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?
Multimodal data → structurally different signals

- Face masks (breathing pattern, airborne pathogens, inflammation markers)
- Smart textiles (skin temperature, metabolites)
- Electronic epidermal tattoos (stress biomarkers, for example cortisol)
- Smartwatches (activity, sleep, resting heart rate, blood O₂ levels)
- Smart lenses (intraocular pressure)
- On-tooth sensors (drugs, for example antivirals and antibiotics)
- Smart patches (electrocardiogram)
- Microneedle patches (metabolites, for example glucose, lactate, inflammation markers, for example C-reactive protein, drugs)
- Wristbands (electrolytes, metabolites, skin temperature)
- Smart rings (blood pressure)
$0.05 / label

easy

DOG
0.05/label

easy

DOG

hard

❓
$0.05/label \rightarrow >80x \rightarrow 4.00+/label \quad \text{*assuming a sleep technician charging $50/h and 90-120 sleep stage transitions per 8 hours of sleep} \quad \text{easy} \quad \text{DOG} \quad \text{hard} \quad \text{SENSOR-BASED SLEEP TRACKING} \quad \text{IMAGE RECOGNITION}
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.
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.
The first models were based on heuristic (pretext) tasks.
We can even create pretext tasks **across different modalities**

1. Pretrain with **self-supervision**
2. Extract representations at the **window level**
   - ~120,000 windows
3. Summarize at the **individual level**
   - ~1,700 participants
4. Predict **demographics** and clinically relevant information
   - BMI, Fitness, Sex, Age, Energy Expenditure, ...
Self-supervised learning enables transferrable models
And it comes in **different variants**
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
Masked Autoencoders offer a simpler architecture based on masking.
Masking has been wildly **successful** in training (Chat)GPT
Core idea: contrasting masked latent embeddings
1. Use a separate encoder for each modality/sensor
2. Merge all embeddings to a joint representation
3. Mask the representations in the latent space
4. Train the network to **contrast** the two views
5. After pre-training is done, we fine-tune the model with labels.
Results
Spatial masking + fine-tuning outperforms other methods

<table>
<thead>
<tr>
<th>Training</th>
<th>Technique</th>
<th>Method</th>
<th>SleepEDF</th>
<th>Dataset</th>
<th>Dataset</th>
<th>Dataset</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>PAMAP2</td>
<td>WESAD</td>
<td></td>
</tr>
<tr>
<td>End-2-End</td>
<td>Supervised</td>
<td>DeepConvLSTM</td>
<td>0.717 (.03)</td>
<td>0.879 (.12)</td>
<td>0.884 (.02)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>0.601 (.02)</td>
<td>0.718 (.18)</td>
<td>0.791 (.04)</td>
<td></td>
</tr>
<tr>
<td>SSL (Fixed enc.)</td>
<td>COCOA</td>
<td>CroSSL (random)</td>
<td>0.628 (.02)</td>
<td>0.839 (.11)</td>
<td>0.669 (.01)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>CroSSL (spatial)</td>
<td>0.628 (.00)</td>
<td>0.802 (.15)</td>
<td>0.642 (.02)</td>
<td></td>
</tr>
<tr>
<td>SSL (Fine-tuned enc.)</td>
<td>COCOA</td>
<td>CroSSL (random)</td>
<td>0.722 (.02)</td>
<td>0.822 (.13)</td>
<td>0.667 (.02)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>CroSSL (spatial)</td>
<td>0.741 (.00)</td>
<td>0.892 (.10)</td>
<td>0.939 (.03)</td>
<td></td>
</tr>
</tbody>
</table>

CroSSL: we test our method in two modes

- Random
- Spatial

Transfer learning

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

Deldari et al., WSDM 2024 & MLMHD @ ICML 2023
CroSSL is robust to missing modalities in prediction time

<table>
<thead>
<tr>
<th>Missing data at:</th>
<th>Technique</th>
<th>Method</th>
<th>Dataset</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>SleepEDF</td>
<td>PAMAP2</td>
<td>WESAD</td>
<td></td>
</tr>
<tr>
<td>Fine-tuning</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>No</td>
<td>-</td>
<td>Supervised</td>
<td>0.717 (.03)</td>
<td>0.879 (.12)</td>
<td>0.884 (.02)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>random</td>
<td>Fixed SSL</td>
<td>0.628 (.00)</td>
<td>0.709 (.18)</td>
<td>0.629 (.02)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>spatial</td>
<td>Fine-tuned SSL</td>
<td>0.726 (.00)</td>
<td>0.825 (.13)</td>
<td>0.890 (.01)</td>
</tr>
<tr>
<td></td>
<td>No</td>
<td>-</td>
<td>Fixed SSL</td>
<td>0.722 (.02)</td>
<td>0.822 (.14)</td>
<td>0.715 (.06)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>random</td>
<td>Fine-tuned SSL</td>
<td>0.741 (.00)</td>
<td>0.892 (.11)</td>
<td>0.925 (.03)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>spatial</td>
<td>Fixed SSL</td>
<td>0.739 (.02)</td>
<td>0.899 (.09)</td>
<td>0.923 (.03)</td>
</tr>
<tr>
<td></td>
<td>Yes</td>
<td>-</td>
<td>Supervised</td>
<td>0.703 (.03)</td>
<td>0.897 (.11)</td>
<td>0.894 (.02)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>random</td>
<td>Fixed SSL</td>
<td>0.602 (.03)</td>
<td>0.742 (.18)</td>
<td>0.622 (.03)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>spatial</td>
<td>Fine-tuned SSL</td>
<td>0.738 (.03)</td>
<td>0.859 (.13)</td>
<td>0.899 (.02)</td>
</tr>
<tr>
<td></td>
<td>Yes</td>
<td>-</td>
<td>Fixed SSL</td>
<td>0.694 (.01)</td>
<td>0.805 (.16)</td>
<td>0.655 (.02)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>random</td>
<td>Fine-tuned SSL</td>
<td>0.739 (.02)</td>
<td>0.899 (.09)</td>
<td>0.923 (.03)</td>
</tr>
</tbody>
</table>

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

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

<table>
<thead>
<tr>
<th></th>
<th>PPG</th>
<th>ECG</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of participants</td>
<td>141,207</td>
<td>106,643</td>
</tr>
<tr>
<td>Number of segments</td>
<td>19,854,101</td>
<td>3,743,679</td>
</tr>
<tr>
<td>Average number of calendar days per participant</td>
<td>92.54</td>
<td>23.27</td>
</tr>
<tr>
<td>Total dataset time span (days)</td>
<td>890</td>
<td>1,240</td>
</tr>
</tbody>
</table>

Apple Heart and Movement Study

- 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

### UK Biobank (wristband) compared to benchmark HAR data

<table>
<thead>
<tr>
<th>Dataset</th>
<th>#Subjects</th>
<th>#Samples</th>
<th>#Classes</th>
<th>Environment</th>
<th>References</th>
</tr>
</thead>
<tbody>
<tr>
<td>UK-Biobank</td>
<td>~100K</td>
<td>6 B</td>
<td>Unlabelled</td>
<td>Free-living</td>
<td>Doherty et al. (2017)</td>
</tr>
<tr>
<td>Rowlands</td>
<td>55</td>
<td>36K</td>
<td>13</td>
<td>Lab</td>
<td>Eshliger et al. (2011)</td>
</tr>
<tr>
<td>WISDM</td>
<td>46</td>
<td>28K</td>
<td>18</td>
<td>Semi free-living</td>
<td>Weiss et al. (2019)</td>
</tr>
<tr>
<td>REALWORLD</td>
<td>14</td>
<td>12K</td>
<td>8</td>
<td>Lab</td>
<td>Sztyler and Stuckenschmidt (2016)</td>
</tr>
<tr>
<td>Opportunity</td>
<td>4</td>
<td>3.9K</td>
<td>4</td>
<td>Semi free-living</td>
<td>Roggen et al. (2010)</td>
</tr>
<tr>
<td>PAMAP2</td>
<td>8</td>
<td>2.9K</td>
<td>8</td>
<td>Lab</td>
<td>Reiss and Stricker (2012)</td>
</tr>
<tr>
<td>ADL</td>
<td>7</td>
<td>0.6K</td>
<td>5</td>
<td>Lab</td>
<td>Bruno et al. (2013)</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Input</th>
<th>→ Token IDs</th>
</tr>
</thead>
<tbody>
<tr>
<td>480, 481, 482</td>
<td>22148, 11, 4764, 16, 11, 4764, 17</td>
</tr>
</tbody>
</table>

**Floats**

<table>
<thead>
<tr>
<th>Input</th>
<th>→ Token IDs</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.14159</td>
<td>18, 13, 1415, 19707</td>
</tr>
</tbody>
</table>

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

Spathis & Kawsar, GenAI4PC @ Ubicomp 2023
A case study with activity timeseries data and the GPT tokenizer

Spathis & Kawsar, GenAI4PC @ Ubicomp 2023
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.

Liu et al., ArXiv 2023

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

Liu et al., ArXiv 2023
Instead of prompt-tuning, first (auto)encode the numerical data.
Everything-to-everything **multimodal models**

![Diagram](attachment:image.png)

**Interleaved Modalities**
(Image + Motion Sensor)

(Given the motion signals)
Write a social media caption for this view.

Pedaling along the San Francisco Bay, taking in breathtaking views of the Bay Bridge! The salty sea breeze invigorates me as I make my way to my next destination.

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)

Spathis & Kawsar, GenAI4PC @ Ubicomp 2023
Where are we now and what is missing?

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

- 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

We still have to "ground" them through

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

Spathis & Kawsar, GenAI4PC @ Ubicomp 2023
**Latent Masking for Multimodal Self-supervised Learning in Health Timeseries**

Shohreh Deldari 1,2, Dimitris Spathis 3, Mohammad Malekzadeh 3, Fahim Kawar 3, Flora Salim 3, 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 CrossSL (Cross-modal SSL). CrossSL 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 modality-specific 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 (Saered 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 unlabeled samples.

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

Dimitris Spathis 1,2,3, Fahim Kawar 1,2,3

**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.
Dimitris Spathis
Sr Researcher
Nokia Bell Labs
Cambridge, UK

dispathis.com

Q&A
Hiring PhD interns!

Microsoft AI & Pizza talk • November 2023