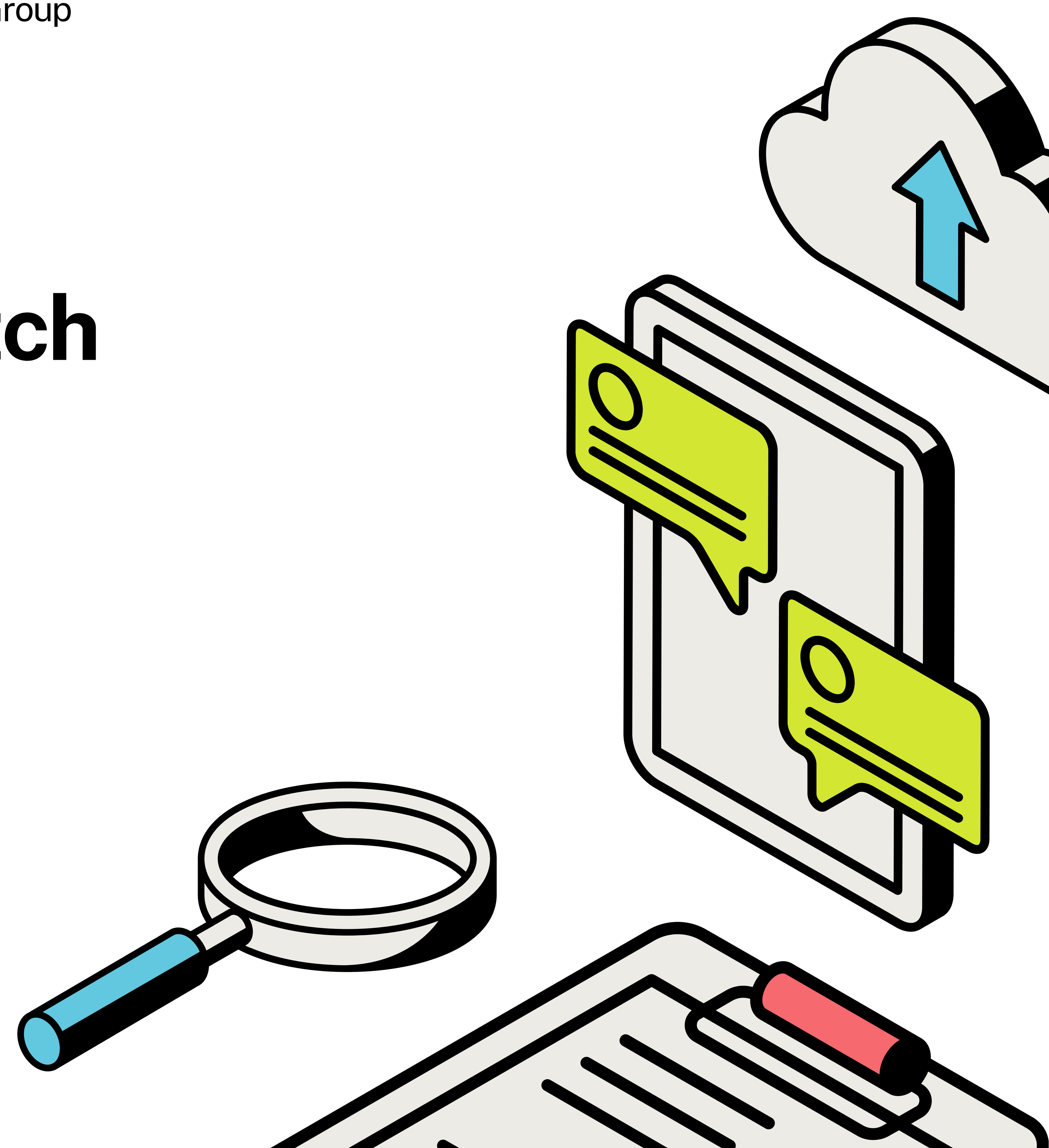
National Technical University of Athens
School of Electrical and Computer Engineering
Division of Signals, Control and Robotics
Computer Vision, Speech Communication and Signal Processing Group

Human Activity Recognition using Smartphone & Smartwatch Sensor Data

Alexandra Vioni

Supervisor: Professor Petros Maragos
Co-supervisor: Dr. Nancy Zlatintsi

20/10/2023



1. Human Activity Recognition
2. Motivation & Contributions
3. The ExtraSensory Dataset
4. Models, Experiments & Results
5. Discussion & Future Work

Human Activity Recognition (HAR)

HAR refers to the procedure of analyzing human body gesture or motion, using data retrieved from **sensors**, to automatically determine the activity performed by the user.

HAR has widespread **applications** in everyday life, predominantly in healthcare, elderly care, assisted living, human-computer interaction, assisted learning, and sports.

Sensors

- **Visual sensors**
  e.g., video cameras
- **Non-visual sensors**
  - **Wearable sensors**
    - **Biosensors**
      e.g., EEG, ECG
    - **Inertial sensors and others**
  - **Ambient sensors**
    (dense sensing)

Human Activity Recognition (HAR)

Data collection in-the-wild must abide by the following conditions:

- Naturally used devices
- Unconstrained device placement
- Natural environment
- Natural behavioral content

Motivation & Contributions

2

- To study the relevant **literature** on HAR based on non-visual, wearable sensors, to figure out its standard processing **pipeline**, to find **open datasets** and to understand major HAR **challenges**, trade-offs and open problems
- To get acquainted with HAR based on sensor data collected **in-the-wild**, to understand its **inherent flaws** and to study **existing approaches**
- To investistigate the use of **ML/DL models** to improve HAR on **ExtraSensory**, an open, **multi-label** dataset collected **in-the-wild** in an everyday life setup

The ExtraSensory Dataset

- Large-scale: 60 users, over 300k examples (minutes) in total
- In-the-wild data collection using everyday devices: multiple sensors from smartphone (Android/iOS) and smartwatch
- Created by UCSD researchers; participants recruited at the campus
- Real-time annotations via the ExtraSensory App, 51 labels
- Multiple labels annotation for each example (minute)

[VEL17] Vaizman, Y., Ellis, K., and Lanckriet, G. "Recognizing Detailed Human Context in the Wild from Smartphones and Smartwatches". In: IEEE Pervasive Computing 16.4 (Oct. 2017), pp. 62–74. doi: 10.1109/mprv.2017.3971131.
[VWL18] Vaizman, Y., Weibel, N., and Lanckriet, G. "Context Recognition In-the-Wild: Unified Model for Multi-Modal Sensors and Multi-Label Classification". In: Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies 1.4 (Jan. 2018), pp. 1–22. doi: 10.1145/3161192.

The ExtraSensory Dataset: Labels

ExtraSensory labels grouped conceptually	
Type	**Labels**
Posture/Movement	Lying down, Sitting, Standing, Walking, Running, Bicycling
Special Movement	Strolling, Stairs - Going up, Stairs - Going down, Elevator
Phone Location	Phone in pocket, Phone in hand, Phone in bag, Phone on table
Work-related	In class, Lab work, Computer work, In a meeting
Location-based	At home, At school, At main workplace, At a restaurant, At a bar, At a party, At the gym, At the beach
Transportation	In a car, On a bus, Drive - Driver, Drive - Passenger
Chores	Shopping, Cooking, Cleaning, Doing laundry, Washing dishes
Self-care	Bathing - Shower, Toilet, Grooming, Dressing, Sleeping
Leisure Time	Exercise, Eating, Drinking alcohol, Watching TV, Surfing the internet, Talking, Singing
Companion	With co-workers, With friends
Environment	Indoors, Outside

Table 1: Intuitive grouping of activity and context labels of the ExtraSensory dataset

Each example of the dataset has annotations for all labels: 1 (relevant), 0 (non-relevant) or NaN (missing)

3

The ExtraSensory Dataset: Labels

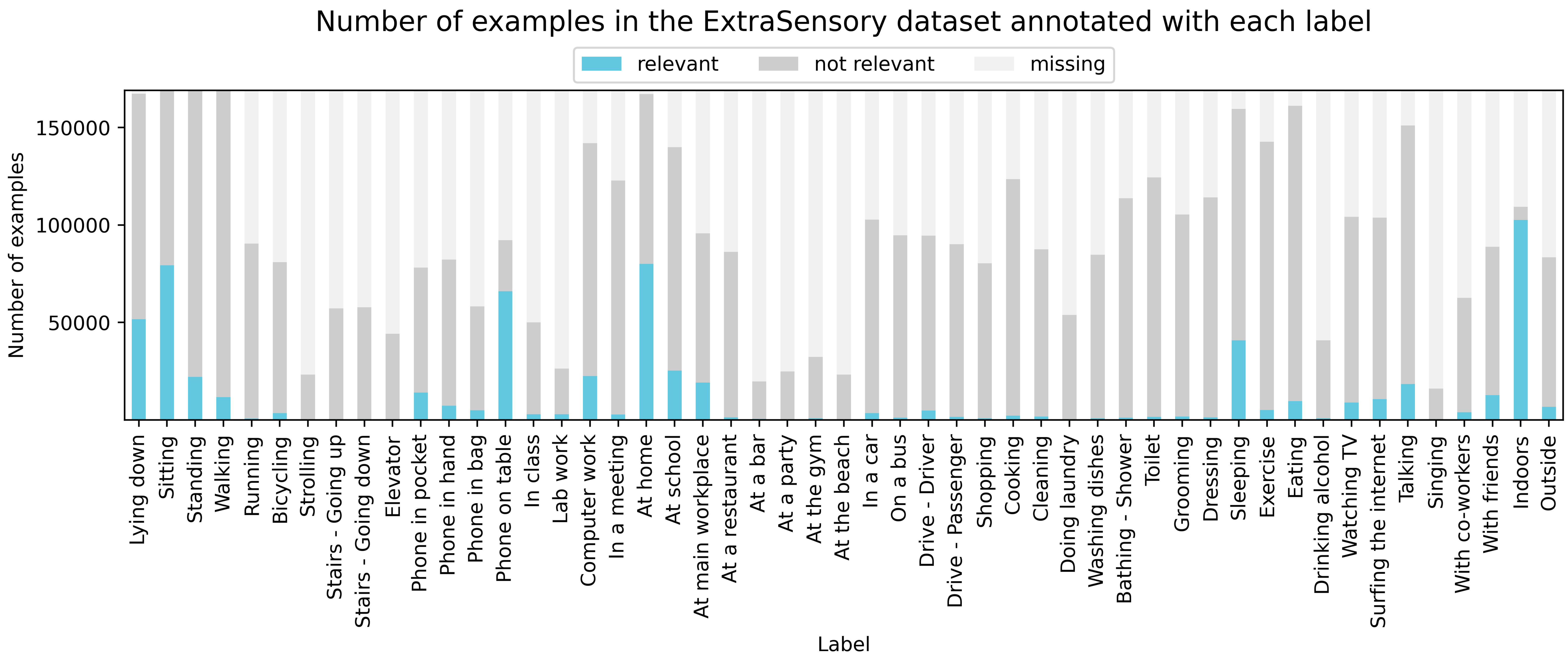


Figure 1: Number of ExtraSensory examples annotated with each label (ExtraSensory Core subset)

The ExtraSensory Dataset: Sensor Data & Features

We use the Core subset, which includes all the examples (169,001) with measurements from the following sensors:

- **Smartphone's accelerometer (Acc)** sampled at 40Hz
  Sensor recordings: 3-axis time-series (3, 800) Extracted features: 26
- **Smartphone's gyroscope (Gyro)** sampled at 40Hz
  Sensor recordings: 3-axis time-series (3, 800) Extracted features: 26
- **Smartwatch's accelerometer (WAcc)** sampled at 25Hz
  Sensor recordings: 3-axis time-series (3, 500) Extracted features: 46
- **Smartphone's location (Loc)** sampled at varying rate (when movement is detected)
  Sensor recordings: long–lat–alt (var) Extracted features: 17
- **Smartphone's audio (Aud)** sampled at 22050Hz
  Time-series data: 13 MFCC (13, 700) Extracted features: 26
- **Smartphone's phone state (PS)** sampled once per example
  Time-series data: - Extracted features: 34

For every minute, the ExtraSensory App recorded a 20sec window of sensor measurements from the phone and watch

The ExtraSensory Dataset: Challenges

HAR challenges that arise when using in-the-wild data collection include:

- Multi-label dataset
- Unbalanced dataset
- Noisy data
- Missing sensors
- Missing or wrong labels
- Inter-personal & intra-personal variability

Experimental Setup

Model design choices:

- Input: **pre-extracted features** or **raw sensor data**
- Time-series modeling: a **single example** or a **sequence of examples**

All the features or raw sensor data that are used as input, are first **standardized** using the mean and standard deviation of the training set.

Missing feature values are **zero-imputated** after standardization.

All neural networks are implemented in **PyTorch**. Logistic Regression and the evaluation metrics are based on **scikit-learn**.

Experimental Setup

Based on **Binary Cross-Entropy loss**, we implement a **custom loss**:

- **per-batch, per-element** we mask the loss elements corresponding to **missing ground-truth labels** for each example.
- **per-label** we multiply the term of the positive examples in the loss, with the **ratio of negative to positive examples** for this label in the training set, to account for the **imbalance** in the number of positive examples per label.

4

We use the **Adam** optimizer and a **batch size** of **32** to train all our models.

In the testing phase, we use a **threshold** of **0.5** to convert the output values after the **sigmoid** activation function to **binary** outputs.

Evaluation Scheme

- The models are always tested on users **unseen** during training.
- We use a **five-fold cross validation (CV)** scheme with **12 users in each fold**. In each of the five CV iterations we have **48 users** in the **training set** and **12 users** in the **test set**.
- For each of the 48 users of the training set, **80%** of their data is used for training, and **20%** is used for validation
- For each label, we count the numbers of **True Positives (TP)**, **True Negatives (TN)**, **False Positives (FP)**, and **False Negatives (FN)** of the prediction results over the test set, for the five CV iterations.

Evaluation Metrics

We calculate the following **metrics** for each **label** (over its non-missing ground-truth examples):

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

$$\text{Sensitivity} = \text{Recall} = \text{TPR} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

$$\text{Specificity} = \text{TNR} = \frac{\text{TN}}{\text{TN} + \text{FP}}$$

$$\text{F1-score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

$$\text{Balanced Accuracy (BA)} = \frac{\text{TPR} + \text{TNR}}{2}$$

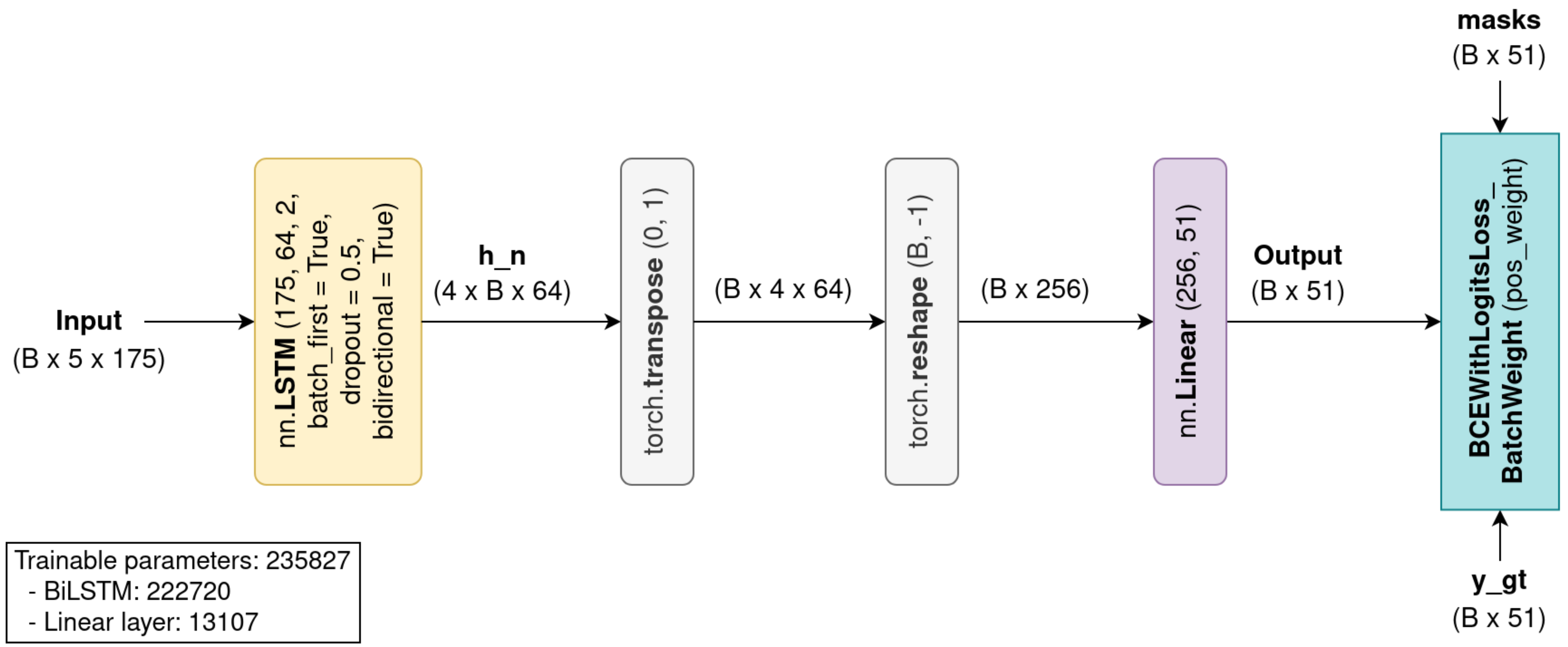
Moreover, we **average** each of the **metrics over all labels**.

Baselines

Comparative results overview of the performance metrics averaged over all labels for the baseline models								
Input	**Time-series modeling**	**Model**	**Accuracy**	**Precision**	**Sensitivity**	**Specificity**	**F1-score**	**BA**
		Random classifier	0.500	0.110	0.500	0.500	0.137	0.500
		Majority class classifier	**0.915**	NaN	0.040	**0.958**	0.037	0.499
Extracted features	Single example	Logistic Regression [VWL18]	**0.832**	-	0.597	**0.838**	-	0.718
		Logistic Regression	**0.839**	**0.246**	0.612	**0.844**	**0.314**	0.728
		MLP [VWL18]	0.773	-	**0.773**	0.773	-	**0.773**
		MLP	0.786	0.228	0.757	0.786	0.298	**0.772**

Table 2: An overview of the recognition scores of the baseline models, averaged for all labels

Bidirectional LSTM (only final hidden states)



Input sequence length: {**5**, 10, 15, 30}
BiLSTM number of layers: {1, **2**, 3}
BiLSTM hidden size: {16, 32, **64**}
BiLSTM dropout: **0.5**
Learning rate: **0.00002**

Figure 2: BiLSTM model architecture, using only the final hidden states

Accuracy	Precision	Sensitivity	Specificity	F1-score	BA
0.813	0.243	0.753	0.814	0.316	0.784

Table 3: Recognition scores of the BiLSTM model using only the final hidden states, averaged for all labels

Bidirectional LSTM (output for all timesteps)

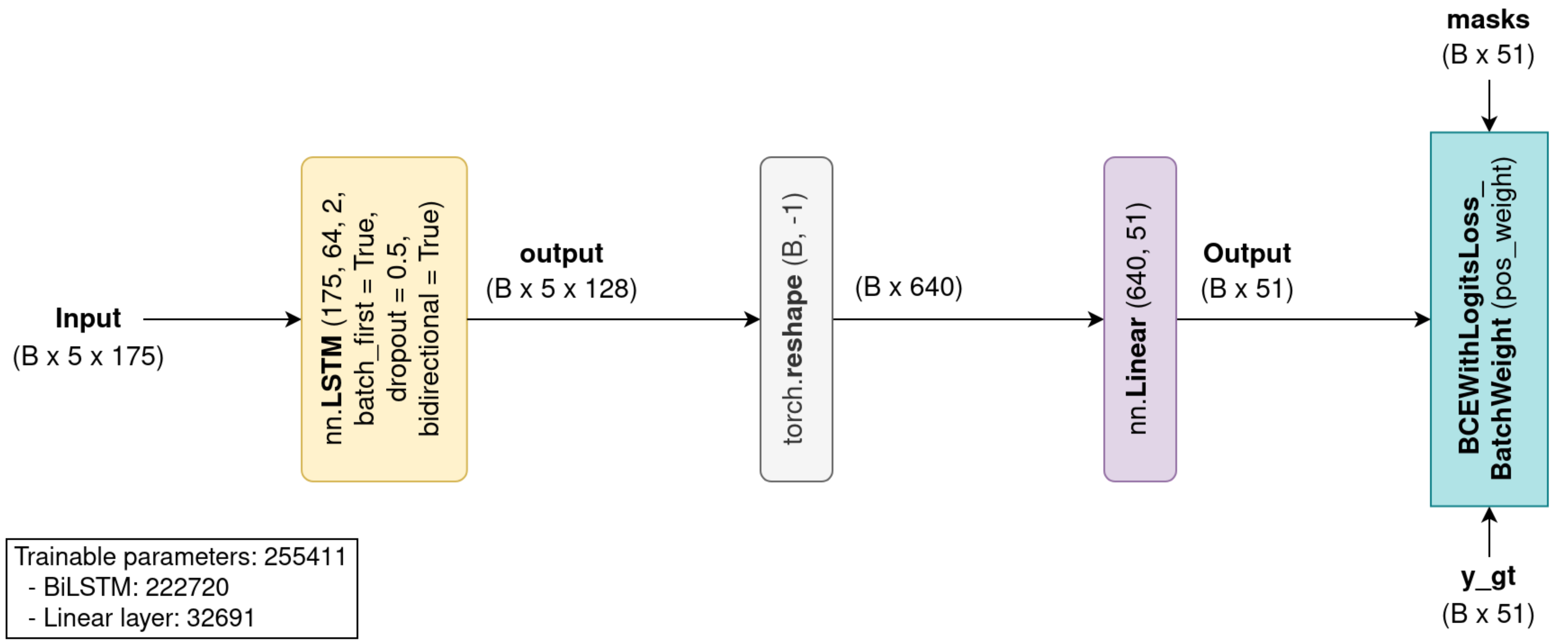


Figure 3: BiLSTM model architecture, using the output for all timesteps

Input sequence length: {**5**, 10, 15, 30}
BiLSTM number of layers: {1, **2**, 3}
BiLSTM hidden size: {16, 32, **64**}
BiLSTM dropout: **0.5**
Learning rate: **0.00001**

Accuracy	Precision	Sensitivity	Specificity	F1-score	BA
0.810	0.241	0.761	0.811	0.314	0.786

Table 4: Recognition scores of the BiLSTM model using the output for all timesteps, averaged for all labels

Self-Attention & Bidirectional LSTM

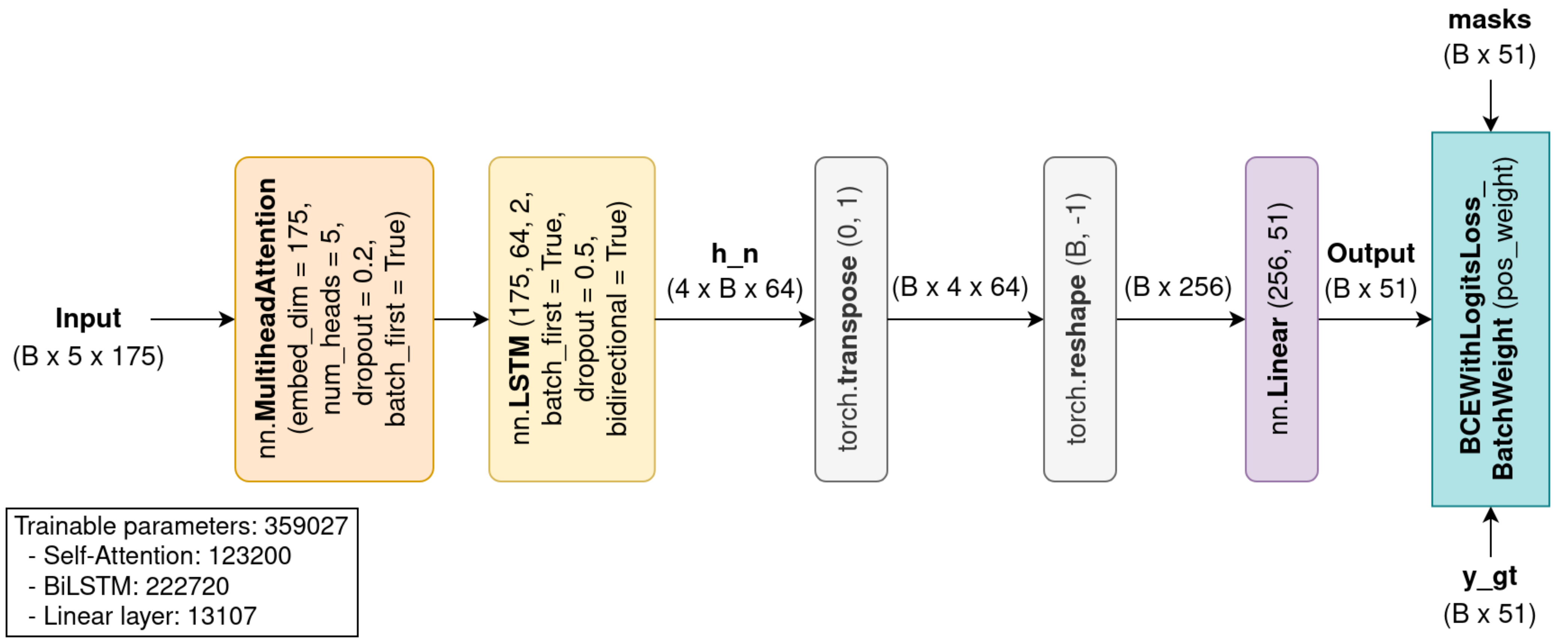


Figure 4: Self-Attention & BiLSTM model architecture, using only the final hidden states

Input sequence length: {**5**, 10, 15, 30}
Attention heads: **5**
Attention dropout: **0.2**
BiLSTM number of layers: {1, **2**, 3}
BiLSTM hidden size: {16, 32, **64**}
BiLSTM dropout: **0.5**
Learning rate: **0.00005**

Accuracy	Precision	Sensitivity	Specificity	F1-score	BA
0.818	0.248	0.756	0.819	0.323	0.788

Table 5: Recognition scores of the Self-Attention & BiLSTM model using only the final hidden states, averaged for all labels

Self-Attention & Bidirectional LSTM: Activity Plots

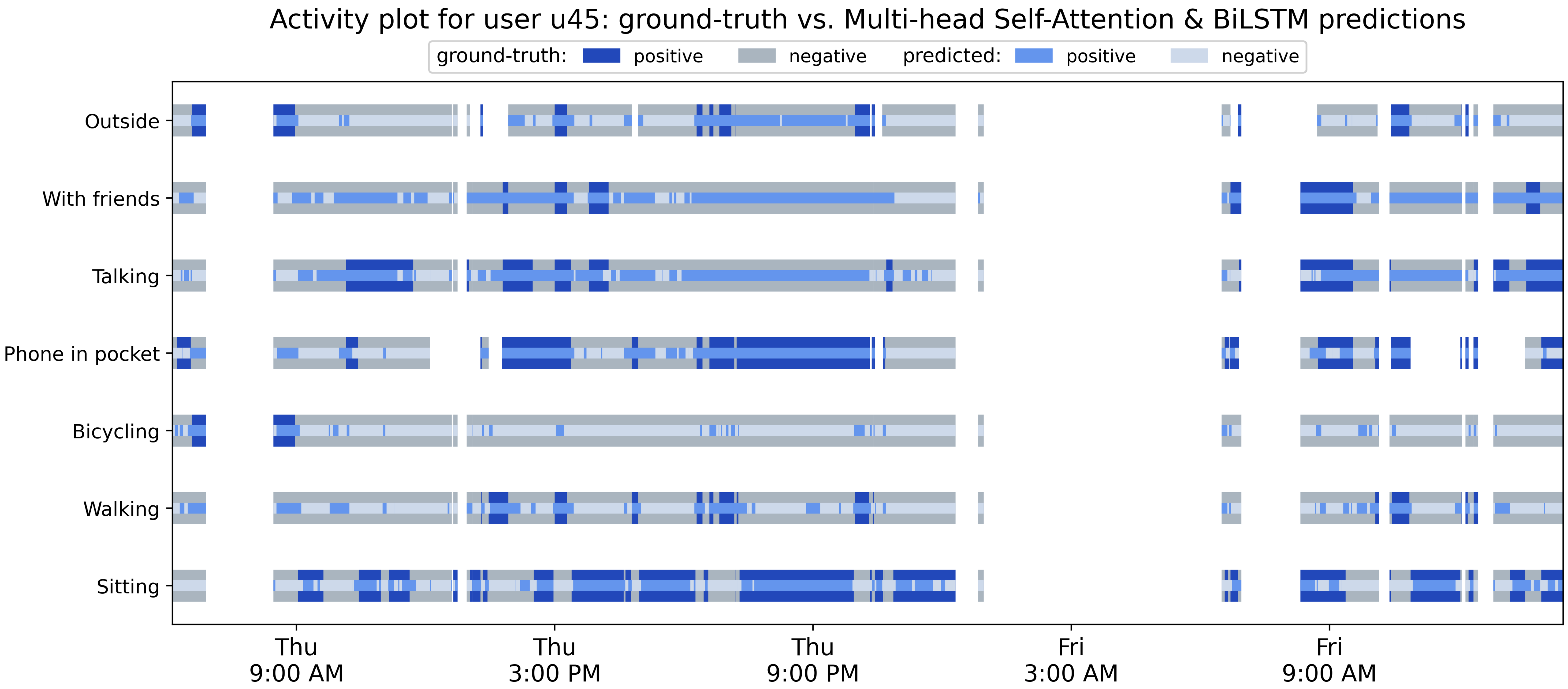


Figure 5: Activity plot including the predictions of the Multi-head Self-Attention & BiLSTM model using final hidden states, for user u45

Self-Attention & Bidirectional LSTM: Activity Plots

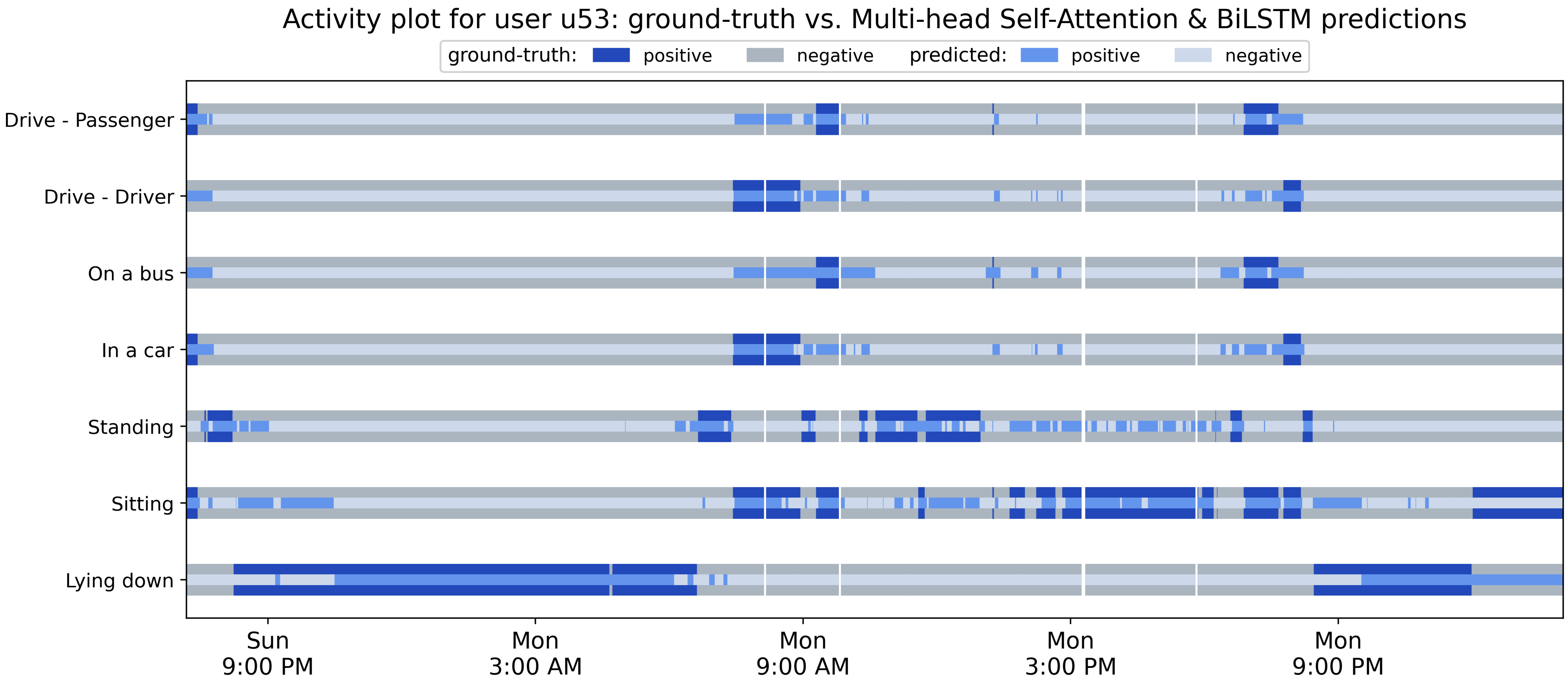


Figure 6: Activity plot including the predictions of the Multi-head Self-Attention & BiLSTM model using final hidden states, for user u53

Self-Attention & Bidirectional LSTM: Activity Plots

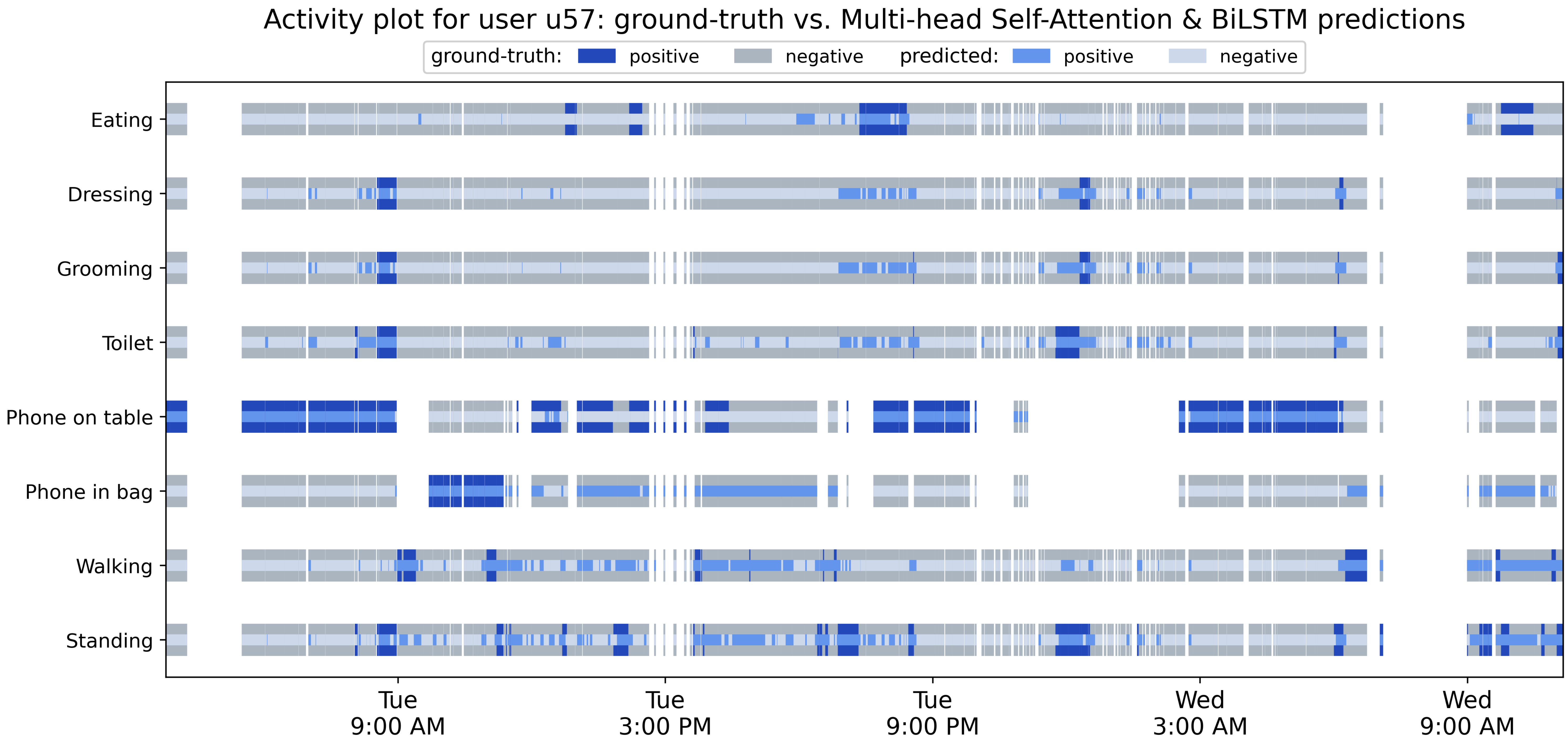
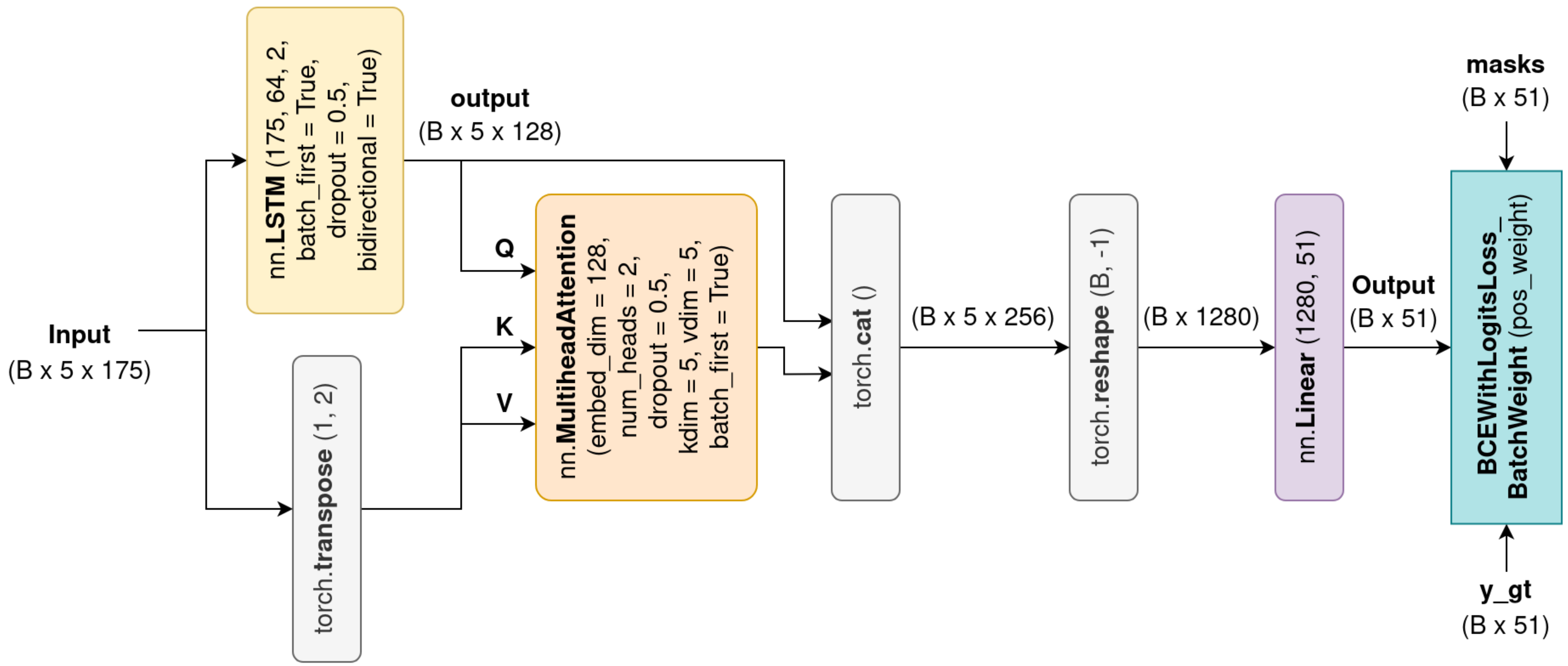


Figure 7: Activity plot including the predictions of the Multi-head Self-Attention & BiLSTM model using final hidden states, for user u57

Bidirectional LSTM & Cross-Attention



Input sequence length: **5**
BiLSTM number of layers: **2**
BiLSTM hidden size: **64**
BiLSTM dropout: **0.5**
Attention heads: **2**
Attention dropout: **0.2**
Learning rate: **0.00005**

Trainable parameters: 322611
- BiLSTM: 222720
- Cross-Attention: 34560
- Linear layer: 65331

Figure 8: BiLSTM & features' Cross-Attention model architecture, where the BiLSTM output for all timesteps is used to produce the query and the input features are used to produce key and value in the Cross-Attention

Accuracy	Precision	Sensitivity	Specificity	F1-score	BA
0.800	0.238	0.767	0.801	0.309	0.784

Table 6: Recognition scores of the BiLSTM & Cross-Attention model, averaged for all labels

Bidirectional LSTM & Cross-Attention: Interpretability

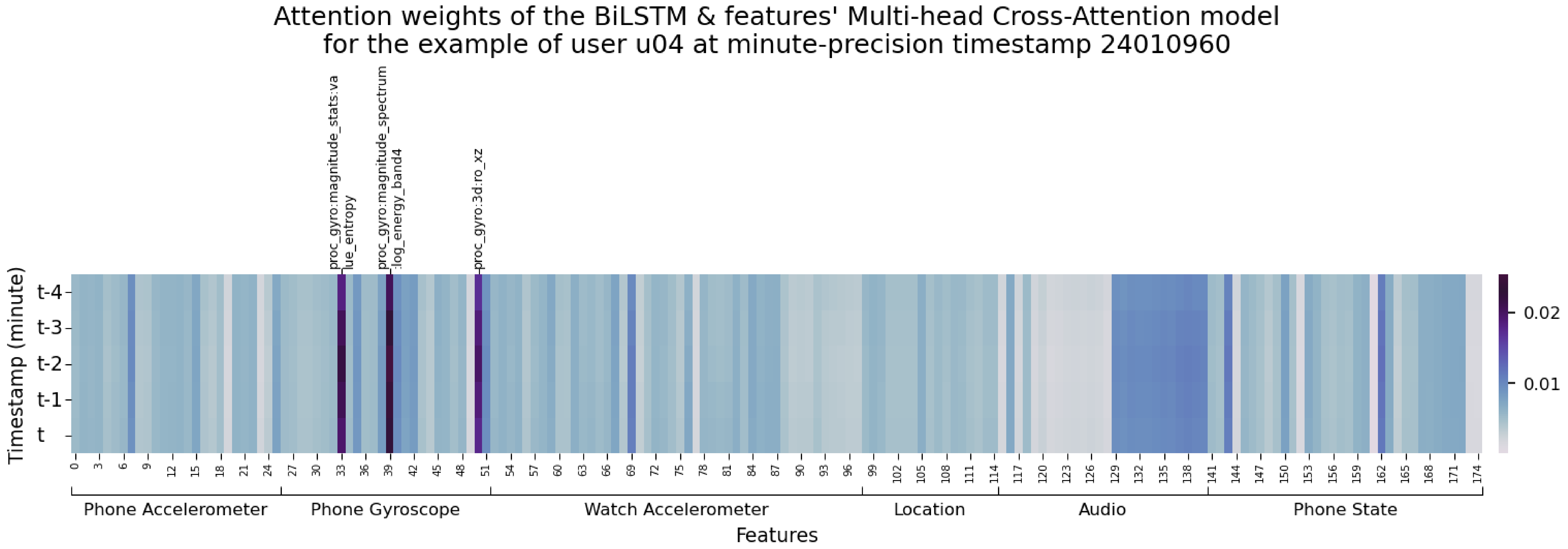


Figure 9: Attention weights of the BiLSTM & features' Cross-Attention model for u04 and t24010960
Ground-truth labels: Sitting, Indoors, At home, Computer work, Phone on table
Predicted labels: Sitting, Indoors, At home, Surfing the internet, Computer work, Eating, Phone on table

Bidirectional LSTM & Cross-Attention: Interpretability

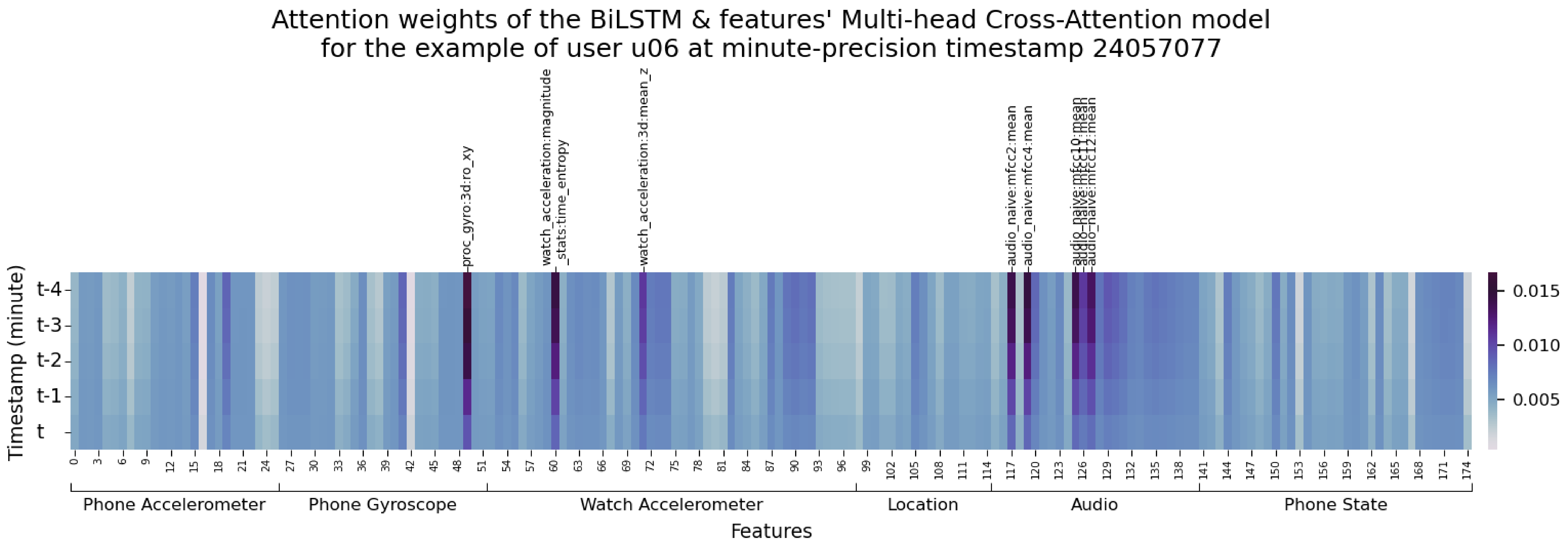


Figure 10: Attention weights of the BiLSTM & features' Cross-Attention model for u06 and t24057077
Ground-truth labels: Lying down, Sleeping, Indoors, At home, Phone on table
Predicted labels: Lying down, Sleeping, Indoors, At home, Phone on table

Bidirectional LSTM & Cross-Attention: Interpretability

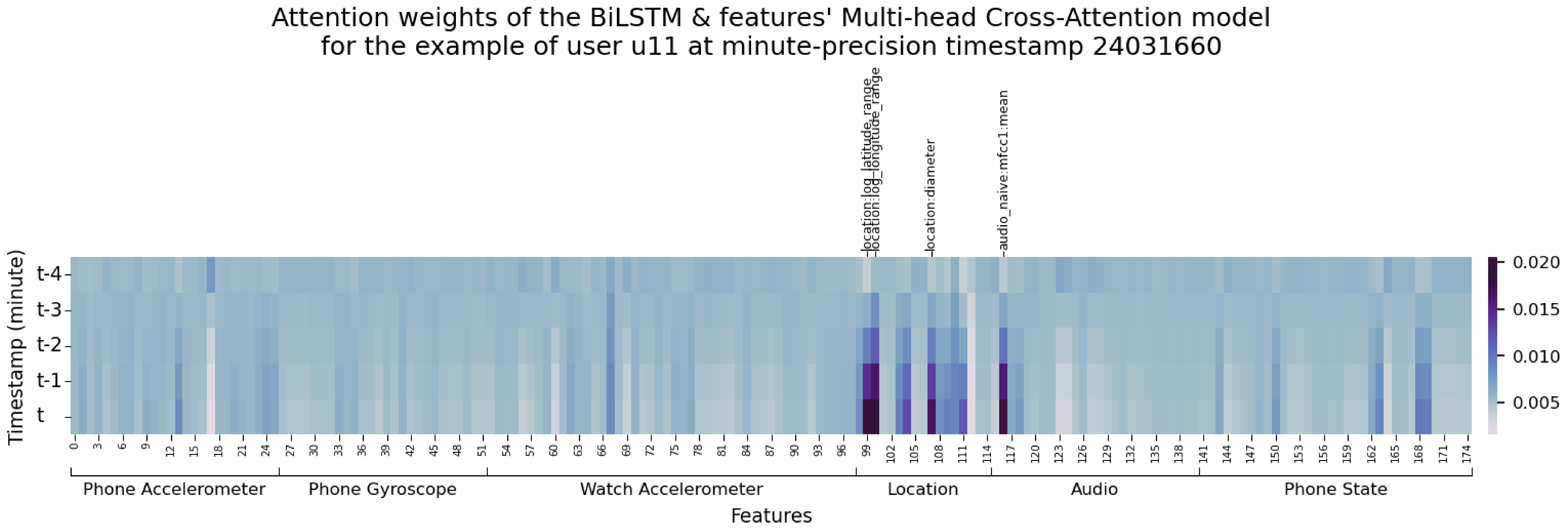
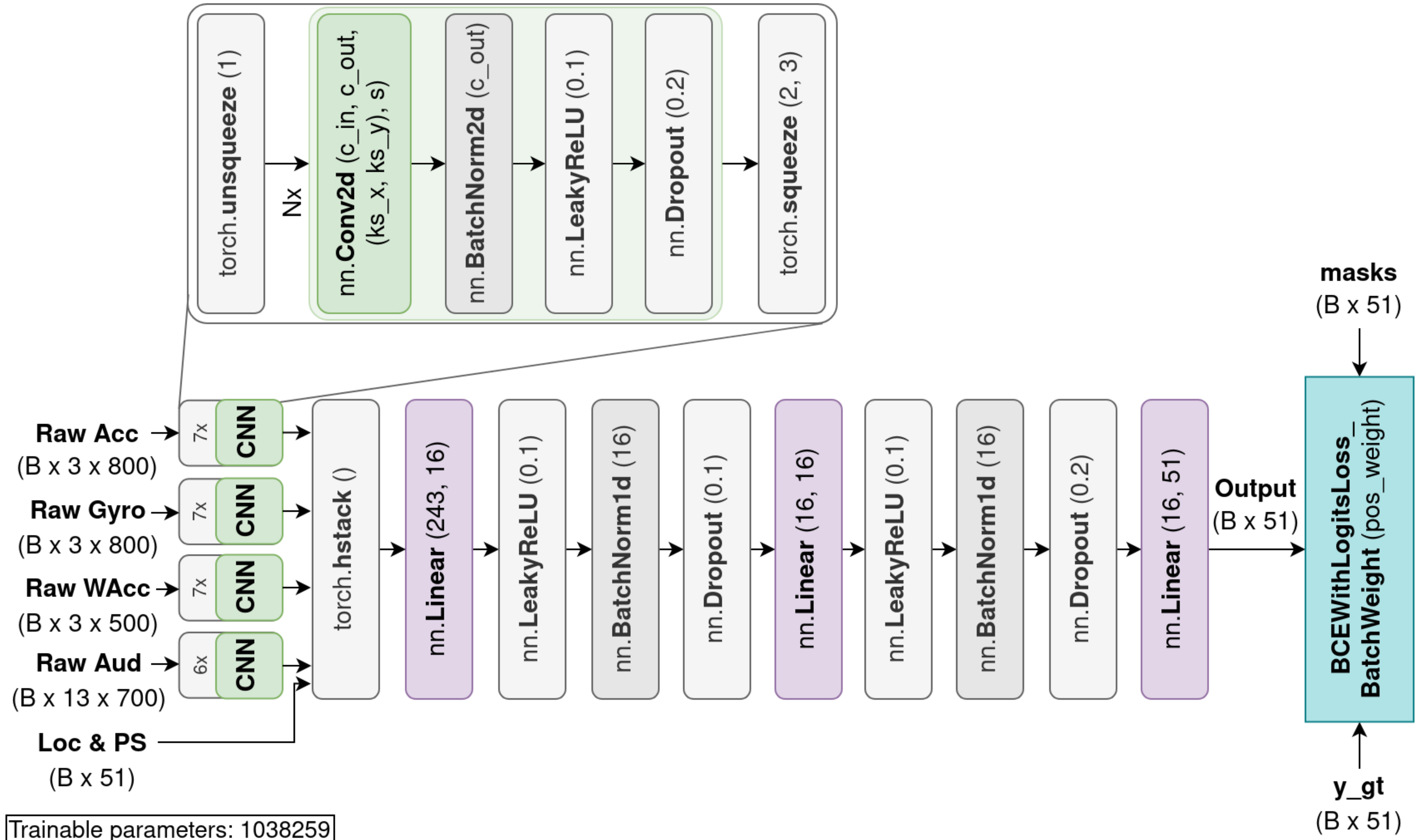


Figure 11: Attention weights of the BiLSTM & features' Cross-Attention model for u11 and t24031660
Ground-truth labels: Sitting, In a car
Predicted labels: Sitting, Outside, In a car, On a bus, Drive - Driver, Drive - Passenger, Phone in pocket, Shopping, At a party, At the beach, Phone in hand, Phone in bag, With friends

CNN-based model



Trainable parameters: 1038259
- CNN modules: 1033152
- Linear layers: 5043
- LL Batch Norm: 64

Figure 12: CNN-based model architecture

Conv layers' number of output channels:
{Acc: 32, Gyro: 32, WAcc: 64, Aud: 64}
CNN dropout: **0.2**
MLP hidden size: **(16, 16)**
MLP dropout: **(0.1, 0.2)**
Learning rate: **0.0005**

Accuracy	Precision	Sensitivity	Specificity	F1-score	BA
0.782	0.228	0.762	0.781	0.296	0.772

Table 7: Recognition scores of the CNN-based model, averaged for all labels

CNN-Transformer model

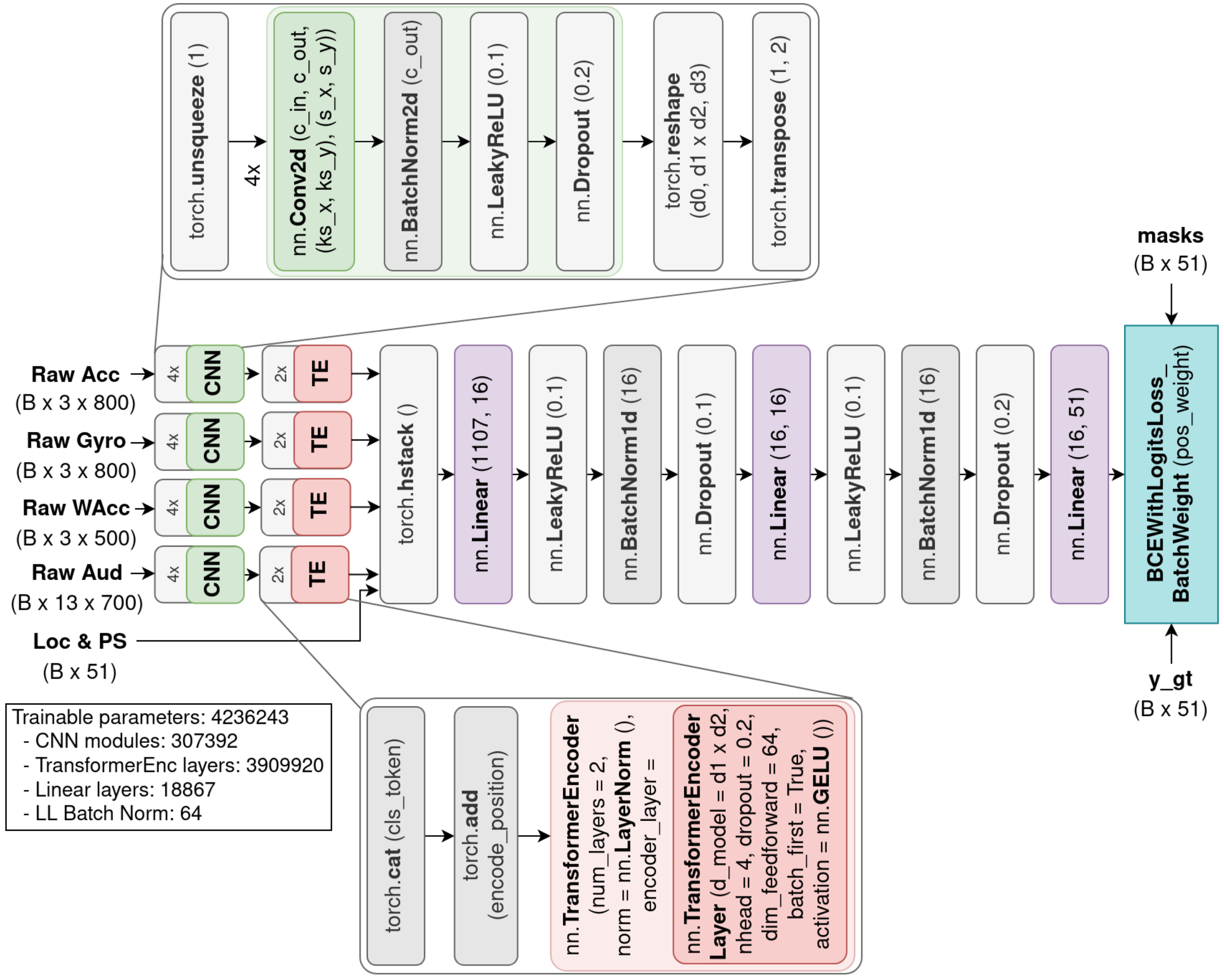


Figure 13: CNN-Transformer model architecture

Conv layers' number of output channels:
{Acc: 48, Gyro: 48, WAcc: 48, Aud: 48}
CNN dropout: **0.2**
Transformer Encoder number of layers: **2**
Transformer Encoder layers' Attention heads: **4**
Transformer Encoder layers' Feedforward dim: **64**
Transformer Encoder layers' dropout: **0.2**
MLP hidden size: **(16, 16)**
MLP dropout: **(0.1, 0.2)**
Learning rate: **0.0005**

Accuracy	Precision	Sensitivity	Specificity	F1-score	BA
0.792	0.229	0.759	0.790	0.300	0.774

Table 8: Recognition scores of the CNN-Transformer model, averaged for all labels

CNN-BiLSTM model

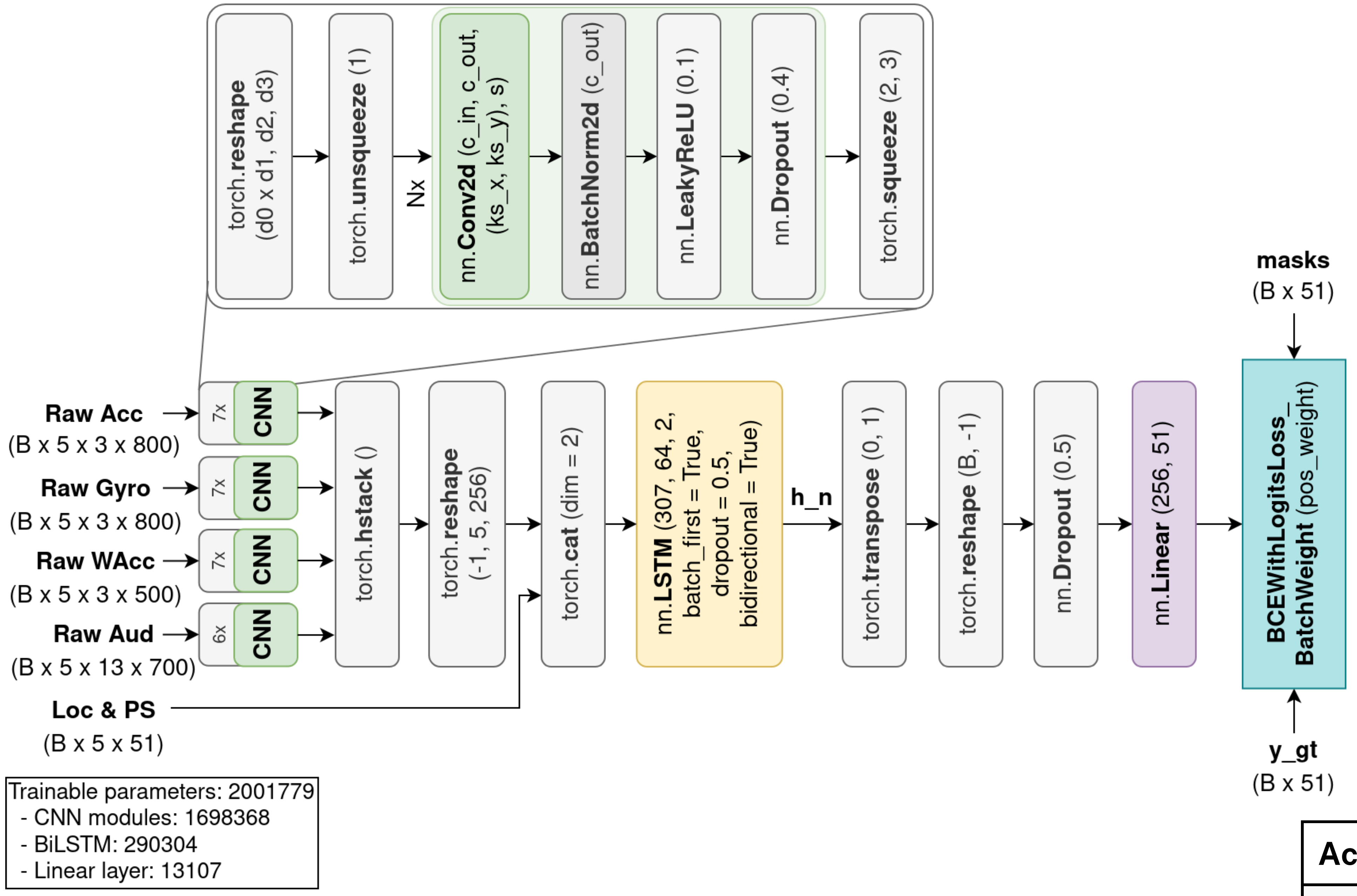


Figure 14: CNN-BiLSTM model architecture

Input sequence length: **5**
Time-Distributed CNN
Conv layers' number of output channels:
{Acc: 64, Gyro: 64, WAcc: 64, Aud: 64}
CNN dropout: **0.4**
BiLSTM number of layers: **2**
BiLSTM hidden size: **64**
BiLSTM dropout: **0.5**
Learning rate: **0.00005**

Accuracy	Precision	Sensitivity	Specificity	F1-score	BA
0.819	0.241	0.728	0.821	0.313	0.775

Table 9: Recognition scores of the CNN-BiLSTM model, averaged for all labels

Results Overview

Comparative results overview of the performance metrics averaged over all labels for each models's best-performing configuration								
Input	**Time-series modeling**	**Model**	**Accuracy**	**Precision**	**Sensitivity**	**Specificity**	**F1-score**	**BA**
		Random classifier	0.500	0.110	0.500	0.500	0.137	0.500
		Majority class classifier	**0.915**	NaN	0.040	**0.958**	0.037	0.499
Extracted features	Single example	Logistic Regression [VWL18]	**0.832**	-	0.597	**0.838**	-	0.718
		Logistic Regression	**0.839**	**0.246**	0.612	**0.844**	**0.314**	0.728
		MLP [VWL18]	0.773	-	**0.773**	0.773	-	0.773
		MLP	0.786	0.228	0.757	0.786	0.298	0.772
Extracted features	Sequence of examples	BiLSTM (last output)	0.813	**0.243**	0.753	**0.814**	**0.316**	**0.784**
		BiLSTM (all outputs)	0.810	**0.241**	0.761	**0.811**	**0.314**	**0.786**
		Self-Attention & BiLSTM	**0.818**	**0.248**	0.756	**0.819**	**0.323**	**0.788**
		BiLSTM & Cross-Attention	0.800	0.238	**0.767**	0.801	0.309	**0.784**
Raw data	Single example	CNN	0.782	0.228	0.762	0.781	0.296	0.772
		CNN & Transformer	0.792	0.229	0.759	0.790	0.300	0.774
Raw data	Sequence of examples	CNN & BiLSTM	**0.819**	**0.241**	0.728	**0.821**	**0.313**	0.775

Table 10: An overview of the recognition scores of all models, averaged for all labels

Conclusions

- Regarding the models using the **extracted features** for a **sequence of examples**:
  - **BiLSTM** and **BiLSTM & Attention** models produce constantly **better** results compared to the baselines, when using a **5-length** examples sequence.
  - The **Self-Attention & BiLSTM** model produces the best results overall.

- Regarding the models using the **raw sensor data**, we did not manage to produce results better than the baseline models.
  - **More hyperparameter tuning** might be required.
  - Deep learning based feature extraction might **not always be the best solution** for HAR when testing on unseen users or out-of-domain data [Ben+22].

5

[Ben+22] Bento, N. et al. “Comparing Handcrafted Features and Deep Neural Representations for Domain Generalization in Human Activity Recognition”. In: Sensors 22.19 (2022). issn: 1424-8220. doi: 10.3390/s22197324.

Conclusions

- Regarding **improvements** in **individual metrics**:
  - Adding a **Cross-Attention** mechanism after the BiLSTM produced the larger improvement in **Sensitivity**.
  - When using a **BiLSTM** to model a **sequence of input examples**, we consistently got higher **Specificity** values.

- Regarding **individual labels**, the labels with the **higher recognition metrics** are:
  - labels with **a lot of positive examples** in the dataset
  - labels **less prone to be mislabeled** by users
  - labels that correspond to activities with **small variability**
  - labels that are **suited** to be predicted using the **specific set of sensors**

5

Conclusions

- The **improvements** we have achieved are relatively **small.**
  - The task is **inherently flawed** because of the dataset's imperfections and the **margin for improvement** might be relatively **small** by default.
  - A **radical change** in our **approach** to the specific HAR task is required to produce greater improvements in the recognition metrics.

- Finally, we should note that all our models and experiments are included in the Github **repository** alexvioni/ExtraSensory-functionality (temporarily private, will be opened), which is a **flexible** and **ready-to-use** codebase for HAR.

5

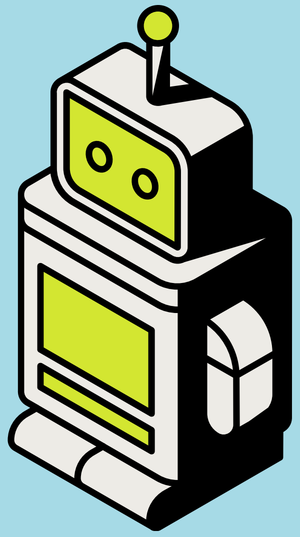
Directions for Future Work

- Activity taxonomies & mutually exclusive labels
- Model personalization
- Unsupervised or semi-supervised methods for label confidence
- More examples for rare labels
- Data representations from self-supervised models
- Activity recognition on other datasets, e.g., e-Prevention [Zla+22]
- Real-life use: shorter sensor recording segments and improved Accuracy

5

[Zla+22] Zlatintsi, A. et al. "E-Prevention: Advanced Support System for Monitoring and Relapse Prevention in Patients with Psychotic Disorders Analyzing Long-Term Multimodal Data from Wearables and Video Captures". In: Sensors 22.19 (2022). issn: 1424-8220. doi: 10 . 3390/s22197544.

Thank you



Appendix - Features

- **Smartphone Accelerometer and Gyroscope (26 features each):**
  - **statistics of the magnitude signal** (mean, standard deviation, third moment, fourth moment, 25th percentile, 50th percentile, 75th percentile, value-entropy, time-entropy)
  - **spectral features of the magnitude signal** (log energies in 5 sub-bands: 0–0.5Hz, 0.5–1Hz, 1–3Hz, 3–5Hz, > 5Hz) and spectral entropy
  - **two autocorrelation features from the magnitude signal**
  - **statistics of the 3-axis time series** (mean and standard deviation of each axis and the 3 inter-axis correlation coefficients)
- **Watch Accelerometer (46 features):**
  - **the features described above for the Smartphone Accelerometer**
  - **spectral features of the 3-axis time series** (log energies in 5 sub-bands: 0–0.5Hz, 0.5–1Hz, 1–3Hz, 3–5Hz, > 5Hz)
  - **five relative-direction features** (the cosine-similarity between the acceleration directions of any two time points in the time series is calculated and then these cosine similarity values are averaged in 5 different ranges of time-lag between the compared time points: 0–0.5sec, 0.5–1sec, 1–5sec, 5–10sec, > 10sec)
- **Location (17 features):**
  - **coordinates-derived features:** standard deviation of latitude, standard deviation of longitude, change in latitude, change in longitude, average absolute value of derivative of latitude and average absolute value of derivative of longitude, number of updates, log of latitude-range, log of longitude-range, minimum altitude, maximum altitude, minimum speed, maximum speed, best vertical accuracy, best horizontal accuracy and diameter

5

Appendix - Features

- **Audio (26 features):**
  - **statistics of the MFCC time series** (mean and standard deviation of each of the 13 coefficients)
- **Phone State (34 features - one-hot representation):**
  - **app state** (active, inactive, background, missing)
  - **battery plugged** (AC, USB, wireless, missing)
  - **battery state** (unknown, unplugged, not charging, discharging, charging, full, missing)
  - **in a phone call** (false, true, missing)
  - **ringer mode** (normal, silent no vibrate, silent with vibrate, missing)
  - **WiFi status** (not reachable, reachable via WiFi, reachable via WWAN, missing)
  - **time-of-day** (eight half-overlapping time ranges: midnight-6am, 3am-9am, 6am-midday, 9am-3pm, midday-6pm, 3pm-9pm, 6pm-midnight and 9pm-3am)

5