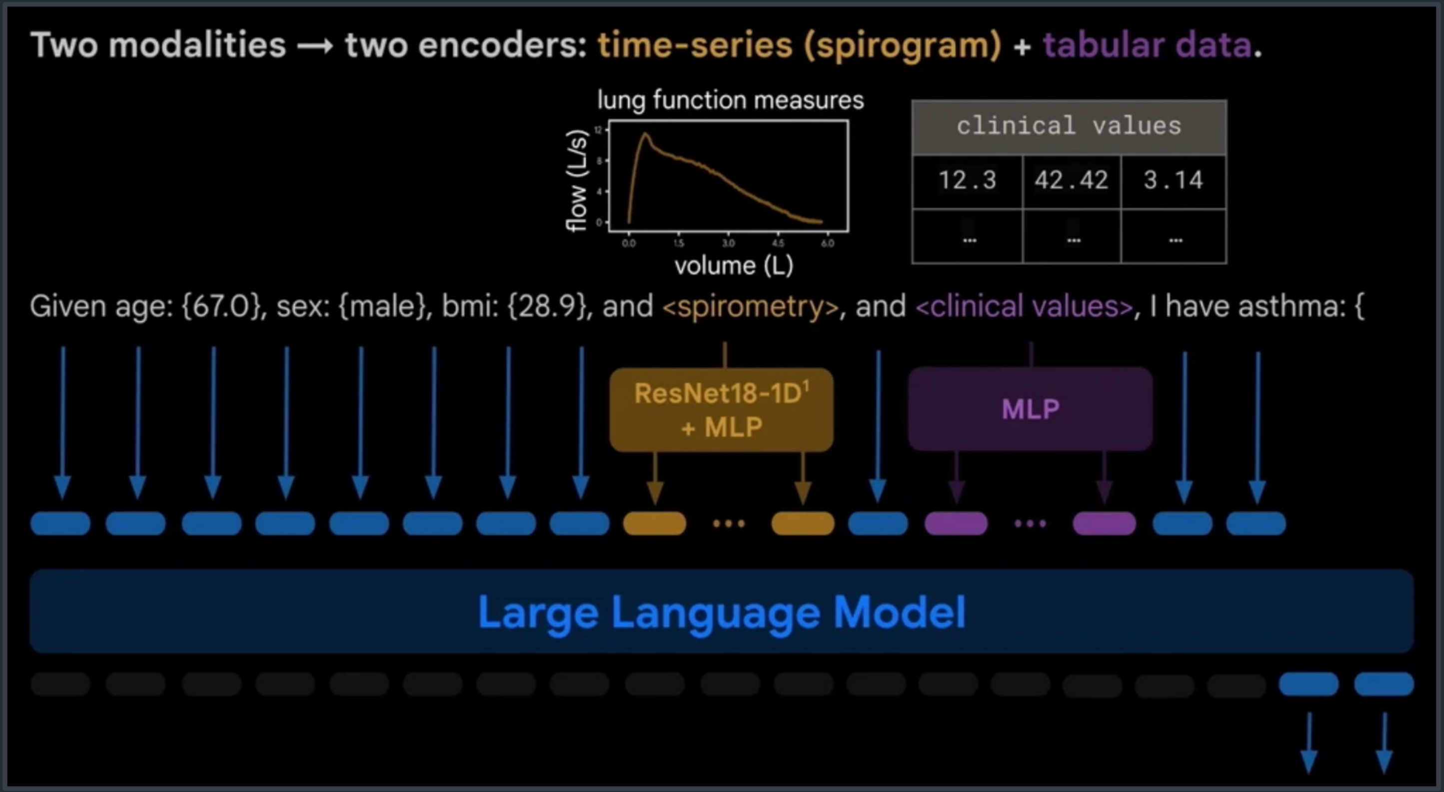
From prompt engineering to few-shot learning to prompt-tuning



Liu et al., ArXiv 2023

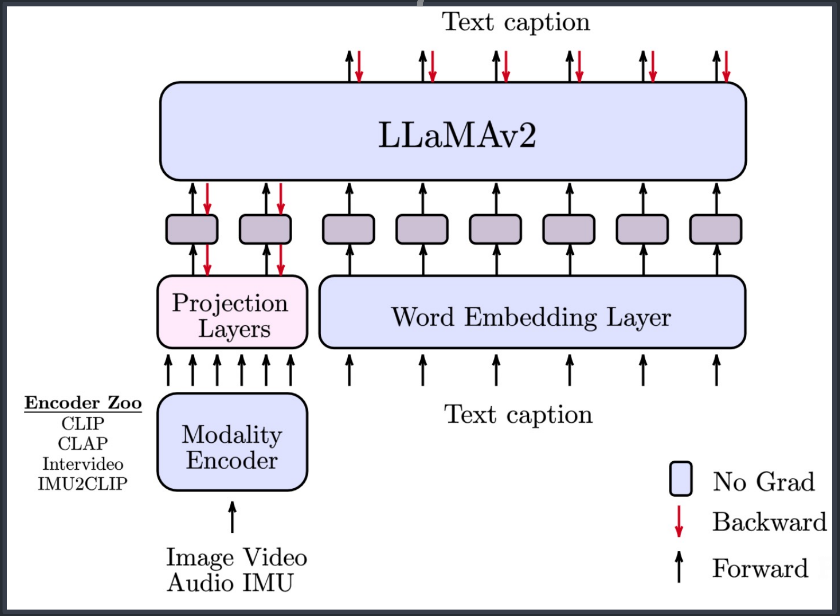
NOKIA BELL LABS

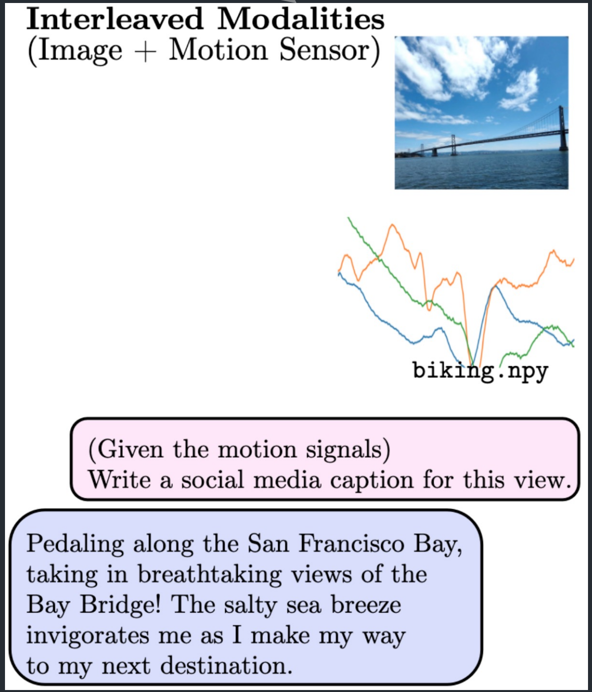
Instead of prompt-tuning, first (auto)encode the numerical data



Belyaeva et al, Arxiv 2023

Everything-to-everything multimodal models





Moon et al, Arxiv 2023

Pros

Computationally efficient

LLM is fixed/frozen

Allows connecting to other high-performing models (e.g. a *sota* ECG encoder)

Breaking down the system to encoder + adapter + LLM enables faster iteration and testing

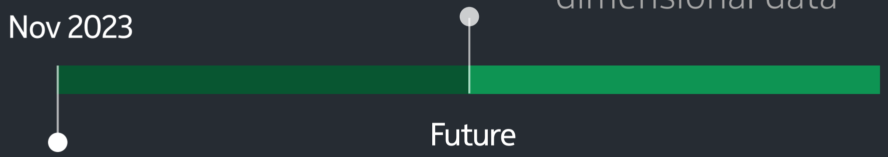
Cons

Modularization introduces complexity, gradients don't propagate all the way

Adapter <-> LLM communication is no longer interpretable (compared to natural language prompts)

Where are we now and what is missing?

- Adapters don’t need elaborate textual prompts
- Multimodal integration through joint embedding spaces
- Improved digit-level tokenizers
- Longer context windows that fit high-dimensional data



Treating LLMs as generic pre-trained models seems to be working (!)

We still have to "ground" them through

- Verbose hand-engineered prompts
- Extensive aggregation/downsampling
- Careful dataset curation

Read more on our papers

Latent Masking for Multimodal Self-supervised Learning in Health Timeseries

Shohreh Deldari [1 2] **Dimitris Spathis** [3] **Mohammad Malekzadeh** [3] **Fahim Kawsar** [3] **Flora Salim** [2] **Akhil Mathur** [3]

Abstract

Limited availability of labeled data for machine learning on biomedical time-series hampers progress in the field. Self-supervised learning (SSL) is a promising approach to learning data representations without labels. However, current SSL methods require expensive computations for negative pairs and are designed for single modalities, limiting their versatility. To overcome these limitations, we introduce `CroSSL` (Cross-modal SSL). `CroSSL` introduces two novel concepts: masking intermediate embeddings from modality-specific encoders and aggregating them into a global embedding using a cross-modal aggregator. This enables the handling of missing modali-

applications in healthcare, including human activity recognition (HAR) and sleep tracking through brain activity monitoring (Kemp et al., 2000; Tang et al., 2021). However, the reliance on labeled data for training deep neural networks (DNNs) has hindered their scalability (Yuan et al., 2022). Collecting, annotating, and maintaining large labeled datasets can be expensive, time-consuming, and impractical, leading to a growing interest in self-supervised learning (SSL) that learns from unlabeled data (Saeed et al., 2019).

SSL defines an artificial task, known as a pretext task, where the supervisory signal is automatically generated from unlabelled data, enabling the training of an encoder model to learn a latent representation of the input data (Yuan et al., 2022). SSL has shown promise in various applications, such as HAR (Tang et al., 2021), by leveraging large amounts of

Deldari et al, WSDM'24 & ML4MHD @ ICML'23

arxiv.org/abs/2307.16847

***The first step is the hardest*: Pitfalls of Representing and Tokenizing Temporal Data for Large Language Models**

Dimitris Spathis *
Nokia Bell Labs
Cambridge, UK

Fahim Kawsar
Nokia Bell Labs
Cambridge, UK

ABSTRACT

Large Language Models (LLMs) have demonstrated remarkable generalization across diverse tasks, leading individuals to increasingly use them as personal assistants and universal computing engines. Nevertheless, a notable obstacle emerges when feeding numerical/temporal data into these models, such as data sourced from wearables or electronic health records. LLMs employ tokenizers in their input that break down text into smaller units. However, tokenizers are *not* designed to represent numerical values and might struggle to understand repetitive patterns and context, treating consecutive values as separate tokens and disregarding their temporal relationships. Here, we discuss recent works that employ LLMs for human-centric tasks such as in mobile health sensing and present a case study showing that popular LLMs tokenize temporal data incorrectly. To address that, we highlight potential solutions such as prompt tuning with lightweight embedding layers as well as multimodal adapters, that can help bridge this "modality gap". While the capability of language models to generalize to other modalities with minimal or no finetuning is exciting, this paper underscores the unintentional fragmentation of continuous sequences into disjointed tokens. Consequently, the temporal relationships that underpin such data may be lost in translation, potentially undermining the very essence of the information being processed.

In this context, this paper delves into the nuances and obstacles that emerge when LLMs are confronted with the task of representing and tokenizing temporal data. We focus on the interplay between numerical and textual information, uncovering the potential pitfalls that can hamper the effective utilization of LLMs in scenarios where temporal context is important. Last, we discuss potential solutions from the rapidly growing area of parameter-efficient transfer learning and multimodal adapters that could enable better integration of non-textual data into LLMs.

2 TOKENIZATION IN LANGUAGE MODELS

Tokenization is a fundamental process underpinning the operation of LLMs. It involves the division of input and output texts into smaller, manageable units known as tokens. These tokens serve

Spathis & Kawsar, GenAI UbiComp'23

arxiv.org/abs/2309.06236

NOKIA BELL LABS

Dimitris Spathis

Sr Researcher
Nokia Bell Labs
Cambridge, UK

dispathis.com

Q&A

Hiring PhD interns!

Microsoft AI & Pizza talk • November 2023