**Abstract**

*In this project, we aim to classify EEG data using convolutional neural networks (CNNs) and recurrent neural networks (RNNs), and hybrids of the two, into one of four classes (moving right hand, left hand, both legs, or tongue). The data consists of 22 channels taken at 1000 time steps and 9 labels, 4 of which we wish to classify. We investigate different RNNs, and hybrids of CNNs and RNNs, and are able to achieve an accuracy of 73%.*

# **Introduction**

An electroencephalogram (EEG) is a non-invasive procedure for obtaining brain activity. This paper aims to classify EEG data from four motor imagery tasks (imagination of different movements). The first neural network architecture we decided to pursue was a CNN. CNNs have become increasingly prevalent in computer vision tasks since the nature of their convolution and pooling layers allows for the network to routinely and adaptively learn spatial features in an image. With the introduction of temporal filters in the architecture, CNNs also become capable of learning and classifying visual time-series data. Hence, we believed that using a CNN with both spatial and temporal filters would allow us to capture visual nuances and hierarchies over time in the EEG data and achieve a high classification accuracy.

Another class of neural networks we implemented to classify the EEG data was the RNN. Since our dataset contains temporal information about the EEGs, we believed that it would be most appropriate to introduce architectures that would be able to fully capture the dynamic nature of the data and explore patterns and connectivity over time. As such, we explored two conventional RNN architectures: the long short-term memory (LSTM) model and the gated recurrent network (GRU) model.

To fully highlight the benefits of both CNNs and RNNs, we also implemented two hybrid architectures: CNN+LSTM and CNN+GRU. By doing so, we hope to more accurately capture spatial data through the CNN while also grasping recurrency and connectivity information over time with the RNNs.

# **Results**

* 1. **Classification for a single subject**

Before we focused on optimizing our models to achieve high classification accuracy, we deemed it necessary to delve deeper into the dataset itself. Namely, we wanted to determine whether including data from all nine subjects would allow us to achieve a higher classification accuracy than using data from just one subject. To do this, we trained a CNN with three batch-normalized convolutional layers, one dropout layer, and one fully connected layer on EEG data from just one subject (subject 1) and then trained another iteration of the same model but with EEG data from all nine subjects. We then evaluated the performance of the two models on the same validation dataset to see that the model trained on all subjects had a higher validation accuracy (70.9%) than the model trained on one subject (46.7%). From this, we ascertain that including data from all nine subjects is necessary to maximize the classification accuracy of the EEG data.

* 1. **Optimal classification time period**

Using data from all subjects and the CNN-LSTM model described in the next section, we tested different segments of the temporal data to optimize our testing accuracy. Our first probe trained the model ten times across 100 epochs, first using data from indices 0 to 100, 0 to 200, and so forth across the 1000 time steps. We found that max testing accuracy improved dramatically using the window of data from 0 to 300 and remained relatively stagnant after that. We conclude that the information encoded between time steps 0 and 300, and 200 to 300 specifically, contain the most important information for our training. This was further corroborated by our second analysis, where we slid time windows of 100, 300, and 800 across the dataset. The optimal time slot of the length-100 windows was 0-100, with a significant drop off after 300. The optimal time slot of the length-300 windows was 0-300, with accuracy decreasing with each subsequent iteration. The optimal time slot of the length-800 windows was 0-800, again decreasing with each subsequent iteration.

* 1. **Overall Classification**

We then tried to optimize classification accuracy across all subjects. Our base convolution architecture took inspiration from Chen et al.[1]. We note that an architecture consisting of convolutional layers and an LSTM layer (72.7%) or GRU layer (73%) This significantly outperforms other architectures we tried, such as a pure CNN (70%), pure LSTM (35%), and pure GRU models (28%).

# **Discussion**

* 1. **Model Architectures**

Convolutional neural networks have been great in computer vision as they allow a model to learn spatial features that allow the model to classify images.

In our CNN\_LSTM model, we used the EEGNeX model developed by Chen et al.[1], involving two convolution layers of expanding filter depth, a depthwise convolution layer, and the requisite batch-normalization and dropouts in each block. Along with a weight decay of 5e-4, this baseline architecture gave us the best results of any that we tried.

However, CNNs are still essentially a feed-forward network, and as such is limited in learning any sort of sequence and temporal dependence in the data. Hence we turn to recurrent neural networks.

Both LSTM (Long Short-Term Memory) and GRU (Gated Recurrent Unit) are popular RNN architectures for modeling time series data, as they are designed to handle the vanishing gradient problem that can occur in traditional RNNs.

LSTMs are known for their ability to handle long-term dependencies in data. They have a more complex structure than GRUs, with additional memory cells and gates that control the flow of information. This allows LSTMs to capture more complex patterns in the data, especially when the data has long-term temporal dependencies.

GRUs, on the other hand, have a simpler structure with fewer parameters, which can make them easier to train and faster to converge. They also have fewer gates than LSTMs, which can reduce the risk of overfitting, especially when working with smaller datasets. As such, GRUs have a more difficult time in learning complex temporal patterns in long-term dependencies. This result is evident in our model results, where CNN+LSTM achieved 72.7% test accuracy, and CNN+GRU achieved only 73% test accuracy.

We also fed the output of each stacked RNN model architecture into a fully connected layer, with an exponential leaky unit (ELU) as our activation after each layer.

Stacked RNN architectures perform better than single LSTM or GRU on their own, as more layers make the model deeper and allow the model to learn more complex features.

Stacked CNN and RNN architectures, from our intuition then, would be best suited for EEG classification, as this architecture is robust to spatial features as well as temporal features. This is confirmed by our model results, in particular with pure LSTM and pure GRUs producing large overfit and producing quite, respectfully, horrendously.

* 1. **Dilation**

Dilation is a technique in CNNs that skips some of the convolutions. Dilated convolution helps expand the area of the input image covered without pooling. This increases our receptive field without increasing computational cost. Dilation also helps to maintain order in the data, which is important for EEGs. The skipping nature of CNNs adds a layer of regularization too, which allows us to train our model for many epochs without overfitting.

Before including dilation, our code was struggling to reach 67% test accuracy. Dilation seems to be a primary factor in boosting our accuracy to 72.7% in CNN\_LSTM and 73% in CNN\_GRU, and 69.7% in CNN.

* 1. **Classification of subjects**

Our results show that in order to classify the motor imagery of one subject, we get better accuracy if we train the model on data from all subjects. This makes sense, as data from all objects allows the model to see more data, and hence the model can generalize better. This is particularly important due to the noisiness of EEG data.

Further, EEG signals in other subjects of the same motor imagery does not differ significantly, as the movements the subjects are asked to imagine are general and would not be, by speculation, so specific to an individual that the movement of one subject would produce dramatically different data to another subject such that the model would not be able to distinguish.

1. **Appendix**
   1. **Classification Accuracy across models**

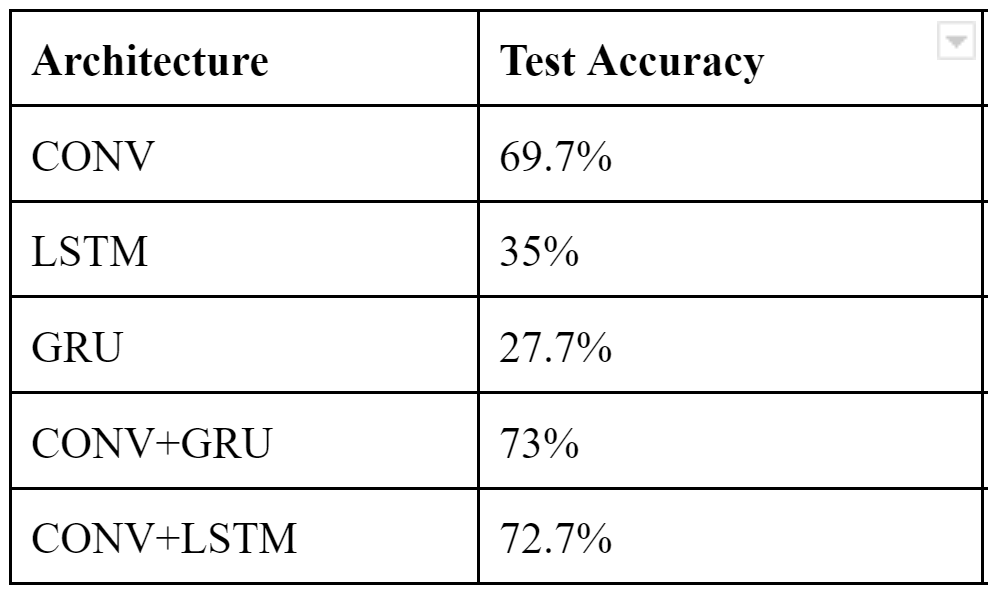


Figure 1: Accuracy comparison across our models when classifying all subjects.

* 1. **Classification accuracy graphs**

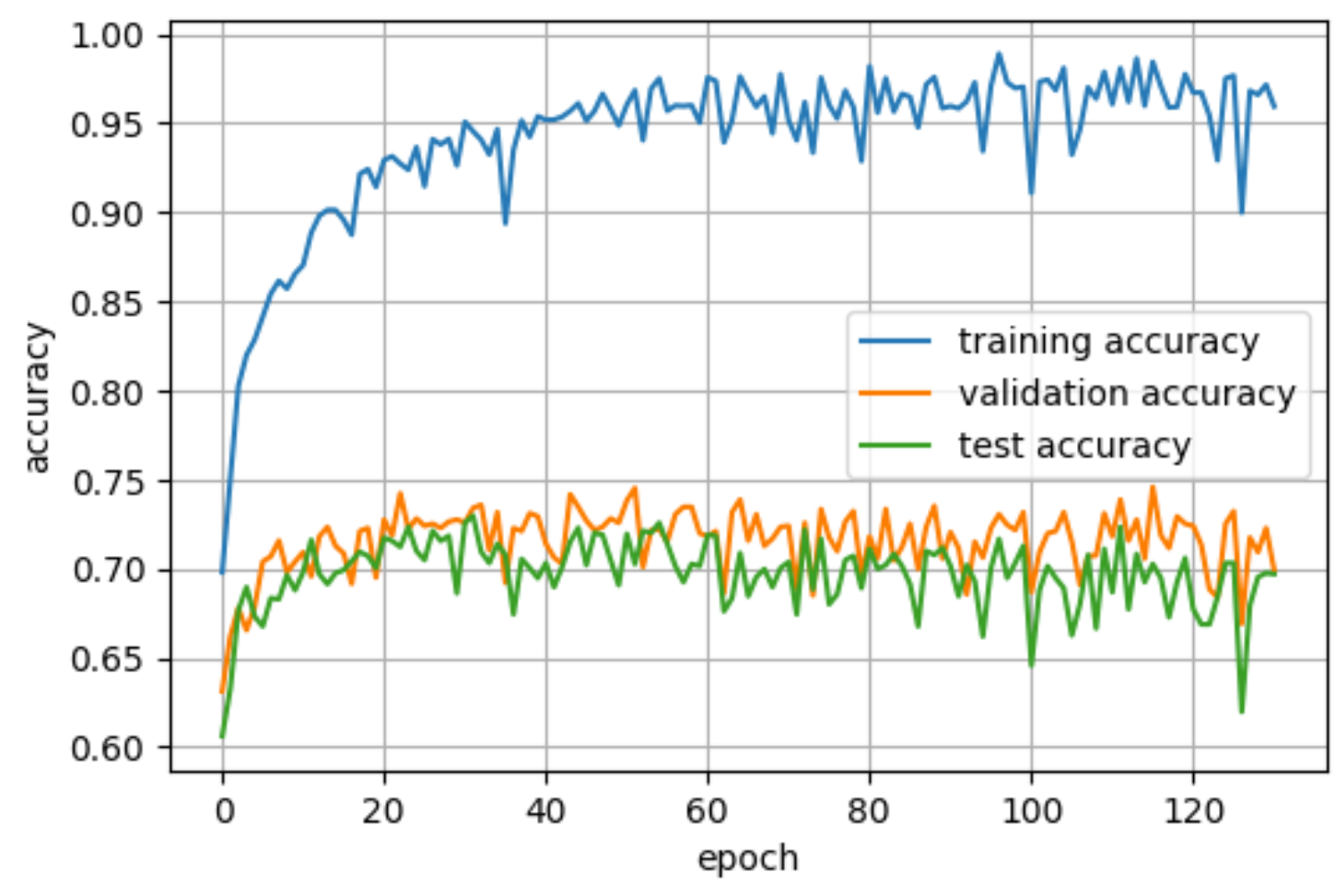


Figure 2: Train/Validation/Test accuracy for pure CNN

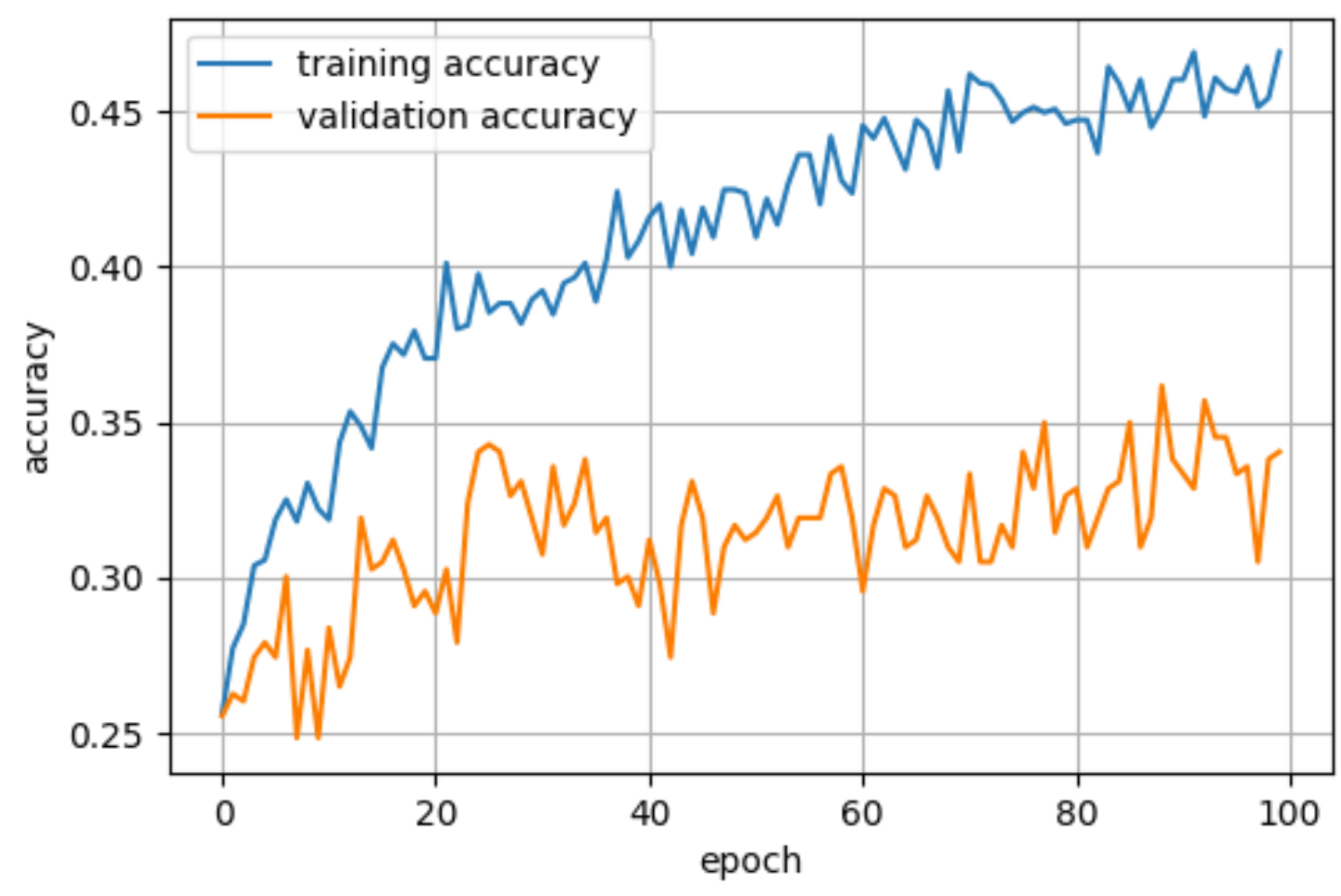


Figure 3: Train/Validation accuracy for pure LSTM

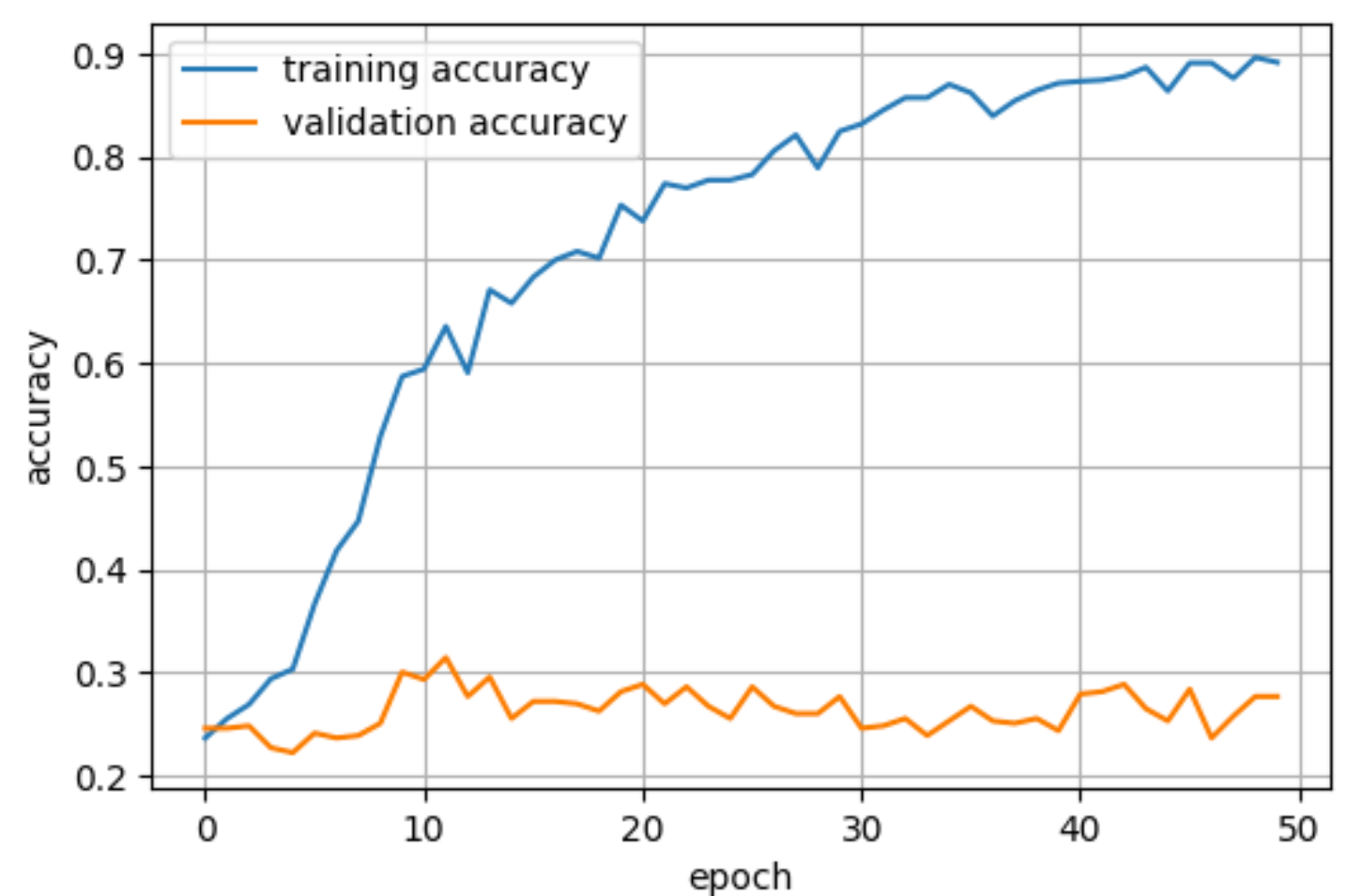


Figure 4: Train/validation accuracy for pure GRU

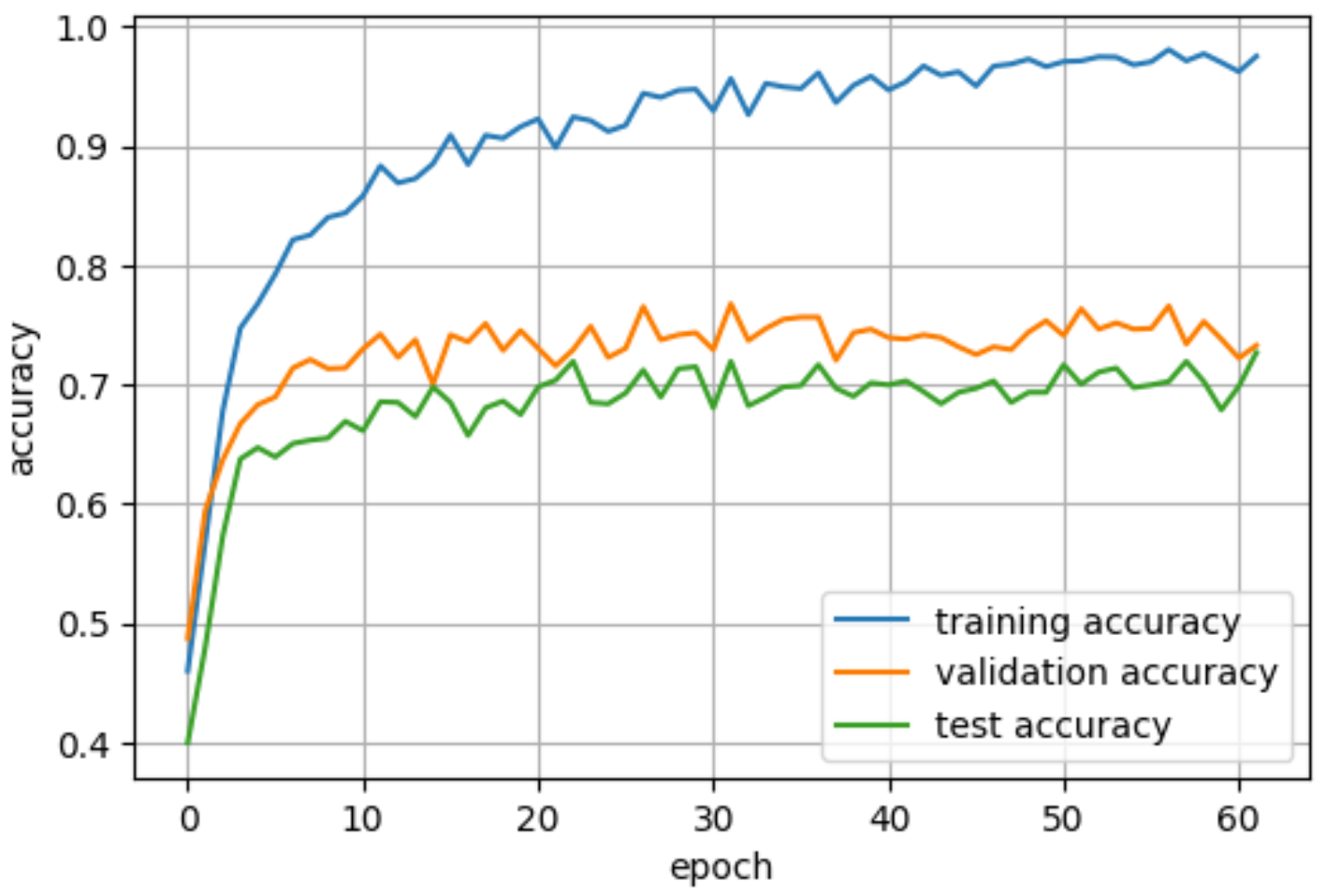


Figure 5: Train/Validation/Test accuracy for ConvLSTM

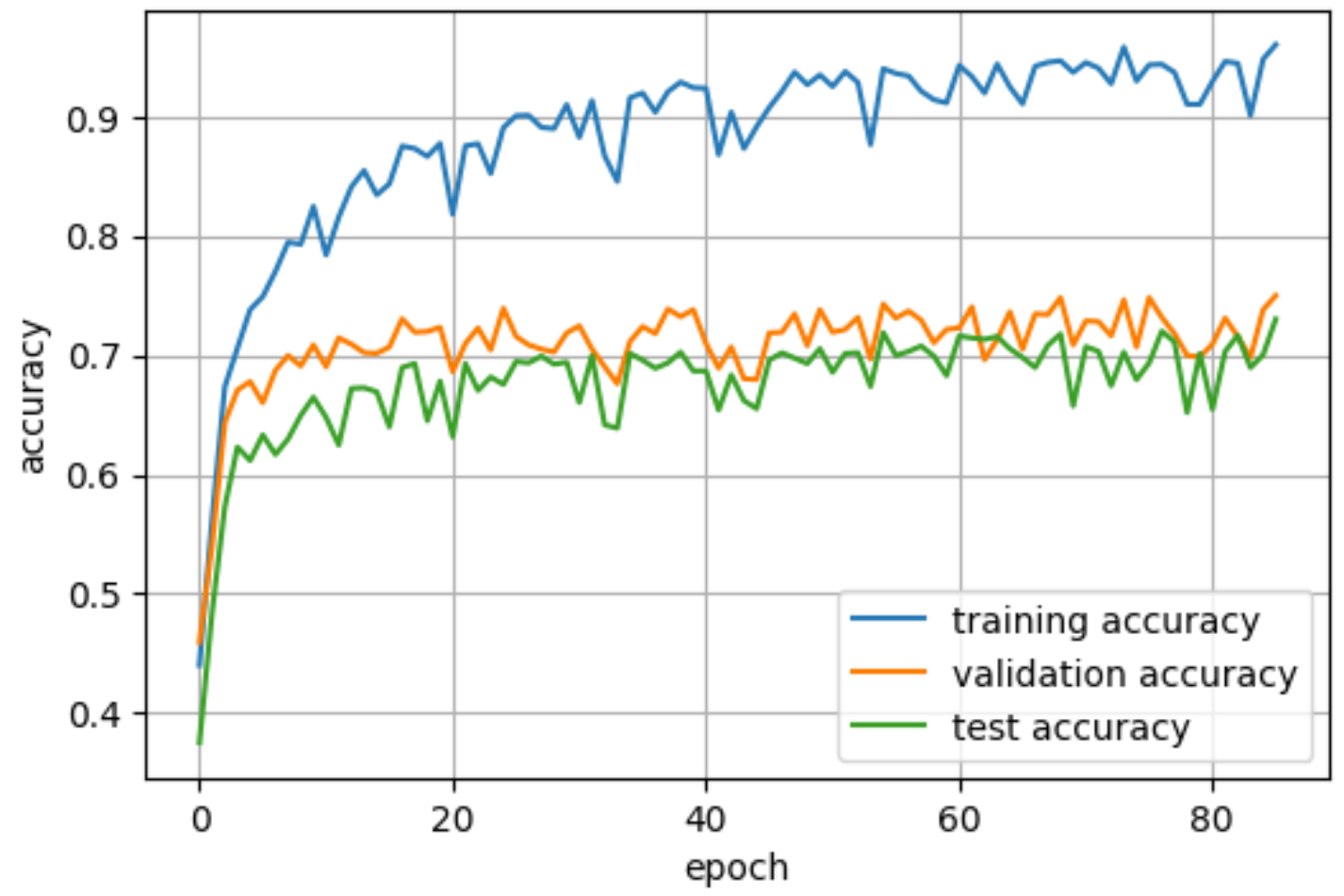


Figure 6: Train/Validation/Test accuracy for ConvGRU

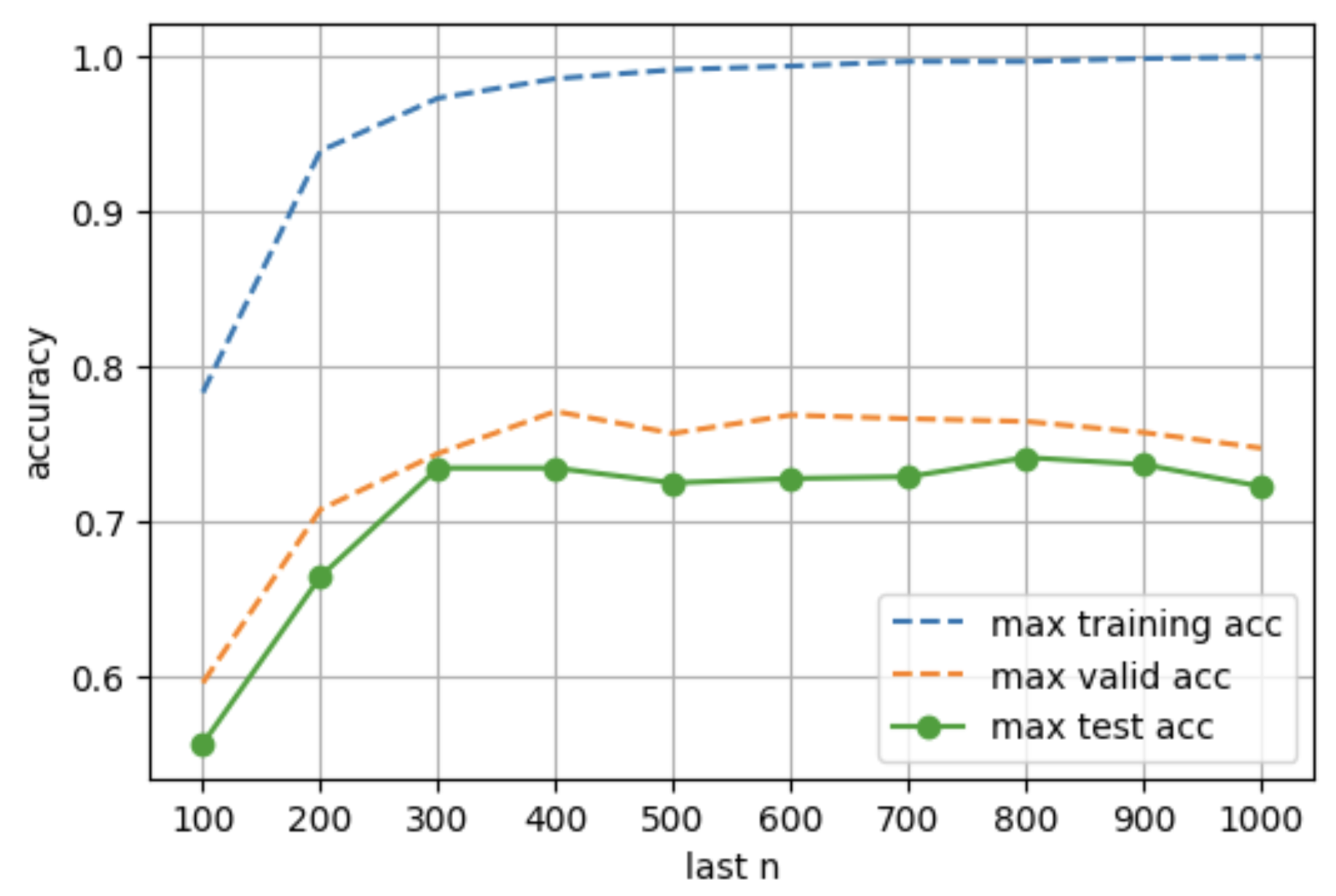
****

Figure 7: Train/Validation/Test accuracy for 0-100, 0-200, etc…

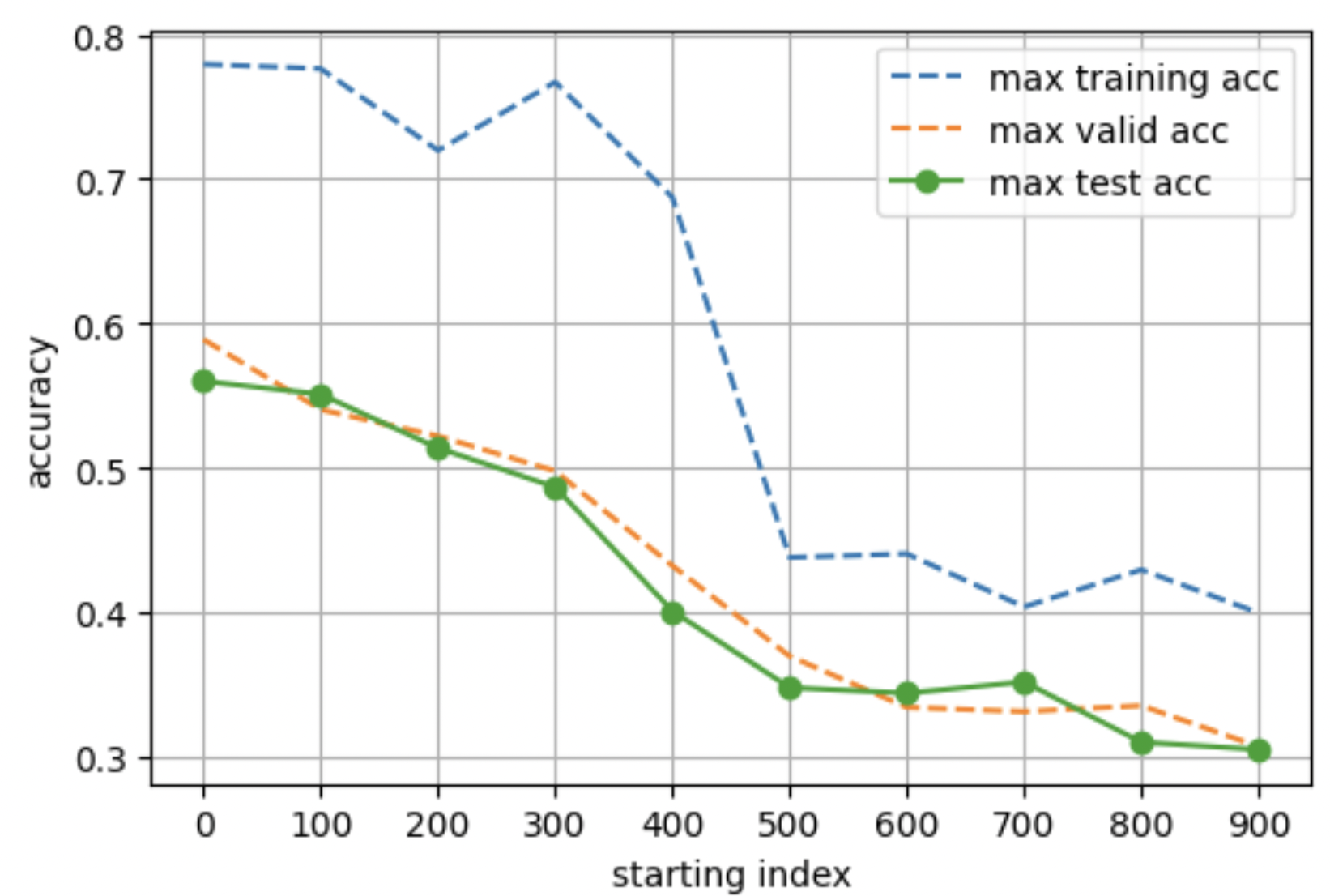
****

Figure 8: Train/Validation/Test accuracy for length-100 window

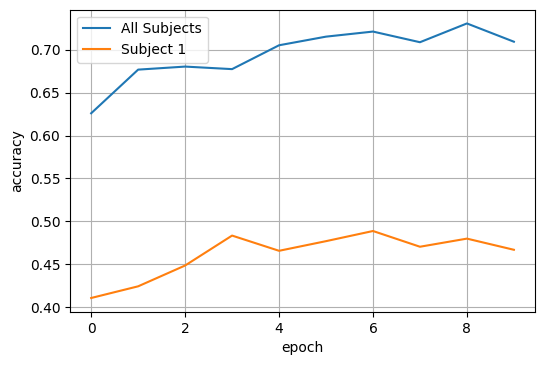


Figure 9: Validation accuracy for all subjects vs subject 1

* 1. **Model Architectures**

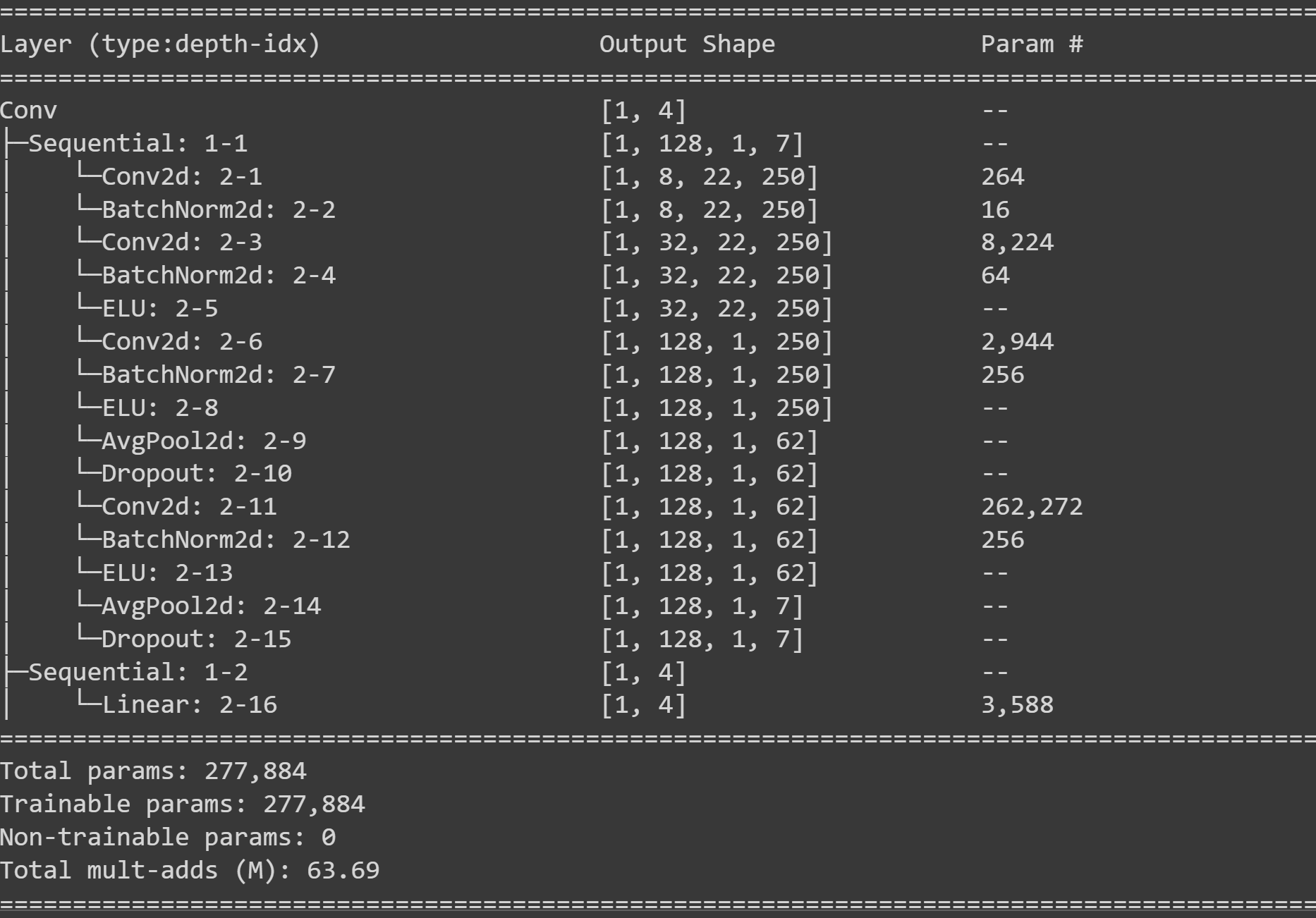


Figure 10: Architecture of pure CNN

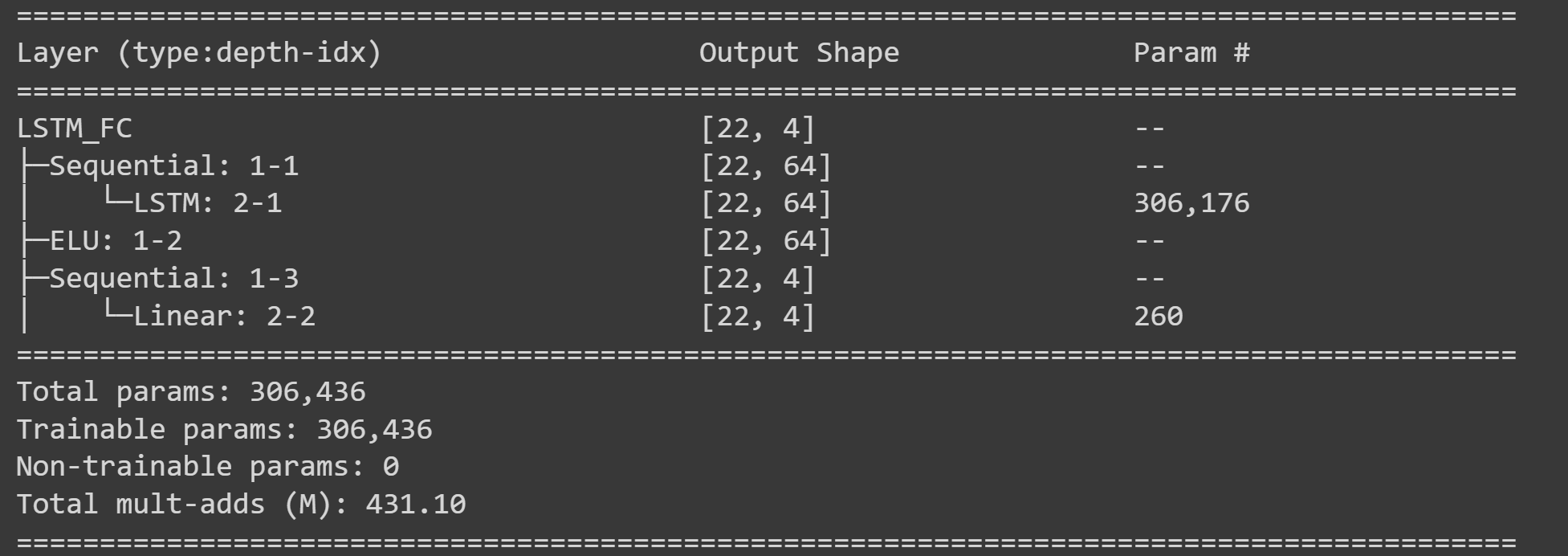


Figure 11: Architecture of pure LSTM

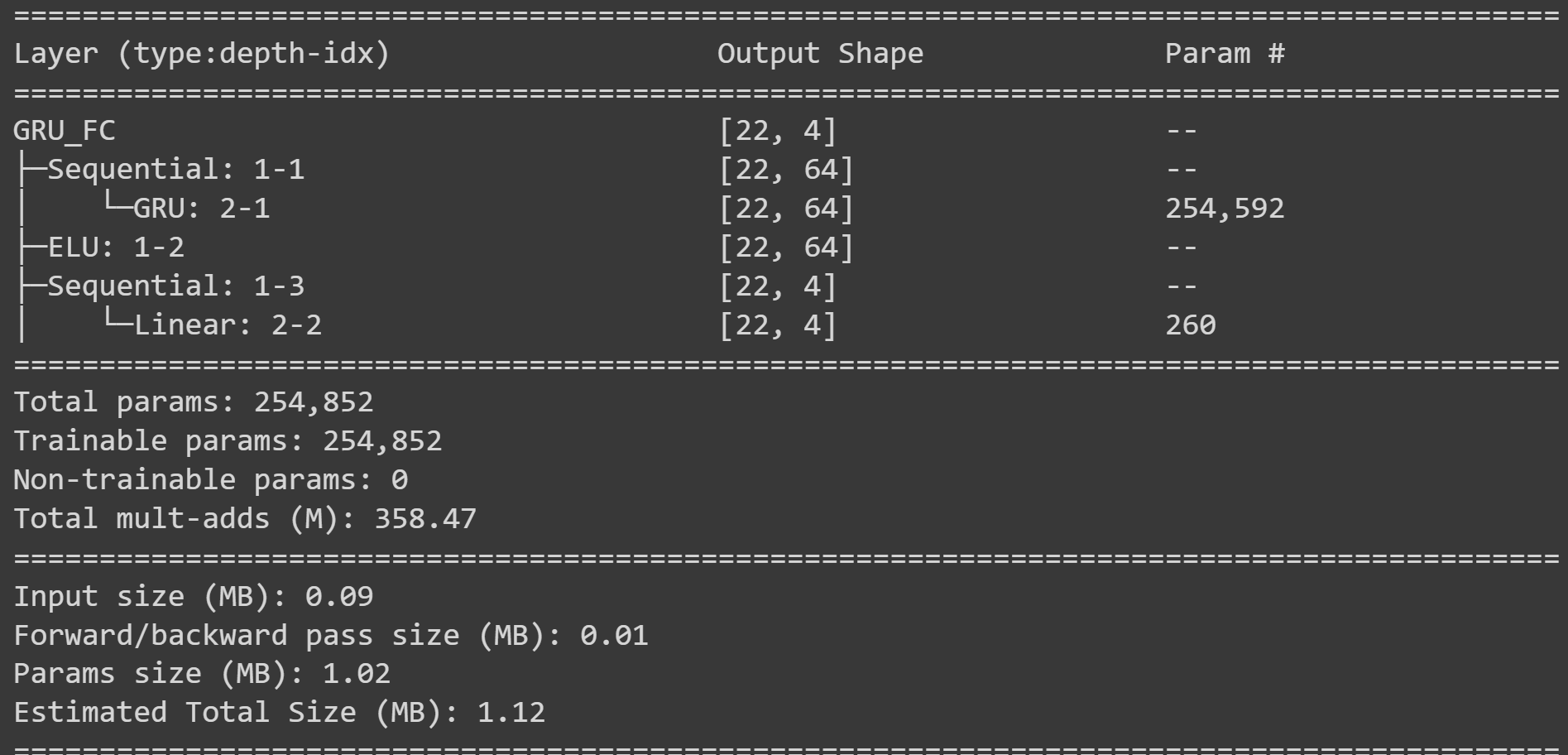


Figure 12: Architecture if pure GRU

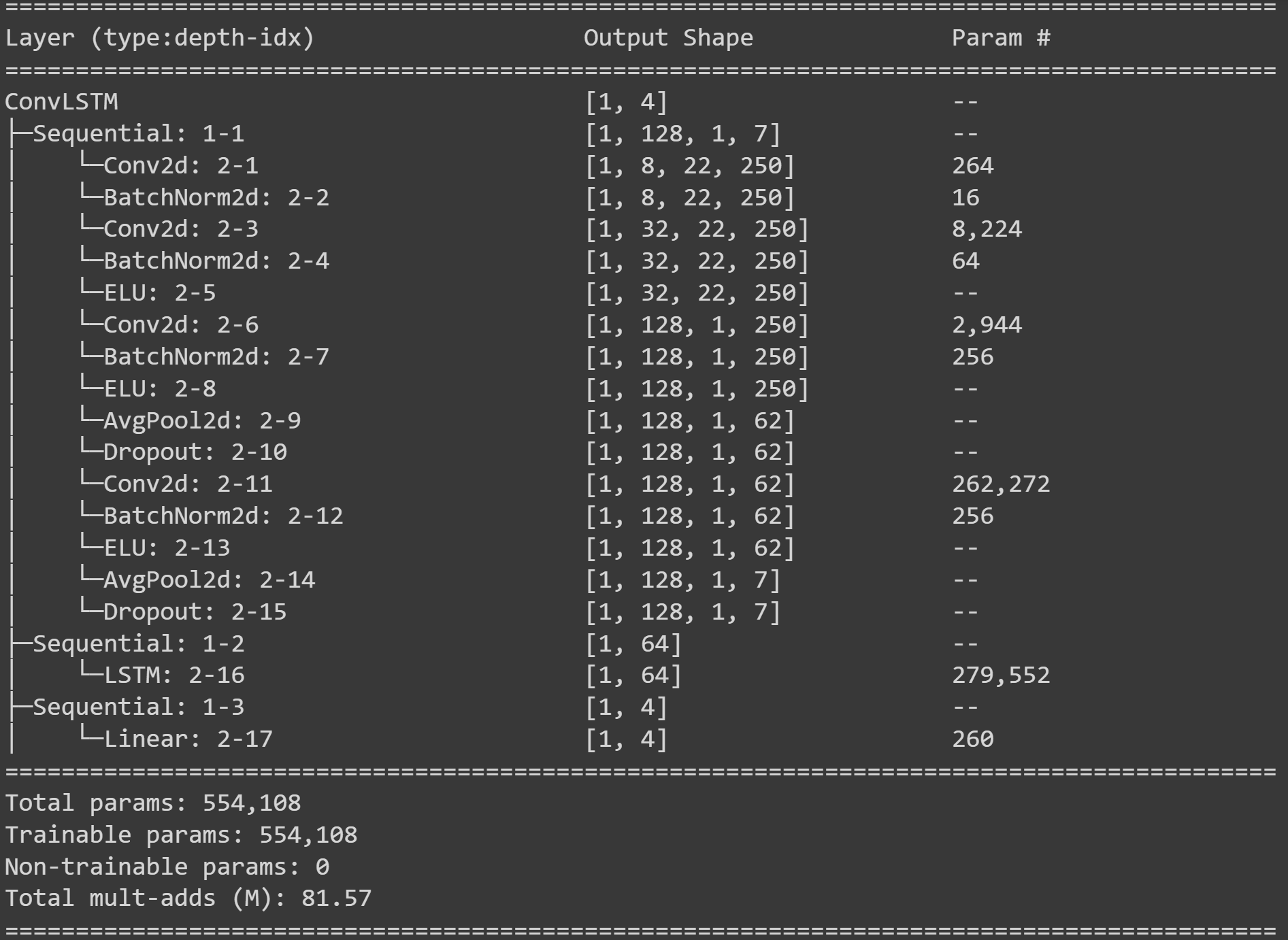


Figure 13: Architecture of ConvLSTM

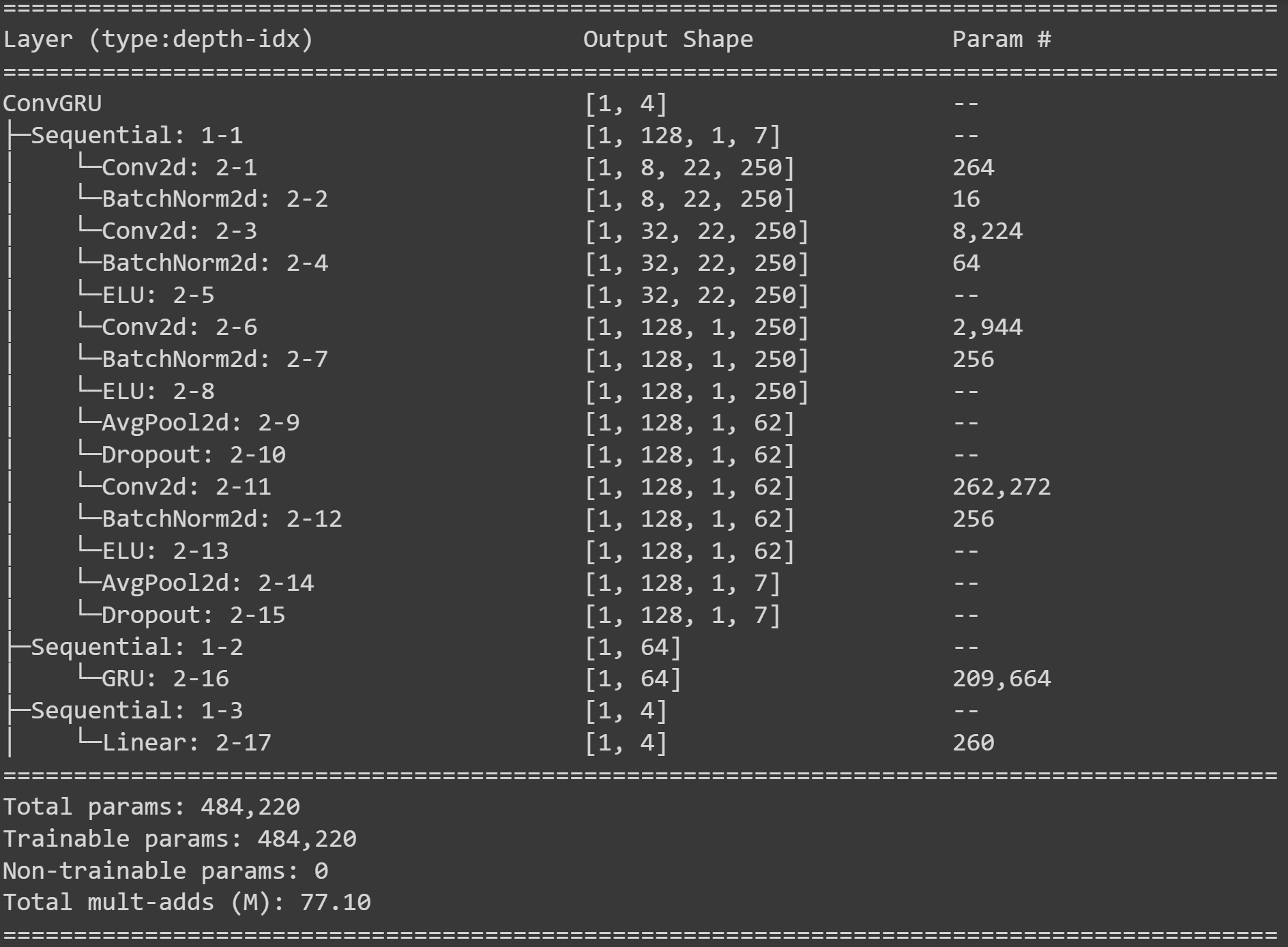


Figure 14: Architecture of ConvGRU

# **References**

1. [Chen, X., Teng, X., Chen, H., Pan, Y., & Geyer, P. (2022). Toward reliable signals decoding for electroencephalogram: A benchmark study to EEGNeX.](https://arxiv.org/abs/2207.12369)