Supplementary Material

January 17, 2025

Transformer-Based Predictive Model

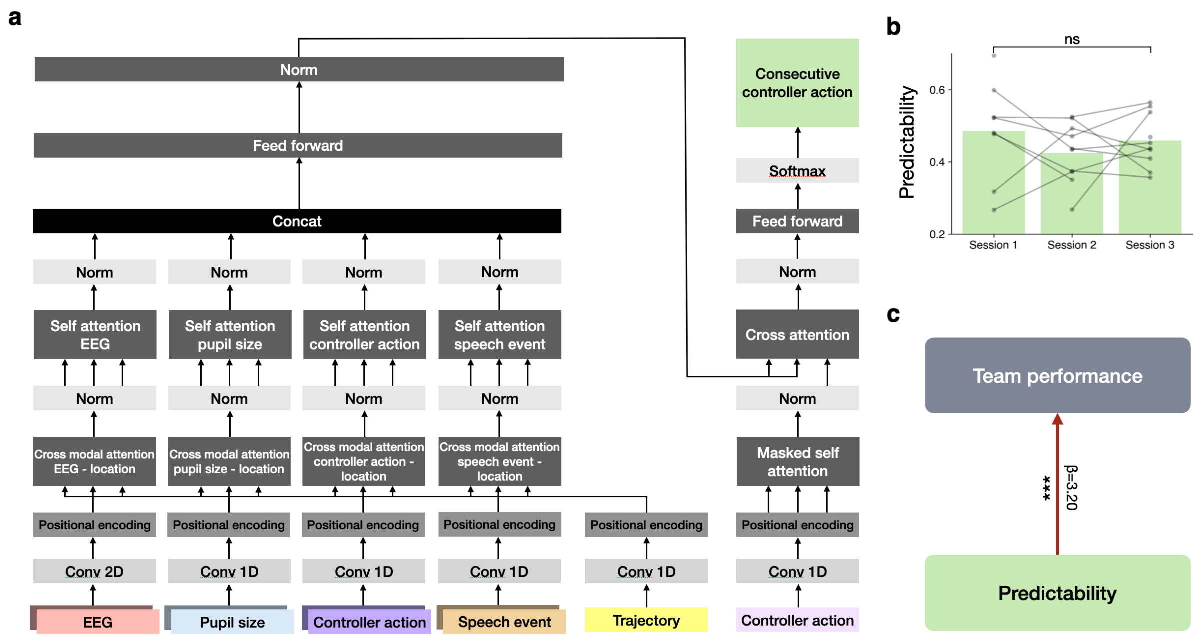
We developed a multi-head attention predictive model that processes multi-modal data to predict an individual team member's future remote controller actions. This model consists of two core components: an encoder that interprets the input data and a decoder that generates predictions of the remote controller actions based on the encoded information (Fig. 3 b in main text).

The model was trained and tested on recorded epochs of multi-modal data collected across experimental sessions, with the dataset divided into training (70%), validation (5%), and testing (25%) sets. To ensure comprehensive evaluation, we employed 4-fold cross-validation, systematically splitting the dataset such that each fold served as the test set once, collectively covering the entire dataset. Each split involved independent training and testing, allowing us to assess the model's performance across all data points while minimizing the risk of over-fitting.

For optimization, we employed the Adam optimizer. We assessed the model's performance by comparing its output against ground truth data of controller actions, using the Pearson Correlation Coefficient to quantify the accuracy of continuous predictions. Since the predicted actions are categorized into -1, 0, and 1, Cross Entropy Loss was utilized to measure classification performance. All models were trained for 100 epochs. Each model completed training in under 10 minutes. The training was executed on a single NVIDIA RTX A6000 GPU, utilizing CUDA version V12.2.140 and leveraging the capabilities of pytorch-lightning version 1.8.6 [1] and torch version 1.13.1 [2].

Model with Speech Event Input

To test whether speech events impact the predictability of team members' actions, we modified the model structure from main text (Fig. 3b) to include the speech events of two team members

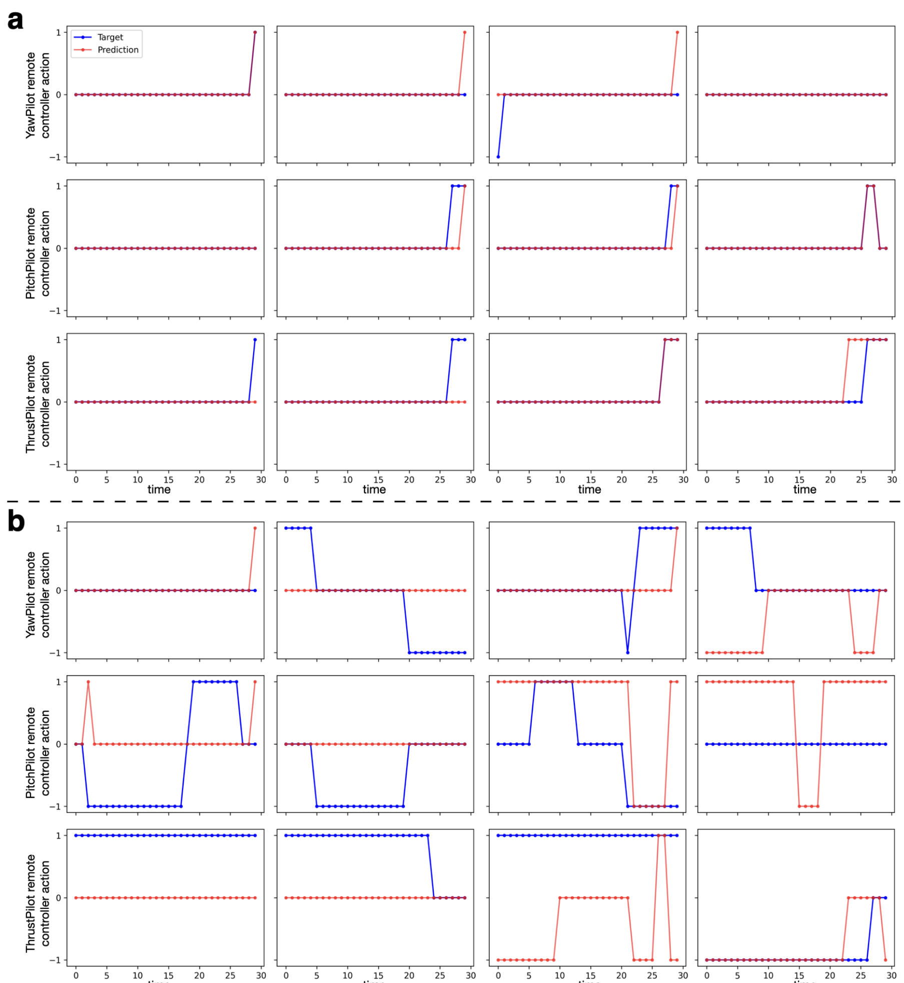


SI Fig. 1: Predictive modal with speech event as input. **a**, Multi-head attention model structure. Similar to the model described in the main text, this model includes both encoder and decoder layers, with speech events incorporated as an additional input in the encoder. **b** Team predictability based on inputs from four data modalities across three experimental sessions. Each dot represents one team, and the bars show the average predictability across all teams ($n = 10$). ns indicates no significant difference. **c**, Correlation between team performance and predictability derived from four data modalities. The red arrow highlights positive correlations, and asterisks denote statistically significant differences, defined as $* * *P < 0.001$.

as an additional input. As shown in Fig. 1 a, the encoder layer now includes an additional branch to process speech event data. The team predictability computed using this model is presented in Fig. 1 b. Experimental sessions had a slight effect on team predictability, though not statistically significant ($F(2, 10) = 0.32, P = 0.733$s). We then evaluated the correlation between predictability from this model and team performance. Our results confirm that the significant positive correlation between predictability and team performance remains consistent (Fig. 1 c). These findings suggest that incorporating speech events into the model enhances its ability to capture key dynamics that contribute to team coordination and overall performance.

Model Prediction Results

The multi-head attention model aims to forecast the future remote controller actions of an individual based on two other team members' past behavioral and physiological data. We first trained a predictive model that takes two team members' remote controller actions, pupil sizes, EEG signals, and spacecraft location and forecasts the other (the left out) teammate's actions (Fig. 3 a in main text). SI Fig. 2 a are forecasts of the remote controller actions that are highly accurate. In contrast, SI Fig. 2 b shows cases in which forecasts were less accurate.



SI Fig. 2: Sample target and predicted remote controller actions. **a**, Prediction and target remote controller actions of each role. These examples were randomly selected from test data. Each example achieves high target-prediction correlations. **b**, Prediction and the target remote controllers with low prediction accuracies.

Generalized Linear Mixed Model (GLMM)

We used the GLMM to analyze the correlation between different variables. The variables are:

- Global performance – the performance evaluated by the total number of ring objects passed by each team.
- Local performance – the performance evaluated based on each ring passing event by considering deviation from the center of the ring and the speed of passing the ring.
- Predictability – the averaged predictability across three co-pilots.
- Helpfulness – the total helpfulness score calculated by adding all the helpfulness ratings that co-pilots have given to each other.
- Familiarity – the total familiarity score calculated by adding all the familiarity ratings that co-pilots have given to each other.
- Pupil size synchrony – the averaged Pearson correlation among all co-pilots' pupil sizes.
- EEG synchrony – the inter-brain synchrony among all co-pilots' EEG signals.
- Remote controller action synchrony – the averaged Pearson correlation among all co-pilots' remote controller actions.
- Speech event synchrony – the averaged Pearson correlation among all co-pilots' speech events.

In our study, the experimental session is an adaptive effect of some variables because teams perform better and get more familiar with each other throughout the sessions (Fig.1 and Fig.2 in main text). Therefore, we used session (i.e., 1, 2, 3) as an additional independentThe number of rings measures the performancession-level effect.

$$\text{Dependent} \sim 1 + \text{Session} + \text{Independent} + (1|\text{Team}) + \epsilon \tag{1}$$

References

[1] Falcon, W. A. Pytorch lightning. *GitHub* **3** (2019).

[2] Paszke, A. *et al.* Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems* **32** (2019).