Building Neuro Foundation Models with torch_brain

Alexandre Andre
PhD Student, University of Pennsylvania
NerDS Lab

July 14, 2026
NeuroDataReHack 2026 @ Janelia Research Campus

Plan

1. Introduction on Neuro Foundation Model (20 min)
2. Demo
    a. Quick Start: training a model with `torch_brain` (40 min)
    b. Quick Hack (20 min)
3. Wrap-up (10 min)

*What is/would be a **Neuro Foundation** Model to **you**?*

What Are Neuro Foundation Models?

Generalize across **data**

- When/Who
  - Sessions/Time (longitudinal recording)
  - Subjects

What Are Neuro Foundation Models?

Generalize across **data**

- When/Who
  - Sessions/Time (longitudinal recording)
  - Subjects

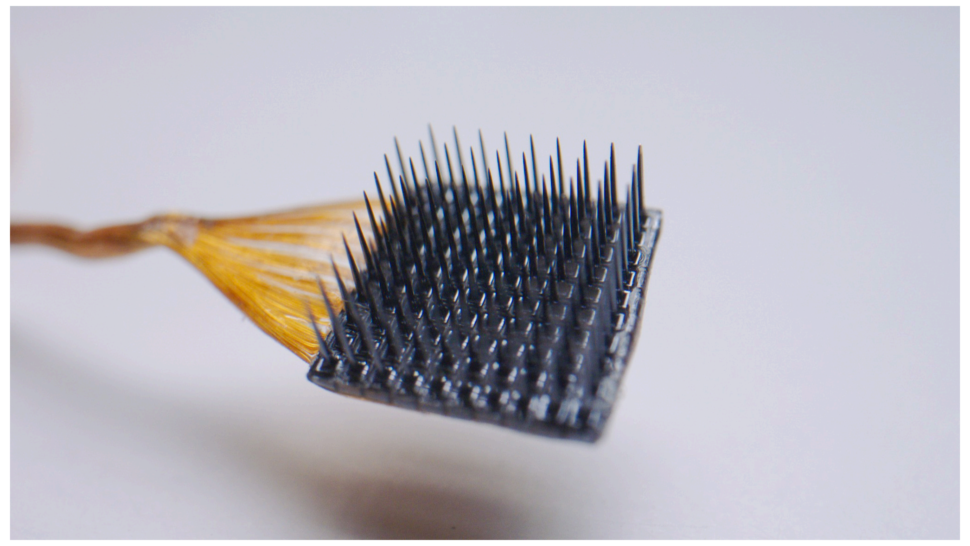
- How/What

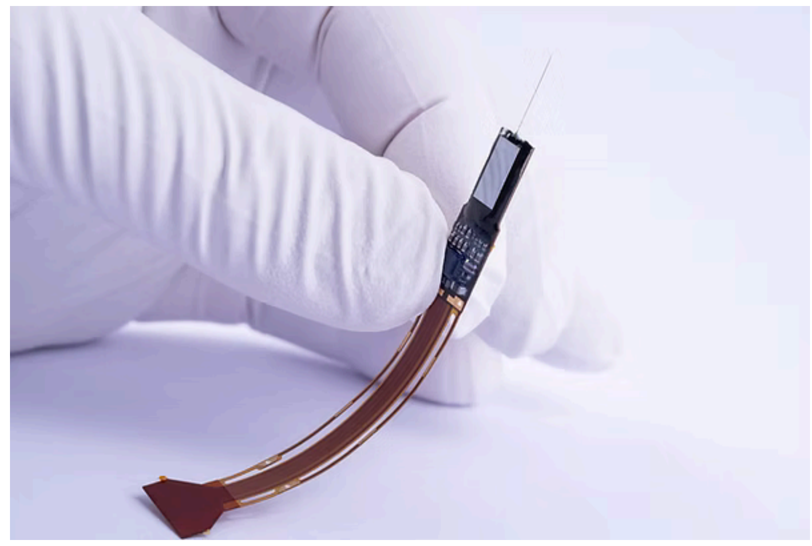
What Are Neuro Foundation Models?

Generalize across **data**

- When/Who
  - Sessions/Time (longitudinal recording)
  - Subjects

- How/What
  - Recording technology



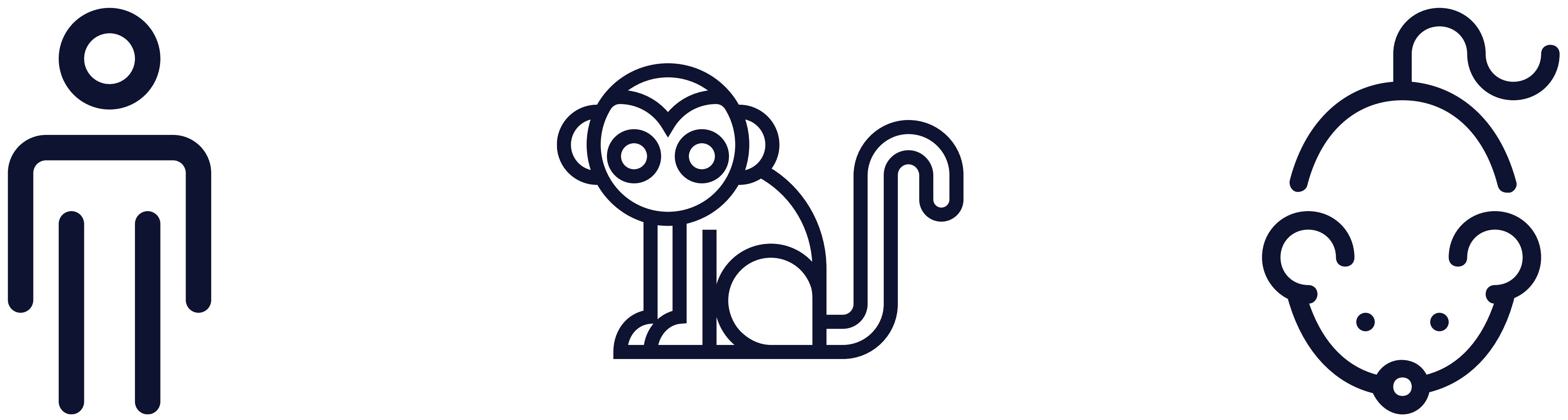


What Are Neuro Foundation Models?

Generalize across **data**

- When/Who
  - Sessions/Time (longitudinal recording)
  - Subjects

- How/What
  - Recording technology
  - Species

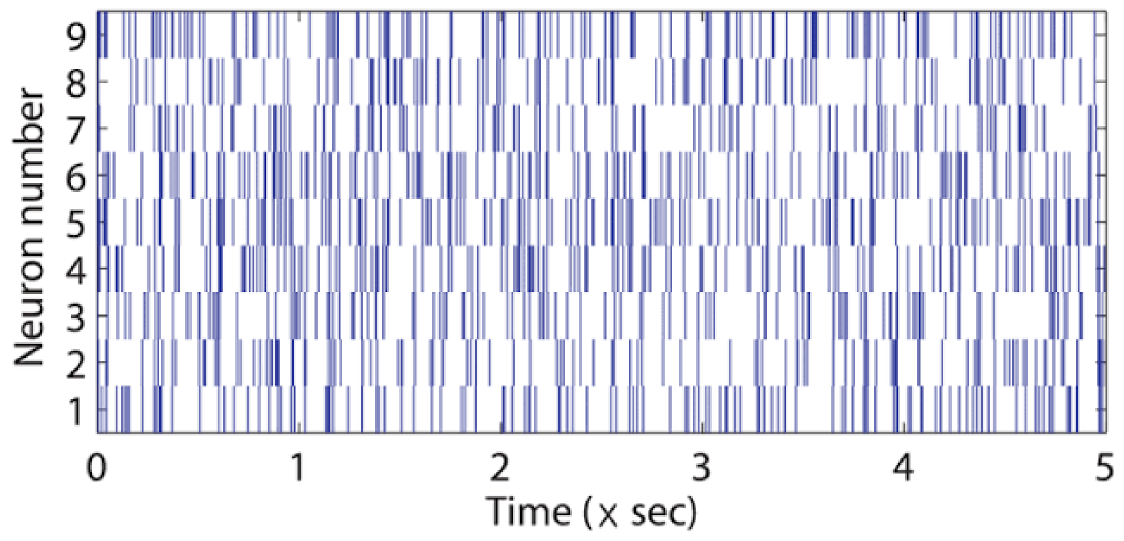


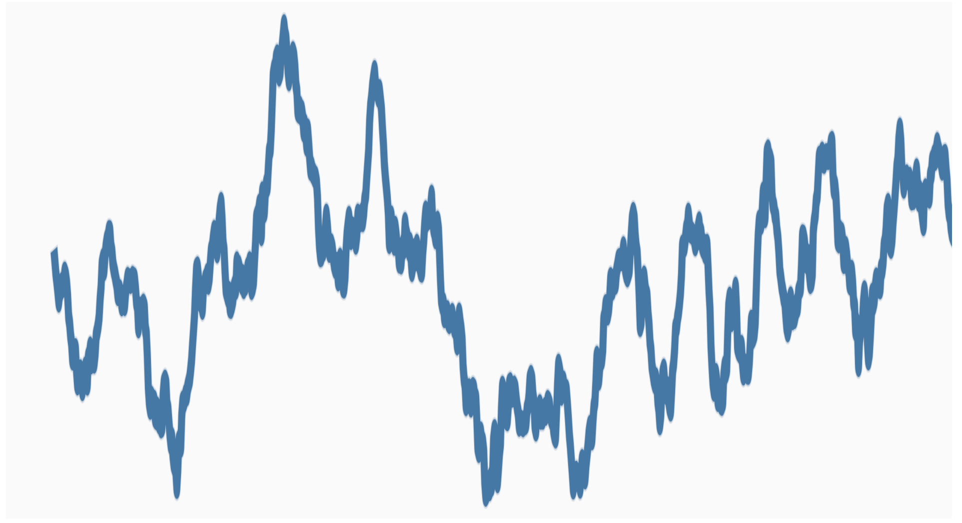
What Are Neuro Foundation Models?

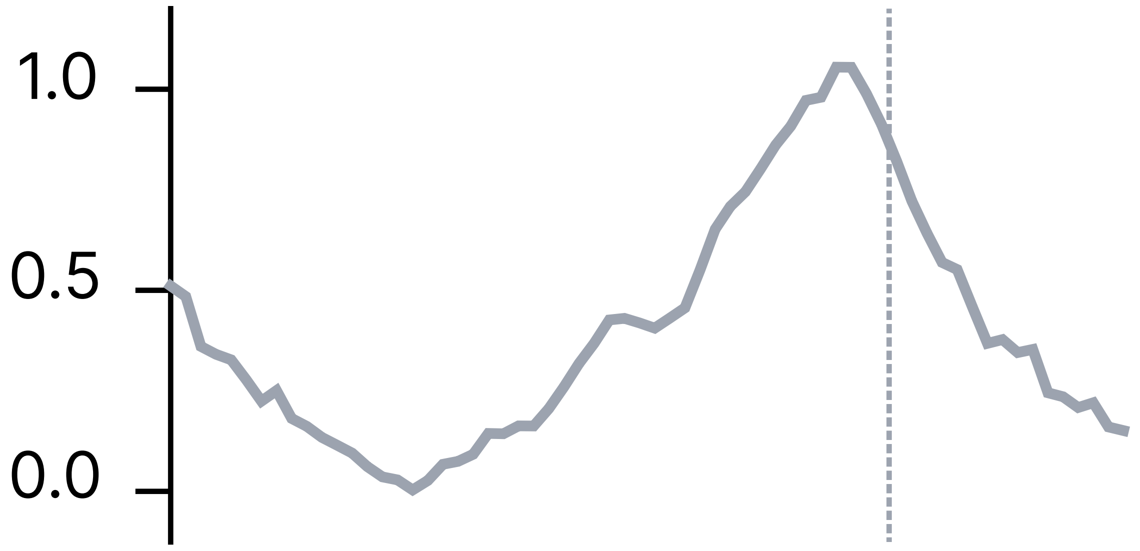
Generalize across **data**

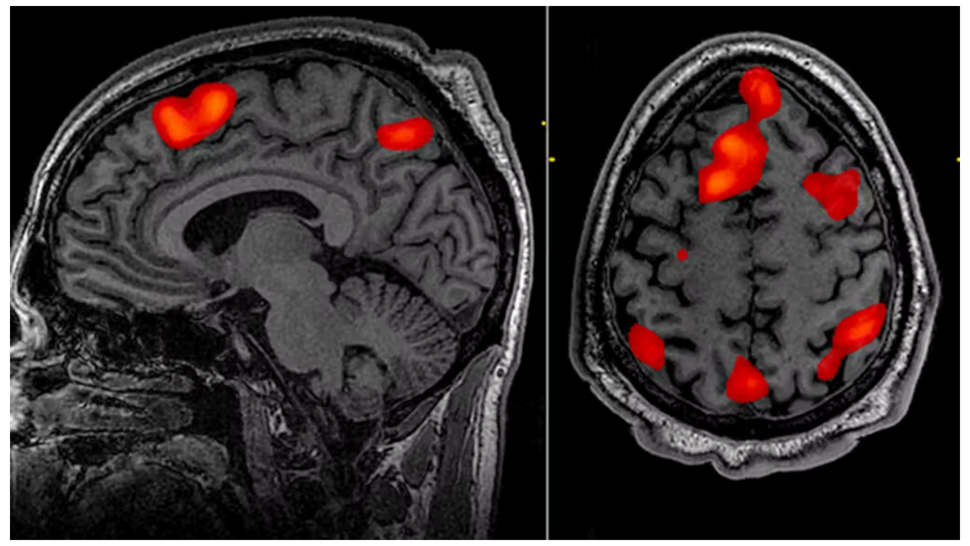
- When/Who
  - Sessions/Time (longitudinal recording)
  - Subjects

- How/What
  - Recording technology
  - Species
  - Modalities









What Are Neuro Foundation Models?

Generalize across **data**

- When/Who
  - Sessions/Time (longitudinal recording)
  - Subjects

- How/What
  - Recording technology
  - Species
  - Modalities

Generalize across **downstream task**

- Data quantity needed to calibrate
  - Fine-tuning
  - Few-shot
  - Zero-shot

What Are Neuro Foundation Models?

Generalize across **data**

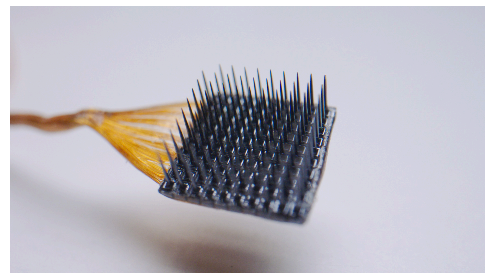
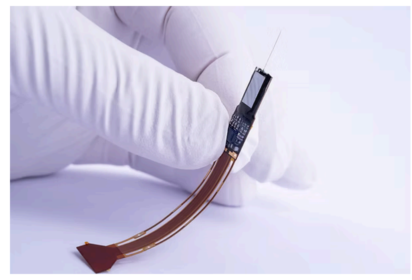
- When/Who
  - Sessions/Time (longitudinal recording)
  - Subjects

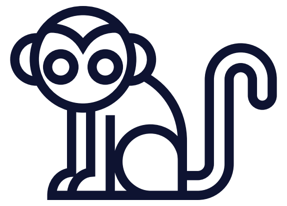
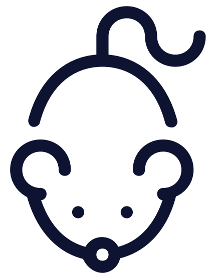
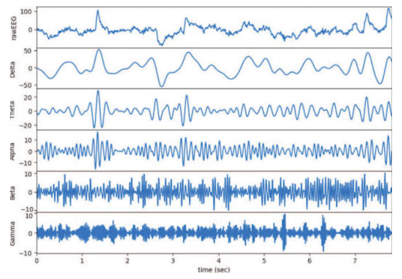
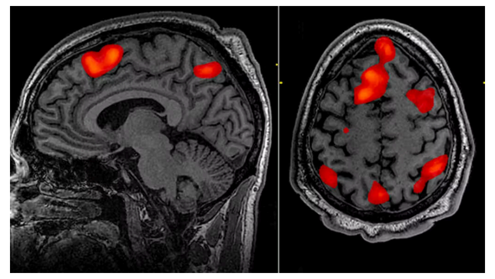
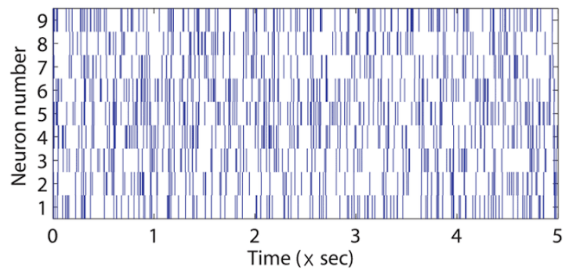
- How/What
  - Recording technology
  - Species
  - Modalities

Generalize across **downstream task**

- Data quantity needed to calibrate
  - Fine-tuning
  - Few-shot
  - Zero-shot



Use Cases for Neuro Foundation Models

Science Discovery

- Digital twins (subject-specific predictive models)
- Hypothesis testing

Use Cases for Neuro Foundation Models

Science Discovery

- Digital twins (subject-specific predictive models)
- Hypothesis testing

Clinical & Biotech Applications

- BCI Speech/Motor (brain-computer interfaces)
- Diagnosis & prediction (e.g. seizure onset)
- Intervention (e.g. neuromodulation/DBS targeting)
- Live probe localization (no histology needed)

Lessons from NLP and Vision

NLP

- Generalizes across language, users, task (summarization, code generation, math reasoning)

Lessons from NLP and Vision

NLP

- Generalizes across language, users, task (summarization, code generation, math reasoning)

Vision

- Generalizes across image domain (medical, natural photos), task (depth estimation, object detection)

Lessons from NLP and Vision

NLP

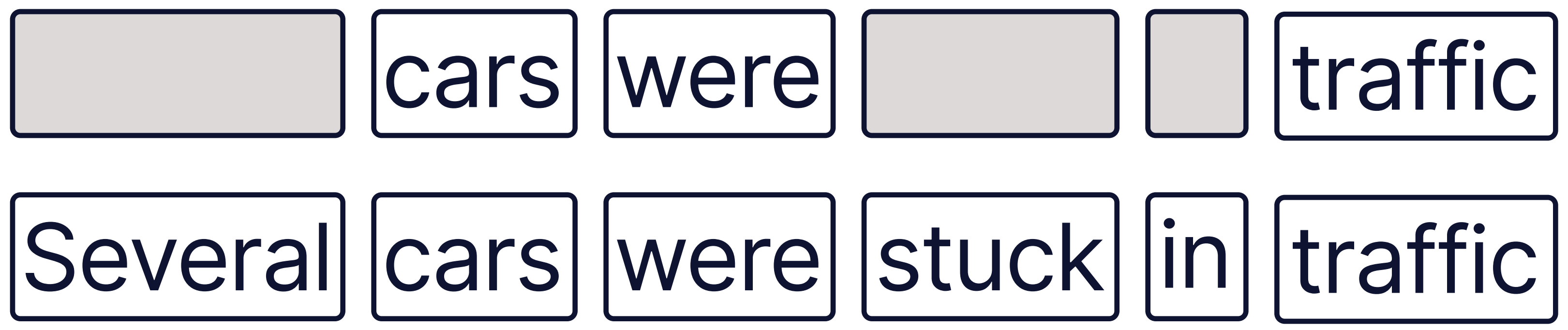
- Generalizes across language, users, task (summarization, code generation, math reasoning)

Vision

- Generalizes across image domain (medical, natural photos), task (depth estimation, object detection)

⇒ Key technique: Pretrain at **scale** with SSL, then adapt to **downstream tasks**

BERT



Lessons from NLP and Vision

NLP

- Generalizes across language, users, task (summarization, code generation, math reasoning)

Vision

- Generalizes across image domain (medical, natural photos), task (depth estimation, object detection)

⇒ Key technique: Pretrain at **scale** with SSL, then adapt to **downstream tasks**

BERT

	cars	were			traffic
Several	cars	were	stuck	in	traffic

GPT

Several	cars		
Several	cars	were	

...

Lessons from NLP and Vision

NLP

- Generalizes across language, users, task (summarization, code generation, math reasoning)

Vision

- Generalizes across image domain (medical, natural photos), task (depth estimation, object detection)

⇒ Key technique: Pretrain at **scale** with SSL, then adapt to **downstream tasks**

BERT

	cars	were			traffic
Several	cars	were	stuck	in	traffic

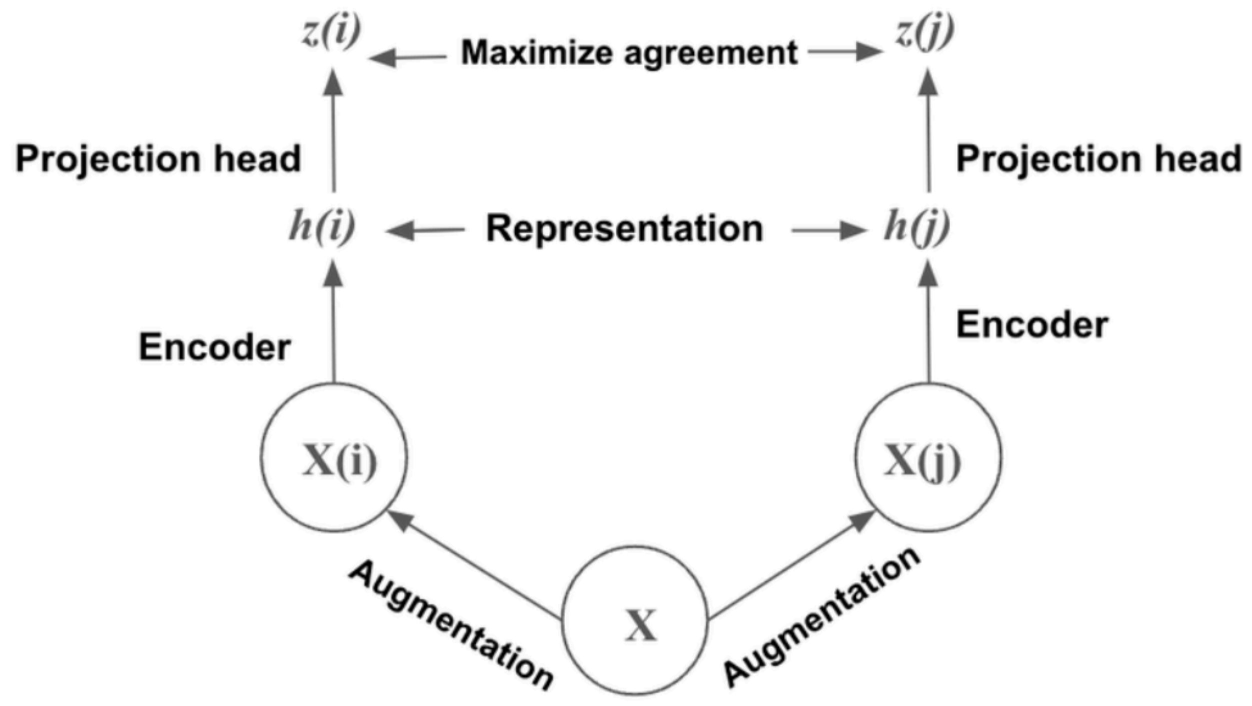
GPT

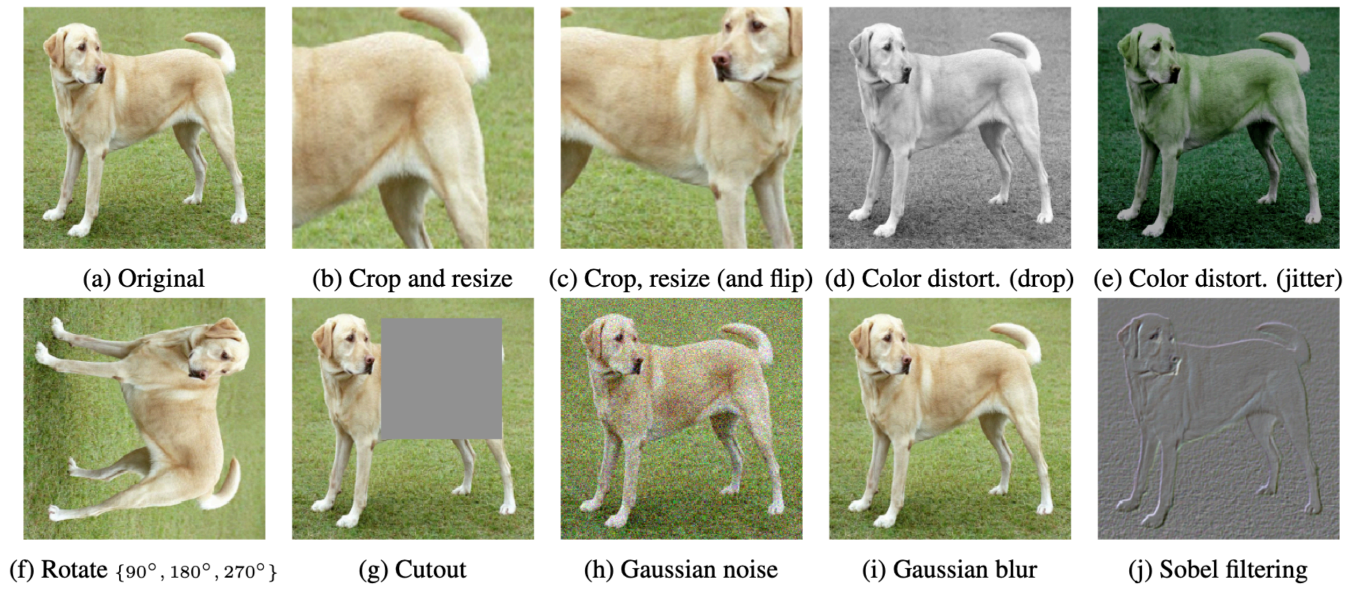
Several	cars	

Several	cars	were	

...

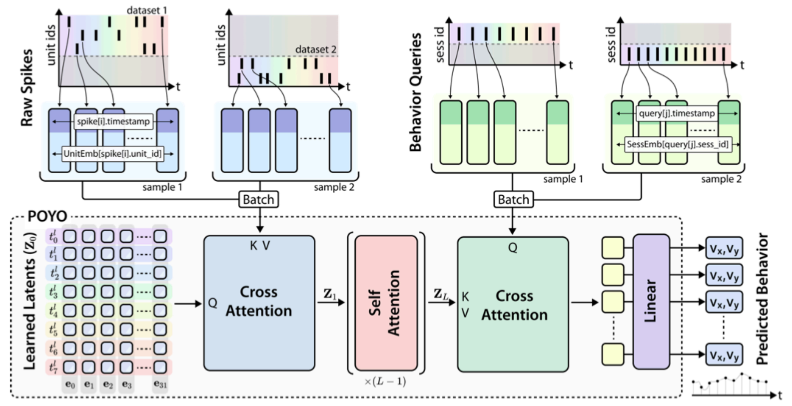
Contrastive (e.g. SimCLR)



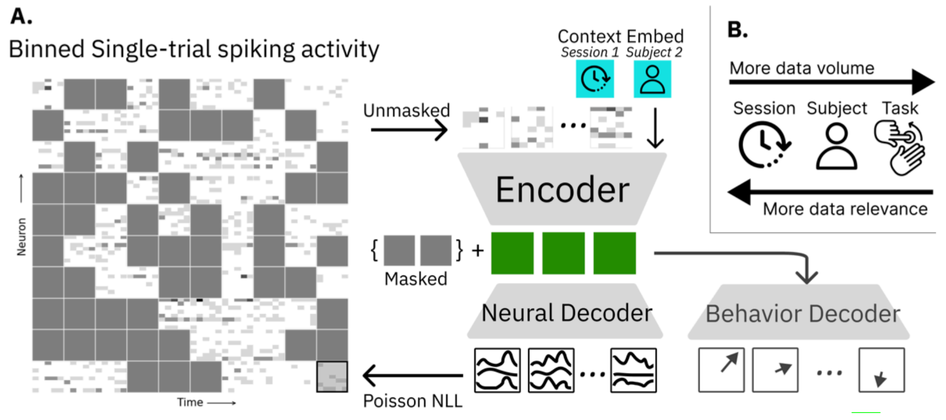


Neuro Foundation Model Success

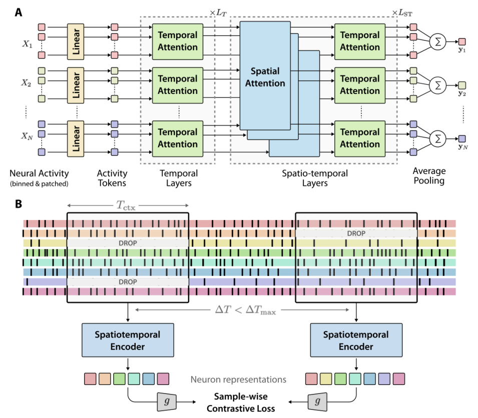
Spikes



POYO: Azabou et al., 2023



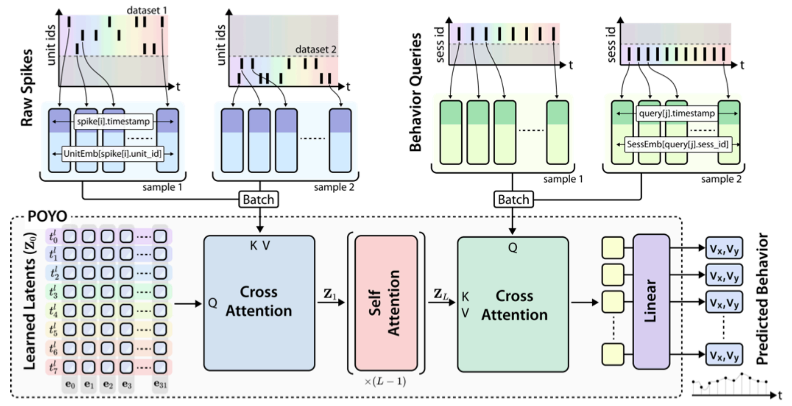
NDT2: Ye et al., 2023



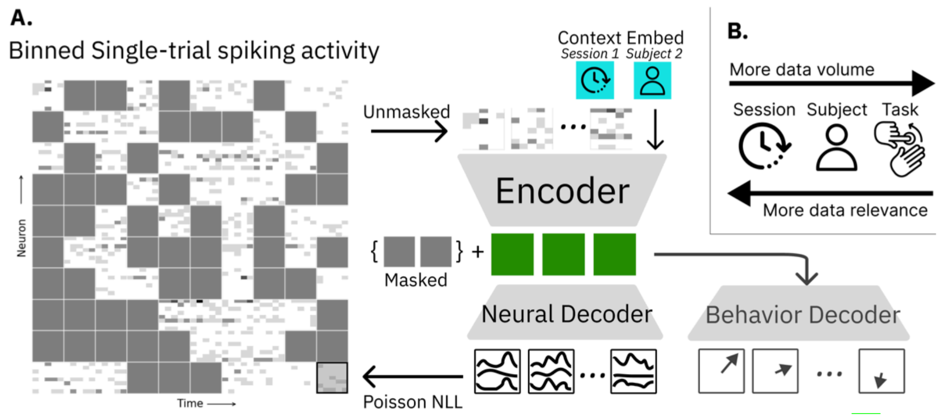
NuCLR: Arora et al., 2025

Neuro Foundation Model Success

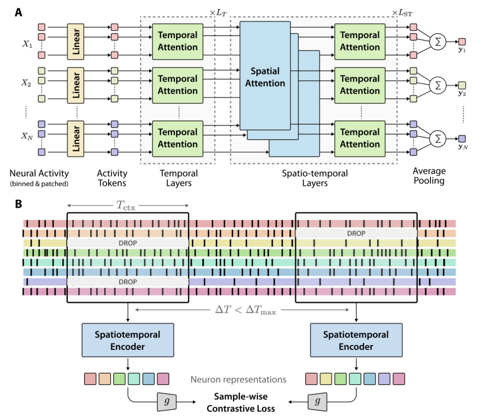
Spikes



POYO: Azabou et al., 2023

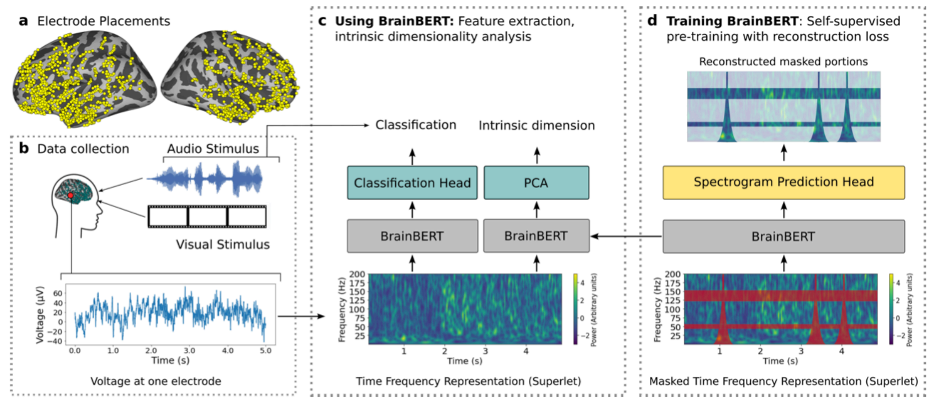


NDT2: Ye et al., 2023

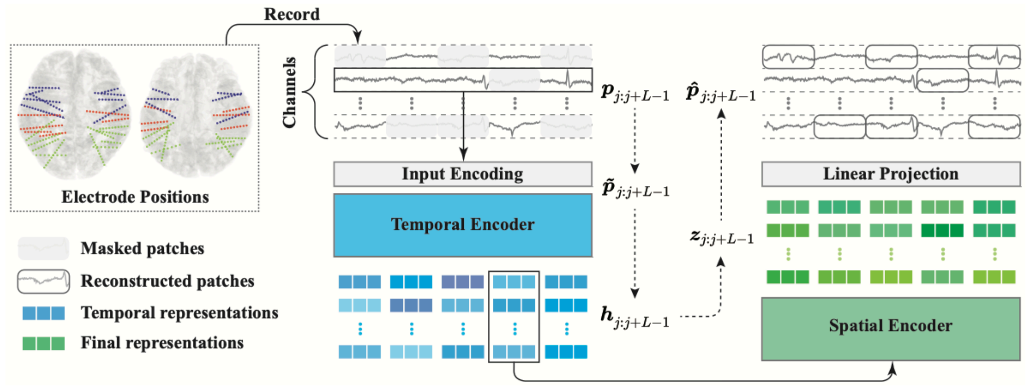


NuCLR: Arora et al., 2025

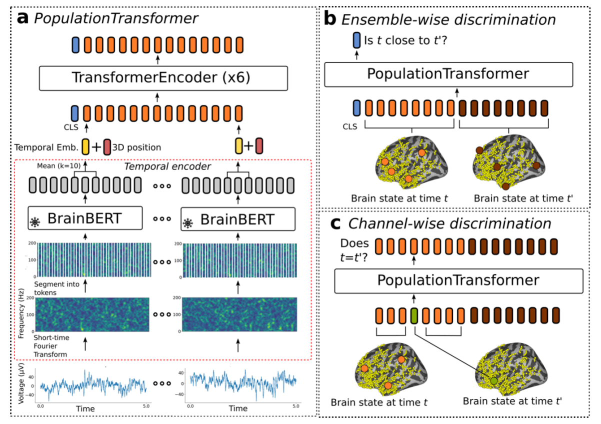
iEEG



BrainBERT: Wang et al., 2023



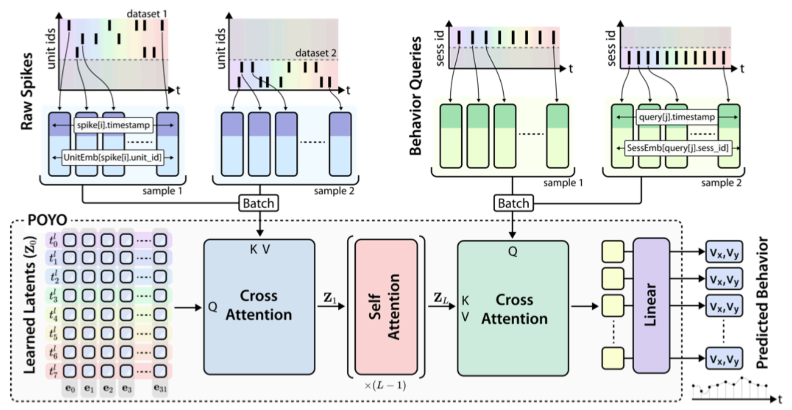
Brant: Zhang et al., 2023



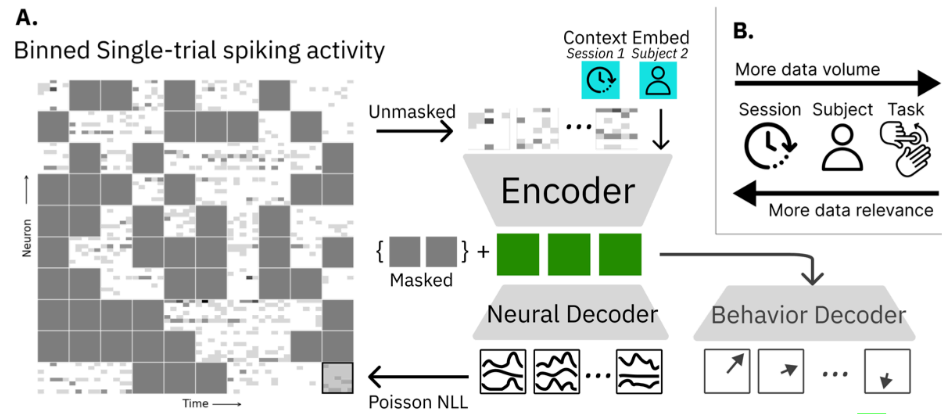
PopT: Chau et al., 2025

Neuro Foundation Model Success

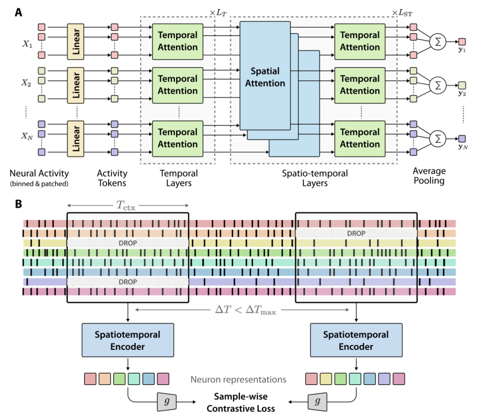
Spikes



POYO: Azabou et al., 2023

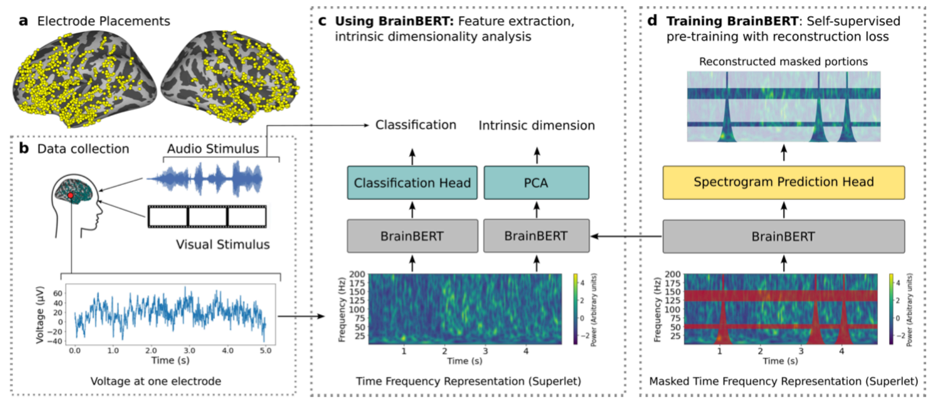


NDT2: Ye et al., 2023

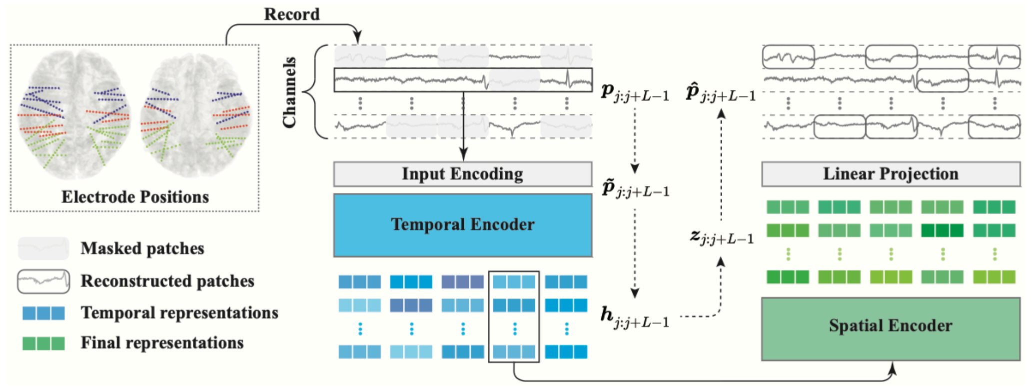


NuCLR: Arora et al., 2025

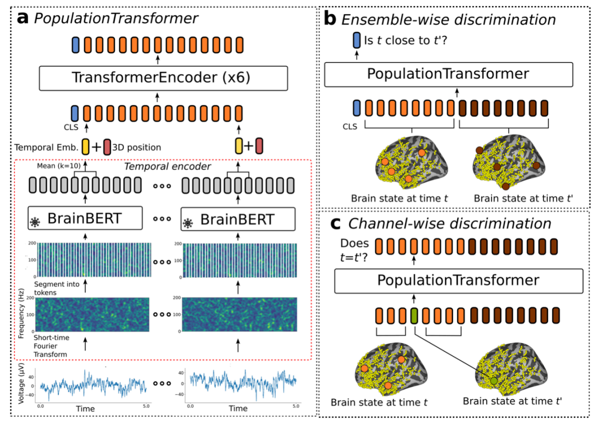
iEEG



BrainBERT: Wang et al., 2023

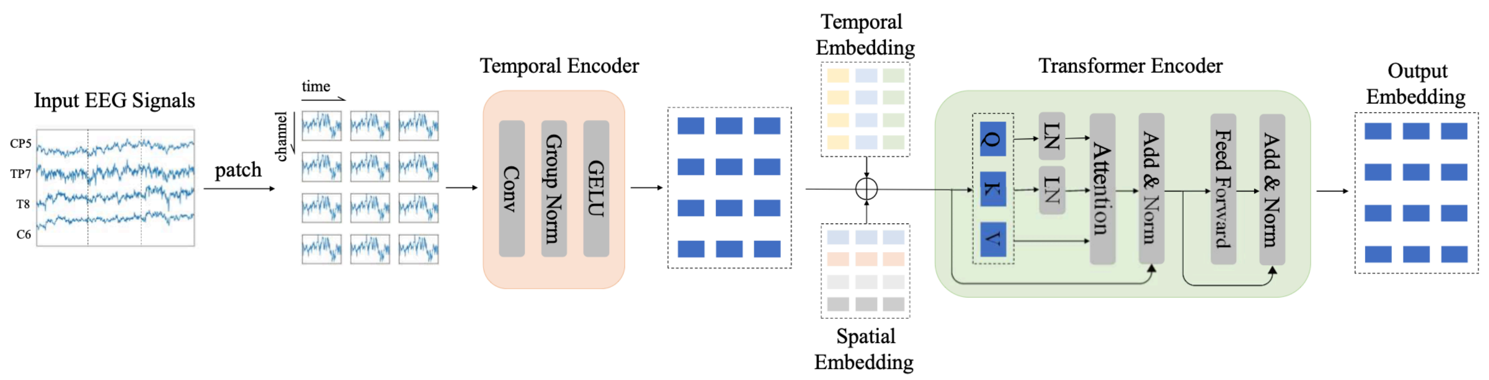


Brant: Zhang et al., 2023

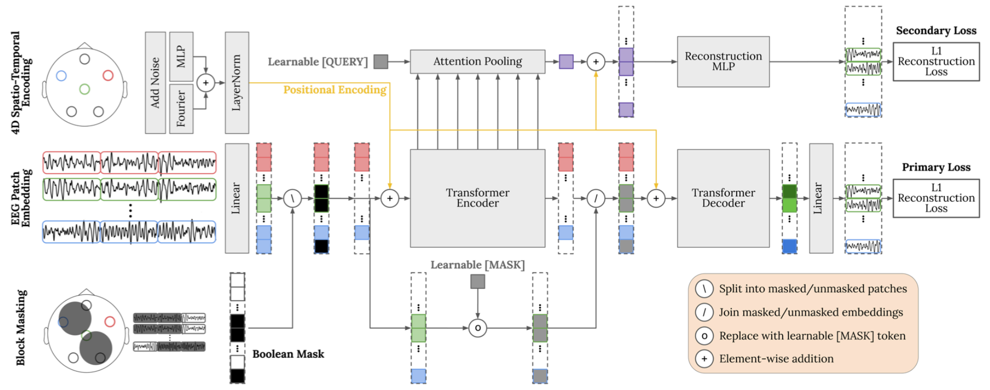


PopT: Chau et al., 2025

EEG



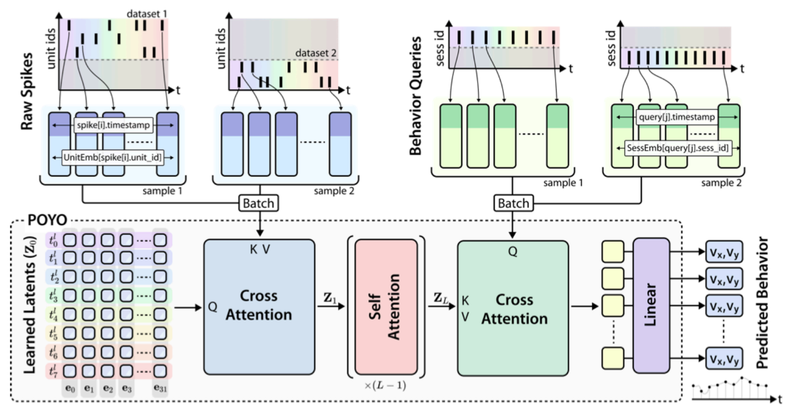
LaBraM: Jiang et al., 2024



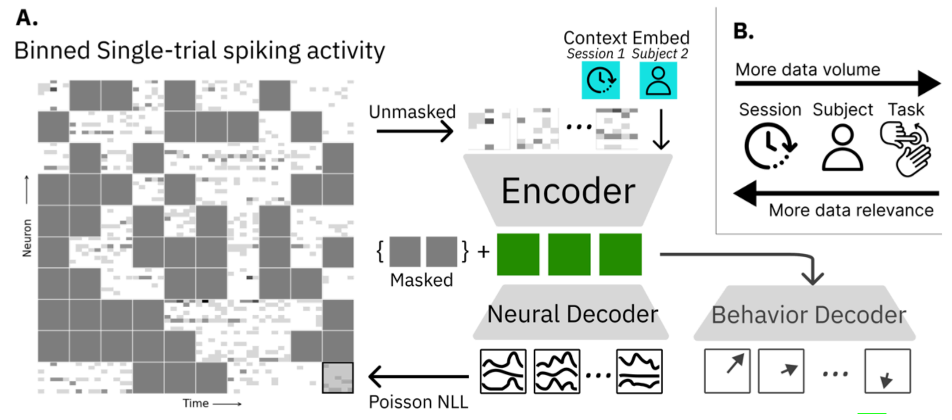
REVE: El Ouahidi et al., 2025

Neuro Foundation Model Success

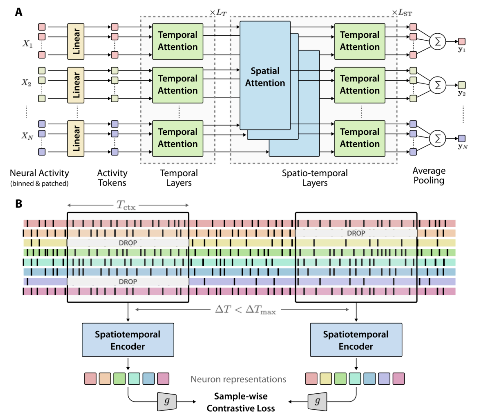
Spikes



POYO: Azabou et al., 2023

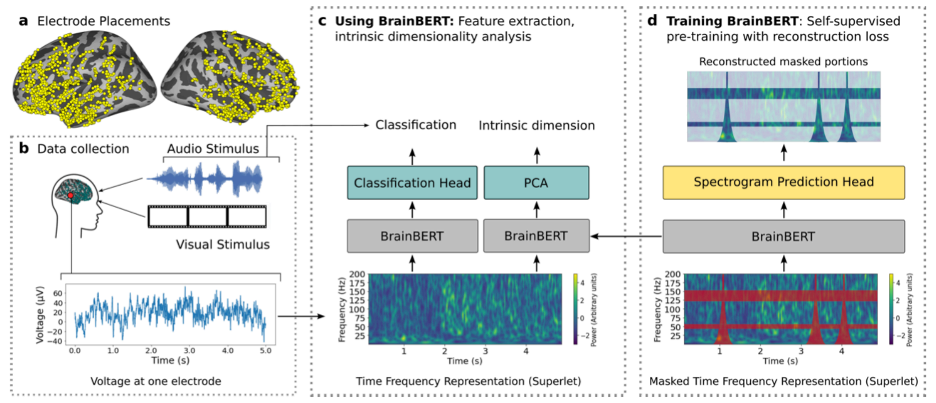


NDT2: Ye et al., 2023

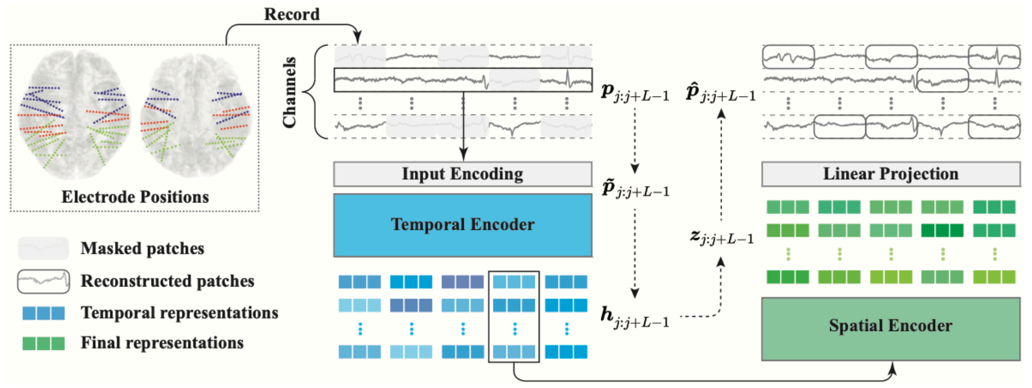


NuCLR: Arora et al., 2025

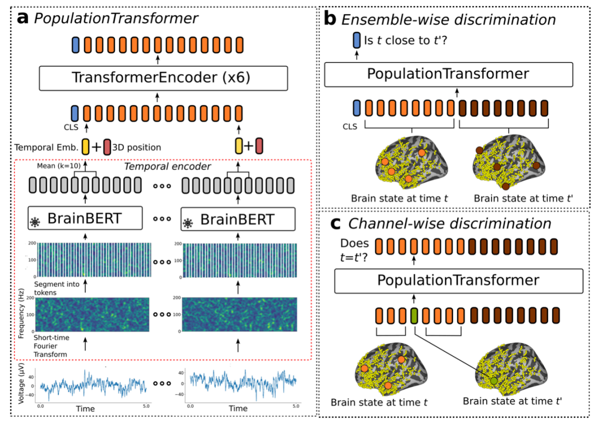
iEEG



BrainBERT: Wang et al., 2023

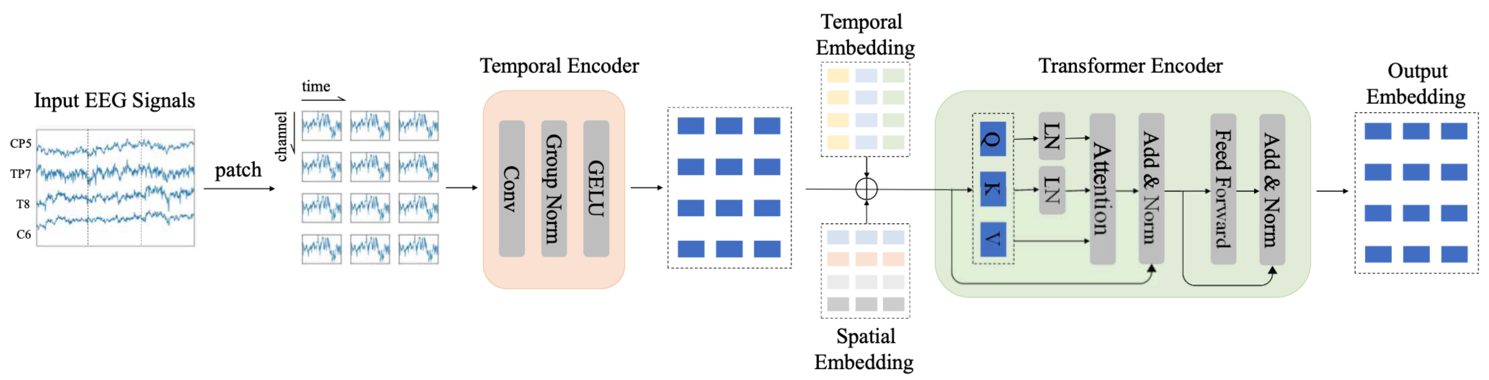


Brant: Zhang et al., 2023

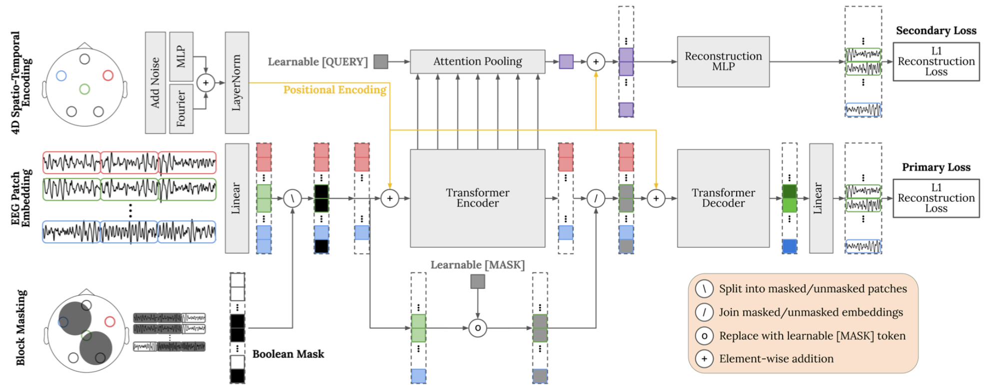


PopT: Chau et al., 2025

EEG

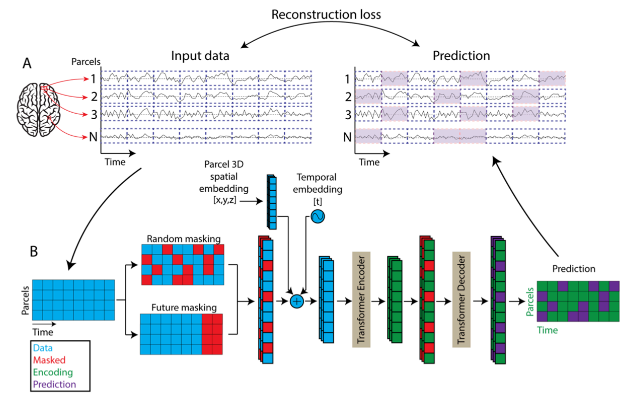


LaBraM: Jiang et al., 2024

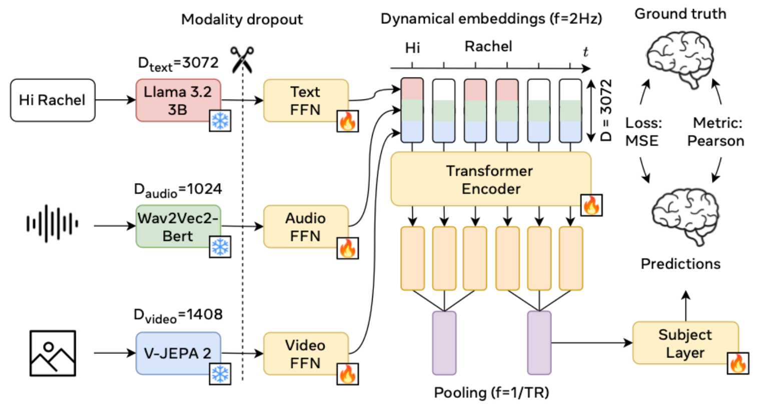


REVE: El Ouahidi et al., 2025

fMRI



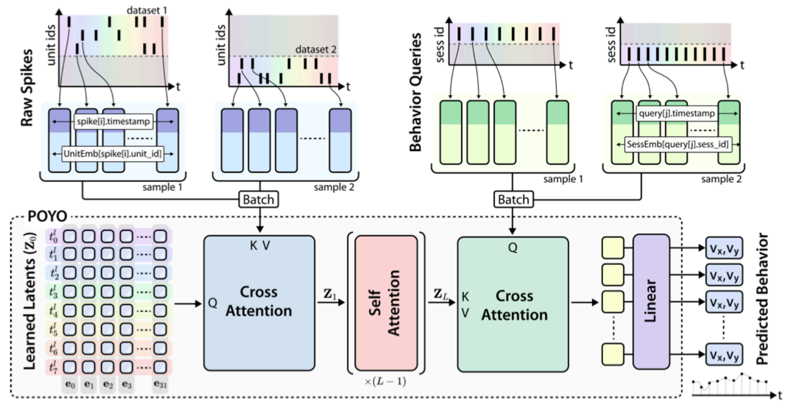
BrainLM: Caro et al., 2024



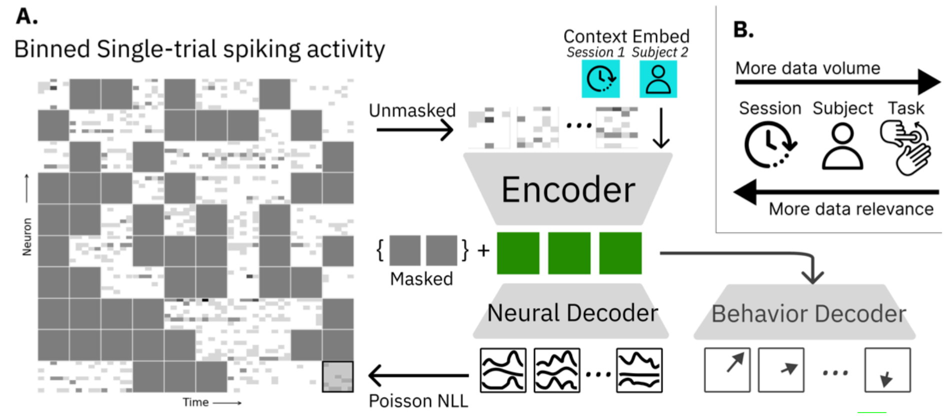
TRIBE v2: d'Ascoli et al., 2026

Neuro Foundation Model Success

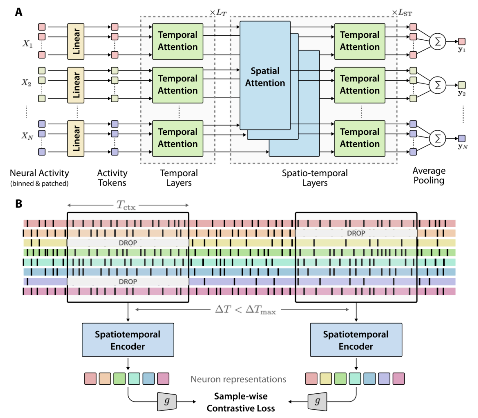
Spikes



POYO: Azabou et al., 2023

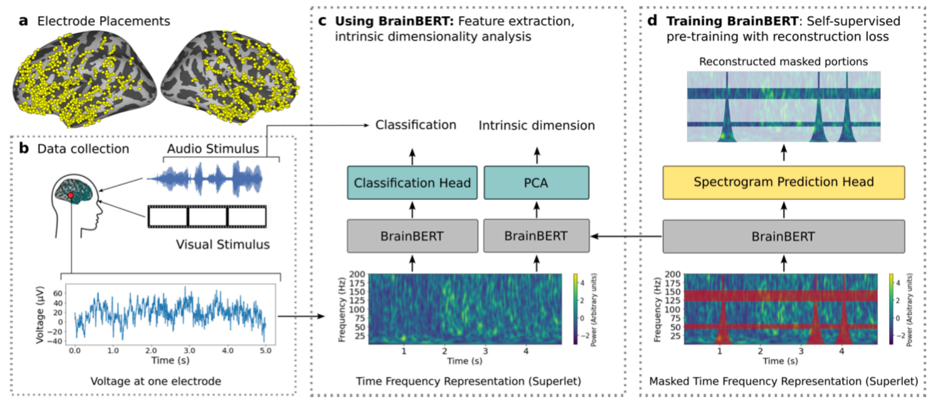


NDT2: Ye et al., 2023

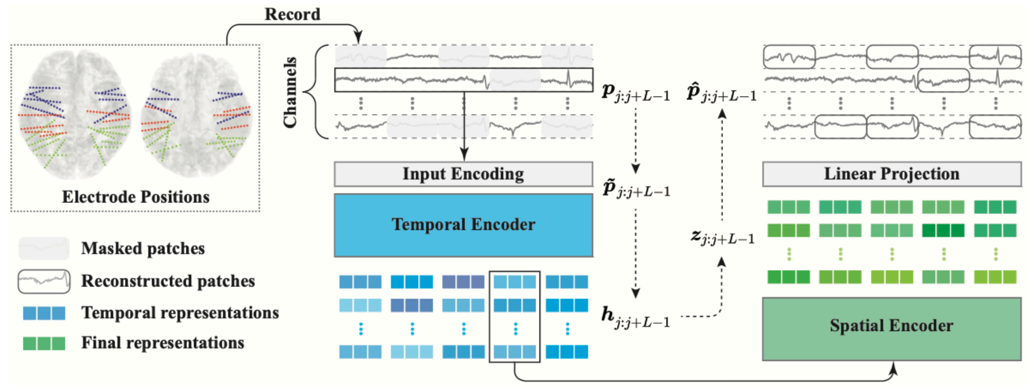


NuCLR: Arora et al., 2025

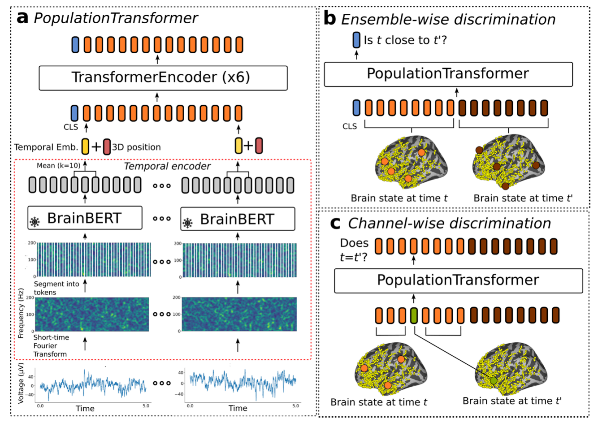
iEEG



BrainBERT: Wang et al., 2023

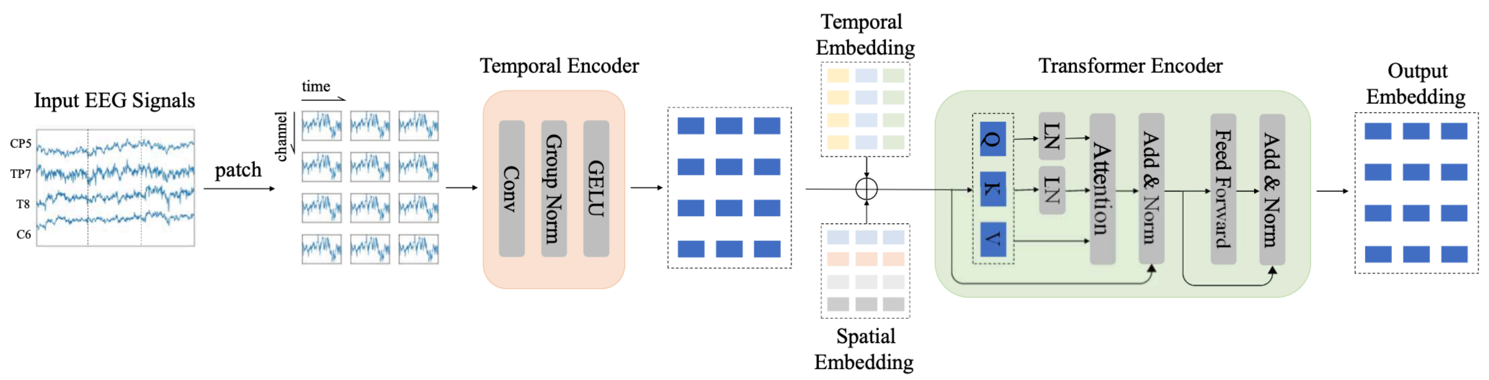


Brant: Zhang et al., 2023

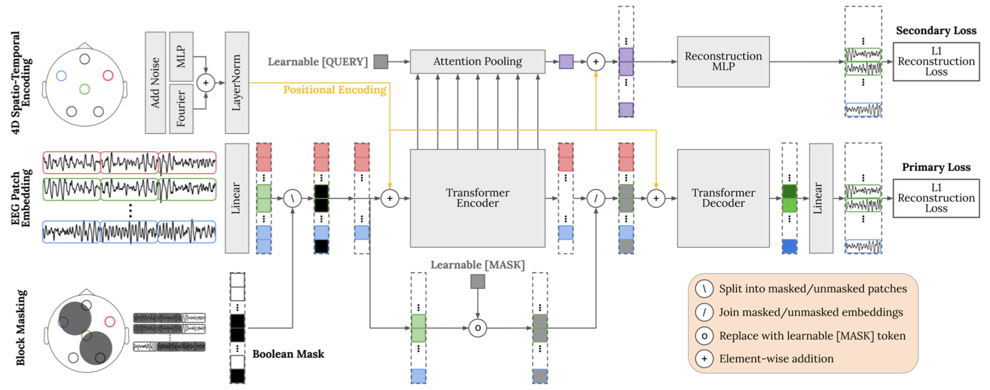


PopT: Chau et al., 2025

EEG

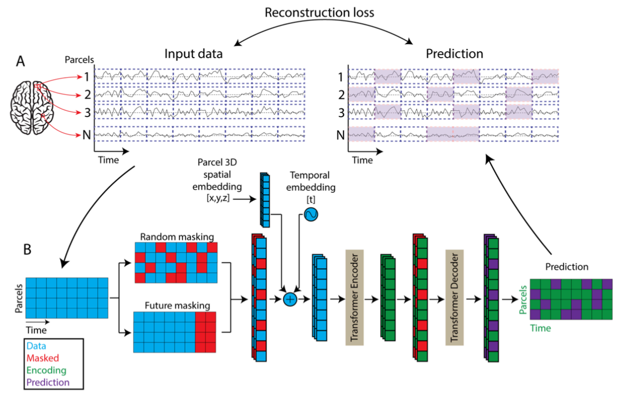


LaBraM: Jiang et al., 2024

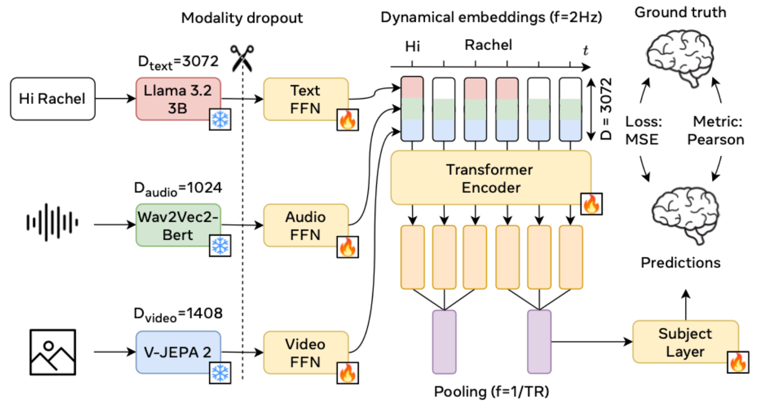


REVE: El Ouahidi et al., 2025

fMRI



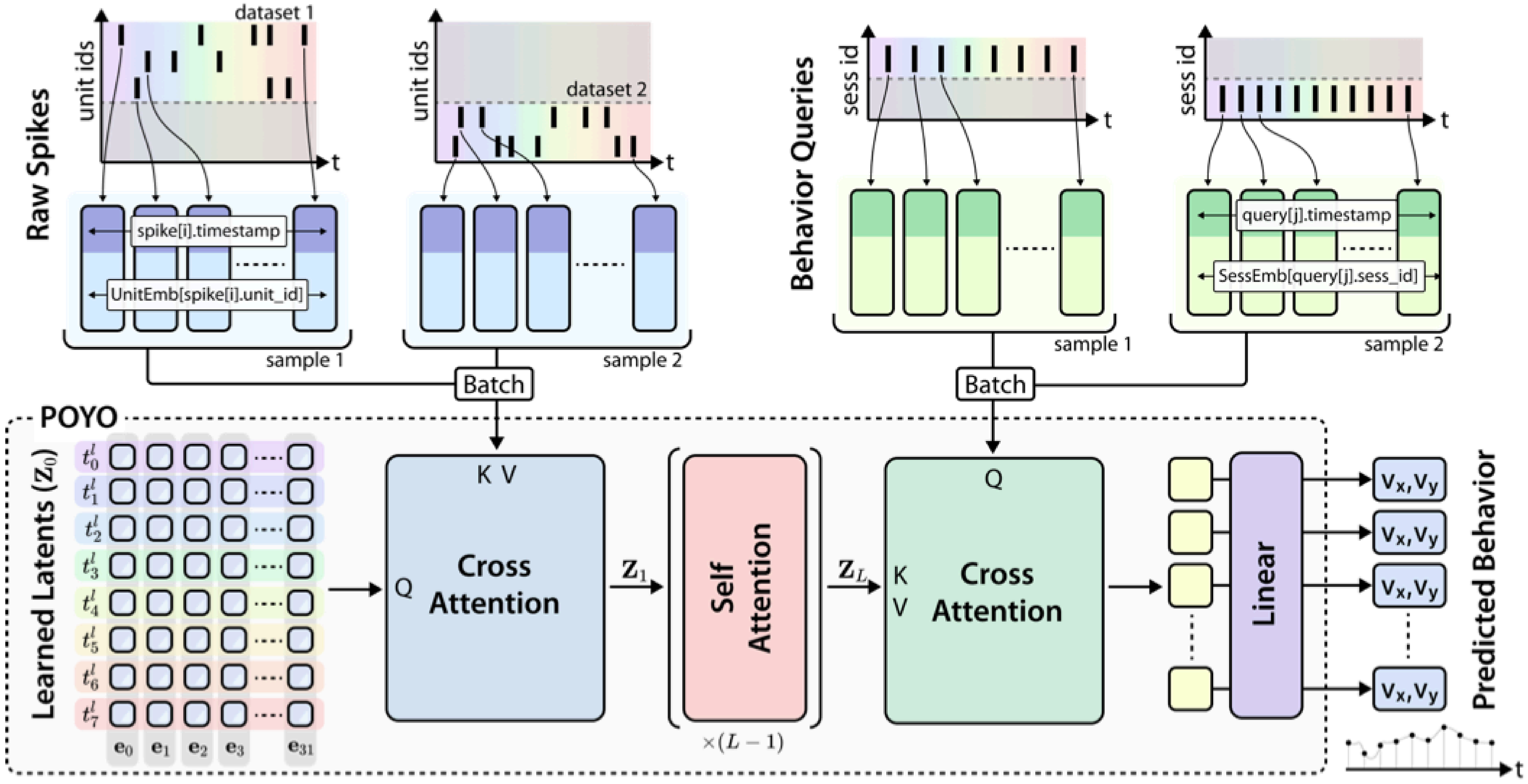
BrainLM: Caro et al., 2024



TRIBE v2: d'Ascoli et al., 2026

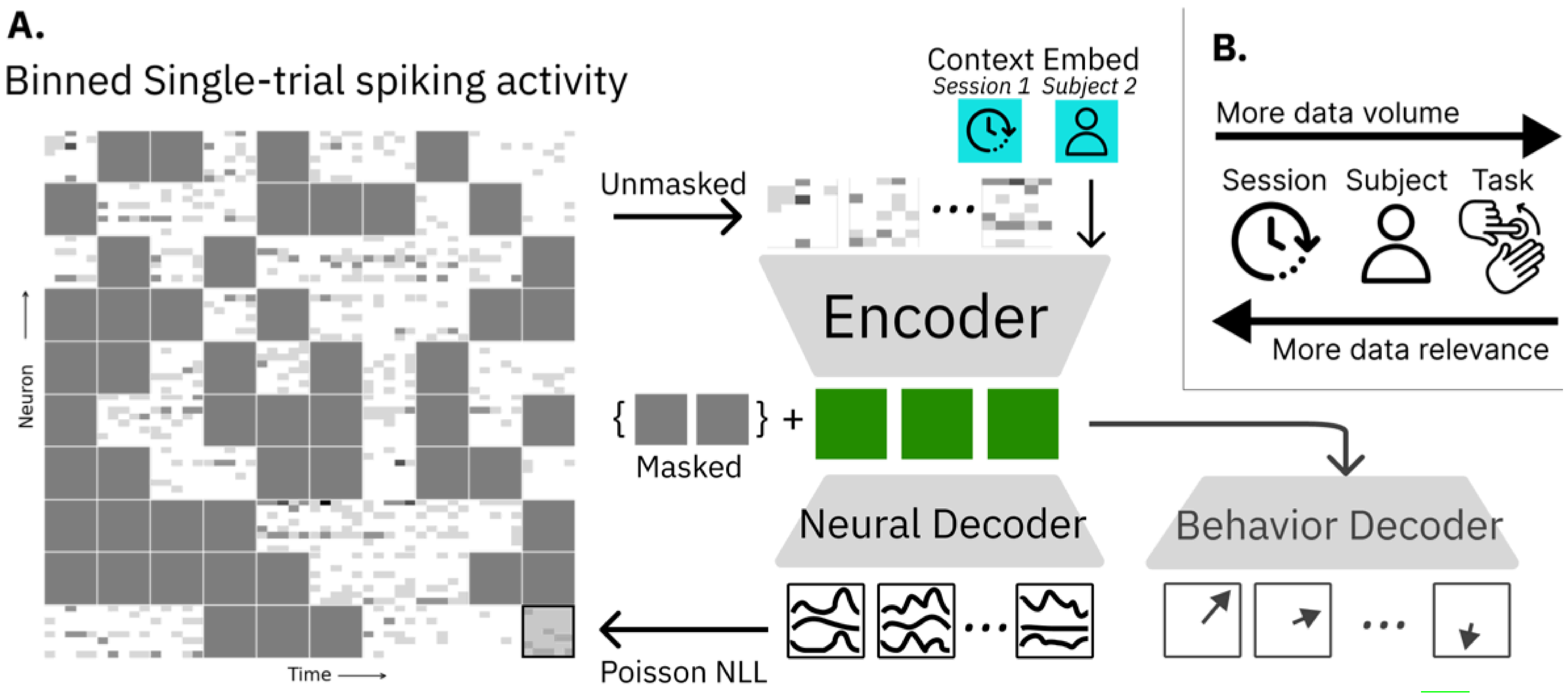
POYO: Azabou et al., 2023

- **Supervised decoding**
- **7 non-human primates**
- **3 labs**
- **158 sessions**
- **27,373 units**
- **>100 hours**



NDT2: Ye et al., 2023

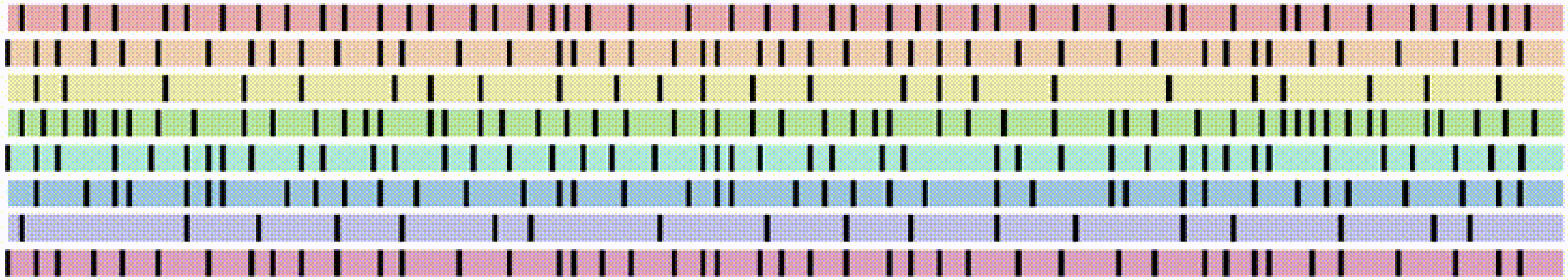
- **Mask SSL**
- **Behavior Decoding as downstream task**



NuCLR: Arora et al., 2025

- **Contrastive SSL**
- **Unit brain region classification as downstream task**

NuCLR: Arora et al., 2025



Why `torch_brain`?

*“A **PyTorch**-friendly open-source library for learning from **Neural Data**. Helps you write **efficient** training **pipelines** and **models**.”*

Why `torch_brain`?

*"A **PyTorch**-friendly open-source library for learning from **Neural Data**. Helps you write **efficient** training **pipelines** and **models**."*

Brainset:

- **Pipelines converting datasets** into `torch_brain`-compatible hdf5 files
- Curated, maintained **collection** of pipelines across recording modalities

DANDI

OpenNEURO

Why `torch_brain`?

*“A **PyTorch**-friendly open-source library for learning from **Neural Data**. Helps you write **efficient** training **pipelines** and **models**.”*

Brainset:

- **Pipelines converting datasets** into `torch_brain`-compatible hdf5 files
- Curated, maintained **collection** of pipelines across recording modalities

DANDI

OpenNEURO

Data structures:

- **Optimized** for **deep learning** (training + inference) on neural data

Why `torch_brain`?

*"A **PyTorch**-friendly open-source library for learning from **Neural Data**. Helps you write **efficient** training **pipelines** and **models**."*

Brainset:

- **Pipelines converting datasets** into `torch_brain`-compatible hdf5 files
- Curated, maintained **collection** of pipelines across recording modalities

DANDI

OpenNEURO

Data structures:

- **Optimized** for **deep learning** (training + inference) on neural data

Reusable building blocks:

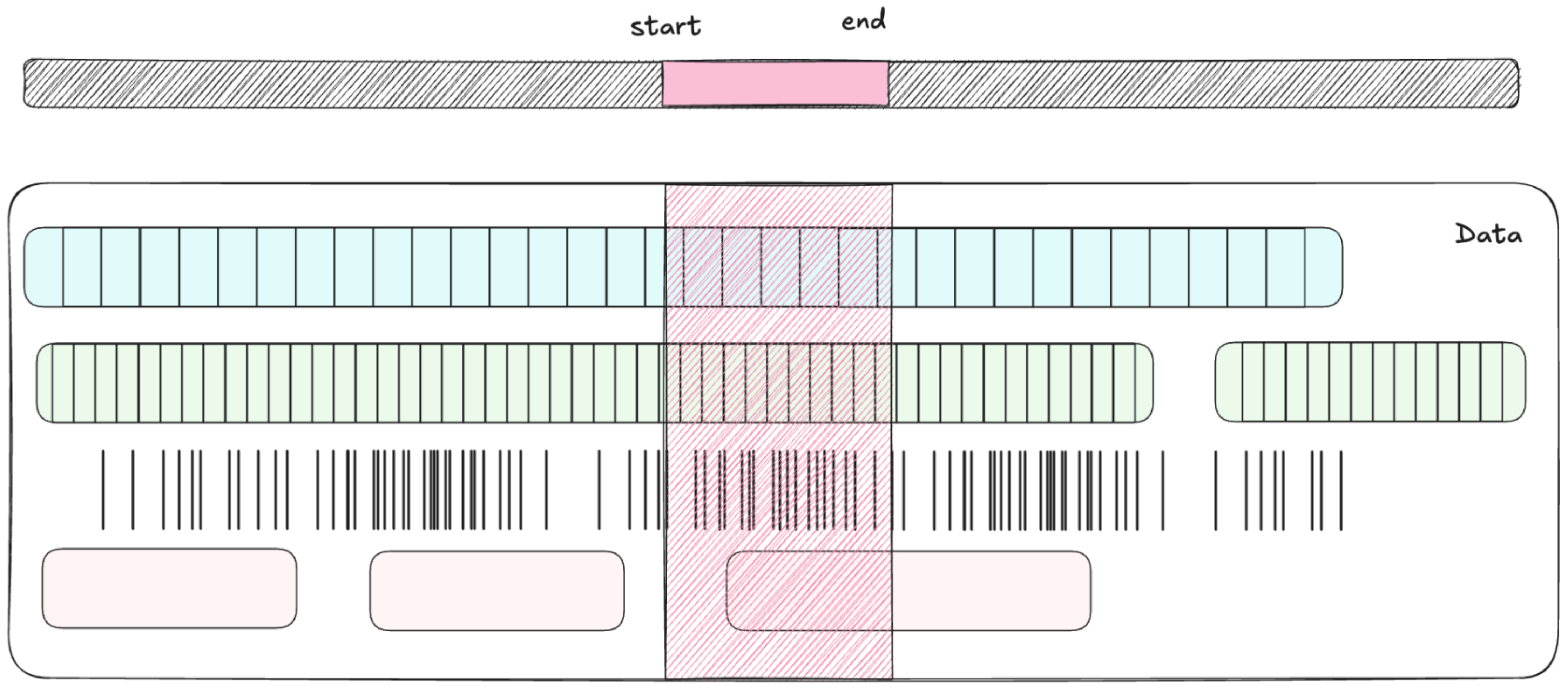
- **PyTorch** components: load, transform, batch diverse neural modalities
- Recent **models** and reference **examples**

PyTorch

Where `torch_brain` started?

The gap

- Load only the small chunk you need, not the full recording



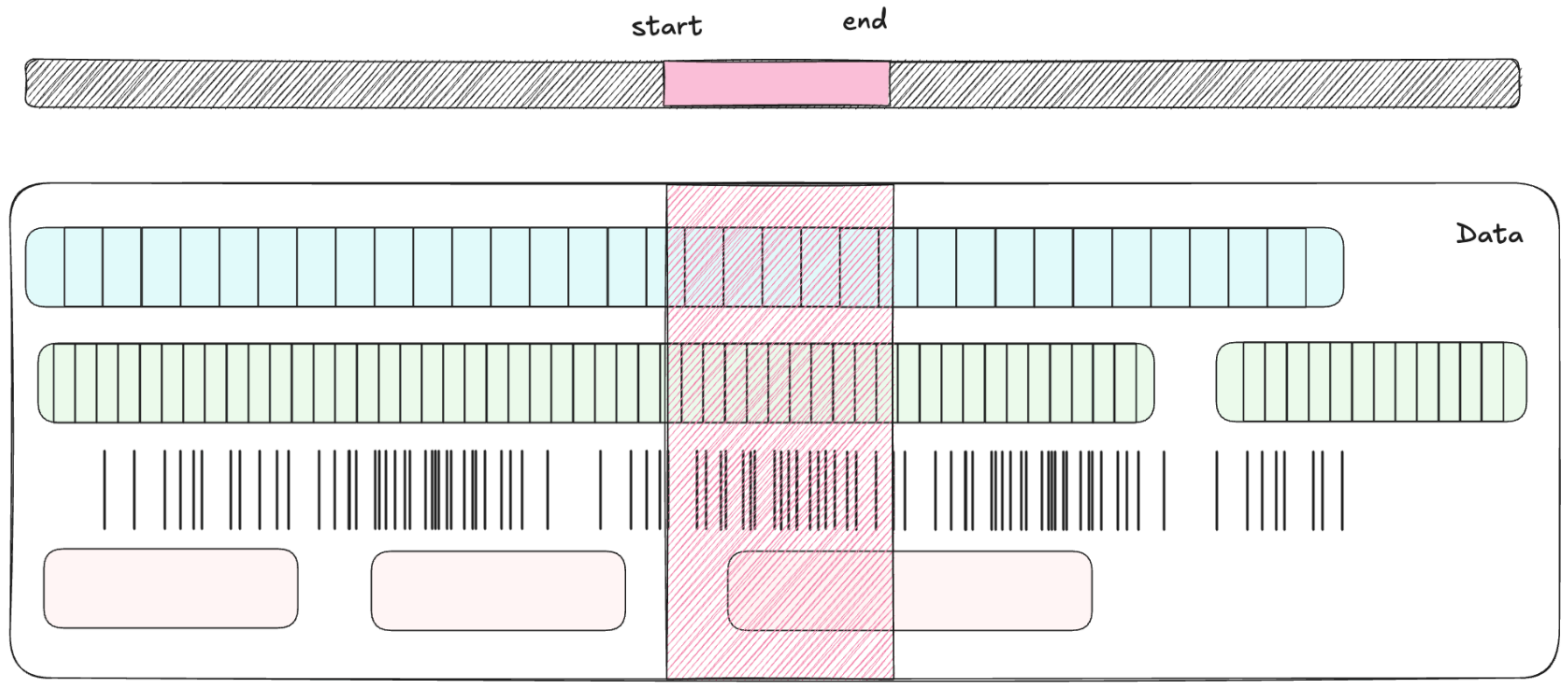
Where `torch_brain` started?

The gap

- Load only the small chunk you need, not the full recording

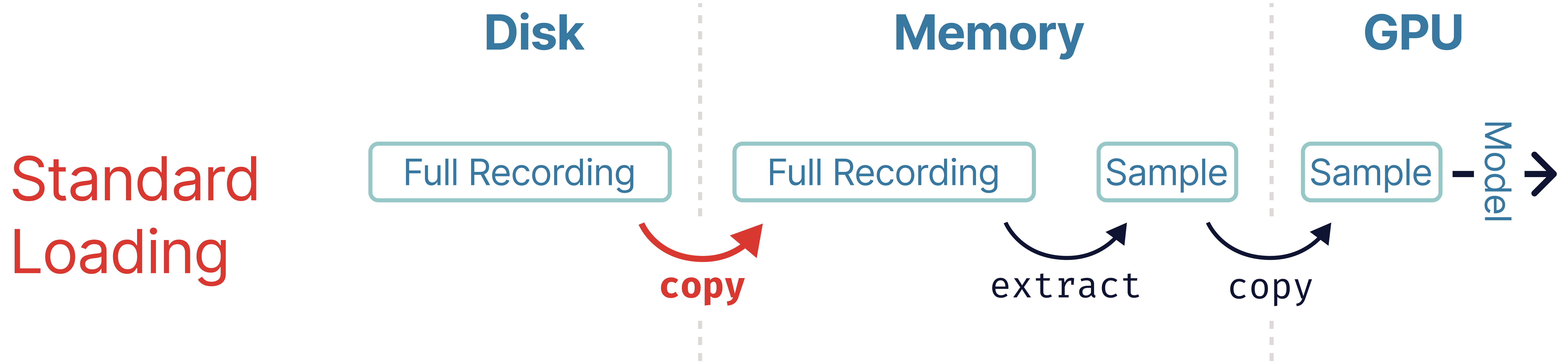
The solution

- Lazy loading, if data preserves temporal structure on disk
- HDF5 storage + optimized data structures (regular/irregular timeseries)

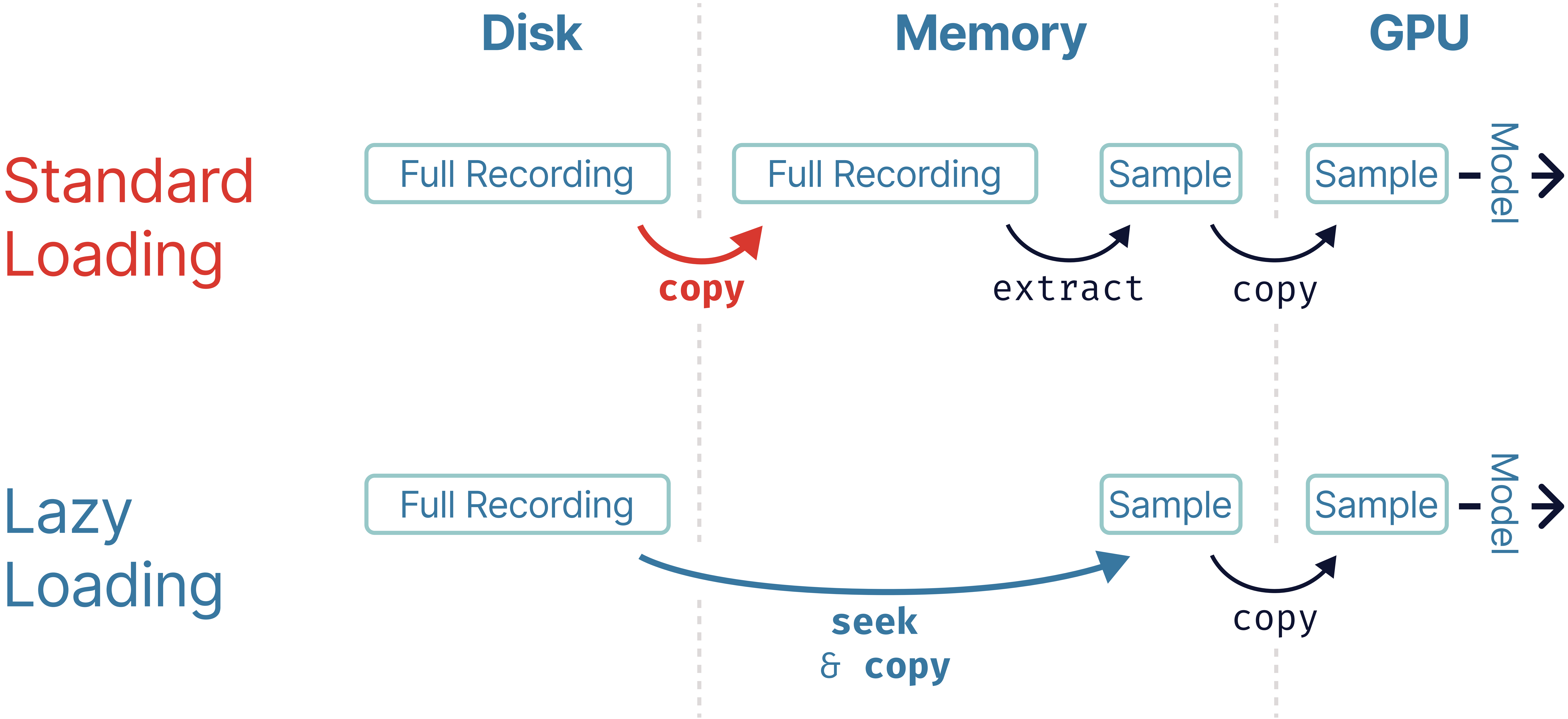


⇒ **Temporal lazy loading**: the initial motivation behind `torch_brain`

Standard Loading vs Lazy Loading



Standard Loading vs Lazy Loading



Demo

https://tinyurl.com/9y5ezksh

Key Takeaways & Next Steps

Data
Archive (Dandi)
Your own

Translation
brainset pipeline
torch_brain collection

Disk→GPU
fast/flexible
dataset/sampler

Model
Traditional PyTorch
`torch_brain`
building block

Key Takeaways & Next Steps

torch_brain

Data
Archive (Dandi)
Your own

Translation
brainset pipeline
torch_brain collection

Disk→GPU
fast/flexible
dataset/sampler

Model
Traditional PyTorch
`torch_brain`
building block

Key Takeaways & Next Steps

torch_brain

Data	Translation	Disk→GPU	Model
Archive (Dandi)	brainset pipeline	fast/flexible	Traditional PyTorch
Your own	torch_brain collection	dataset/sampler	torch_brain building block

- Support **any type of modalities**
- **Temporal lazy loading** is the enabler
- Reshape Disk→GPU pipeline, not the data
  - **Dataset** w/ 1 parameter: bin size, context window
  - **Samplers**
  - **Transforms**: unit dropout/filter
- Model Training Loop agnostic

Key Takeaways & Next Steps

torch_brain

Data
Archive (Dandi)
Your own

Translation
brainset pipeline
torch_brain collection

Disk→GPU
fast/flexible
dataset/sampler

Model
Traditional PyTorch
torch_brain
building block

- Support any **type of modalities**
- **Temporal lazy loading** is the enabler
- Reshape Disk→GPU pipeline, not the data
    - **Dataset** w/ 1 parameter: bin size, context window
    - **Samplers**
    - **Transforms**: unit dropout/filter
- Model Training Loop agnostic

- **Scale** from one session to many
- **Pretrain**, then **transfer**
- Contribute a brainset pipeline
- Explore examples + model zoo

Models Powered by `torch_brain`

- **POYO**: Azabou et al., 2023 (spikes, monkey, supervised)
- **POYO+**: Azabou et al., 2025 (calcium imaging, mice, multi-task)
- **NuCLR**: Arora et al., 2025 (spikes, mice, contrastive)
- **PoSSM**: Ryoo et al., 2025 (spikes, monkey/human, SSM for faster inference)
- **Charmander**: Mahato et al., 2025 (iEEG, human, SSL masking)
- **More coming soon**