Emergence of concept specific information in precleaned fixational eye movements in a self-supervised representation learning regime

Svea Kürten & Saskia Fohs

Eye movements are an issue in studies of the neural system (Mostert et al., 2018), e.g. by inflating decoding accuracy (Quax et al., 2019). This effect has previously been shown in simplistic low-level stimuli, by being able to decode the content of a percept from eye movements using multi-variate pattern analysis. In this project we attempt to decode stimulus and concept (label) information from non-free-viewing eye movements. The goal of this work is to later observe if this information is also present in neuronal data, that ought to be cleaned from eye induced artifacts. This paper lays the first step by showing that concept information is preserved in restricted and pre-cleaned eye movements even in a fully data-driven learning approach.

Dataset

The MEG-1 dataset of the THINGS initiative (Hebart et al., 2023) was used. Here we used the already preprocessed 1200Hz eye tracking data. For this dataset a total of four participants was scanned over twelve sessions. Here we only used the first two participants. A total of ~27k trials were collected per participant. The participants were shown different images out of 1854 categories with a trial consisting of data from -100 to 1300ms relative to stimulus onset, with the task “Is this image an object?”. (Hebart et al., 2023)

Preprocessing

Before using the data for model training, a preprocessing pipeline was applied. We first baselined each trial with the pre-stimulus period and got rid of catch images (*preprocess_eye/preproc_eyes_bin_index_baseline.py*). Then we binned our data, for potential later fine-grained analysis, but in this paper, we only used the complete trials. We excluded all trials/bins where 50% of the signal was missing on any channel (e.g. due to blinks) and set missing values to 0. Furthermore, the data was recoded to a dataset-friendly structure with also corresponding label and image identity ground truth (*preprocess_eye/preproc_things_format.py*).

Supervised Classifiers

First, to see if there was label information decodable in the data, we used a supervised regime to train two different classifiers on the eye movement data: one transformer encoder and one 1D PreResNet (He et al., 2016b). We deployed both structures to be able to later probe timepoints (transformer attention maps) and observe the kind of signal features that were extracted (kernel activations in the CNN). Furthermore, the comparison of both models allowed indirect inference, of what kind of structure is learned. The models trained on 1/3 of the labels, as we first planned to use the classifiers as pretraining for a contrastive learning approach. Generalization of the models was assessed with one left-out sample per label. Both models were trained

with cosine annealing. For each subject a new model instance was trained for 1000 epochs. (*decode_eye/train_model.py*)

We adopted a PreResNet structure (He et al., 2016b) instead of a traditional ResNet (He et al., 2016a) to better apply dropout (Kim et al., 2023). The model consists of one basic convolutional layer and three basic blocks. After a global average pooling (GAP) the averages are fed into a classification head. This resulted in (depending on label cut) ~597k parameters. To avoid overfitting additive amplitude jitter and smooth time masking (Rommel et al., 2022) were applied to the data in training. Furthermore, 10% dropout was applied. (*decode_eye/CNN_model.py*)

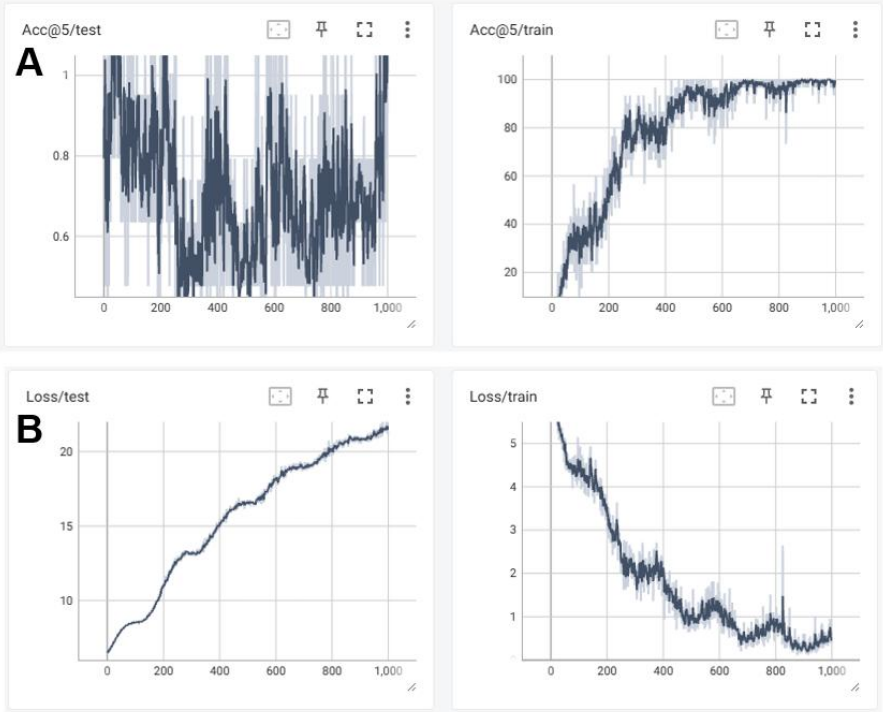

Figure 1. Overfitting of the CNN for one subject. **A** shows the top 5 accuracy in training and validation set. While the training accuracy increases over the epochs, the validation accuracy heavily oscillates around chance level. **B** shows the loss over the epochs. Here it is evident, that while the training loss decreases, the validation loss increases.

The transformer encoder was loosely oriented on a Vision Transformer architecture (Dosovitskiy et al., 2020) and consisted of a 1x1 convolution layer, to project all timepoints into a higher dimensional embedding space (from x, y and pupil to 32 embedding dimensions). Afterwards a CLS token (Dosovitskiy et al., 2020) and a sin-cos positional embedding was added. The model consists of 4 attention heads in 1 layer. After the embeddings are contextualized, GAP is applied to all tokens except the CLS token. Then the CLS token and the GAP are concatenated before being fed into a classification head. By this the model can “decide” if it wants to weigh on the CLS token that was contextualized in each step, or the holistic sequence embedding, resembled by the GAP. The same augmentations were applied as in the PreResNet. The model had a total of ~109k parameters. (*decode_eye/transformer_model.py*)

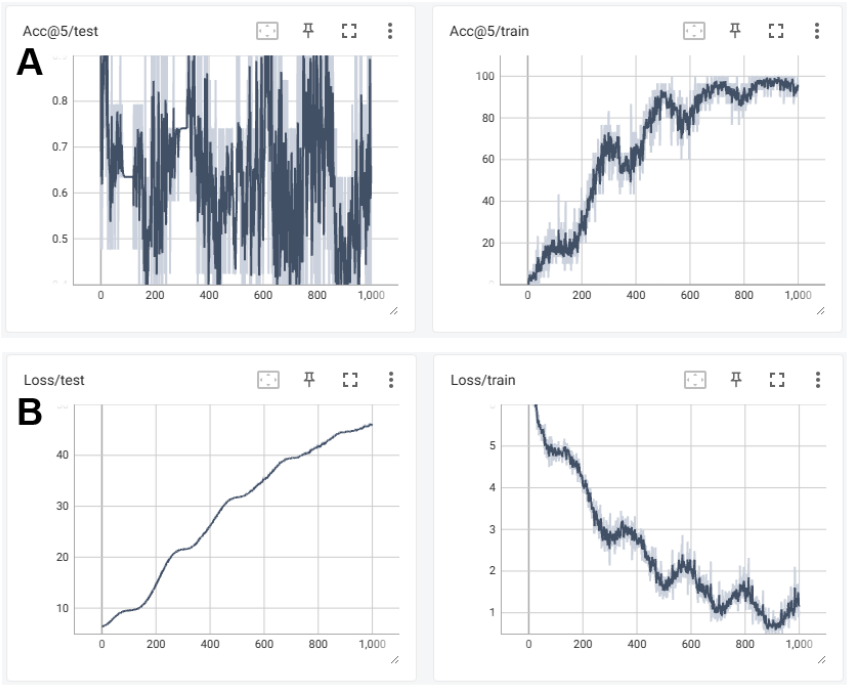

Figure 2. Overfitting of the transformern for one subject. **A** shows the same pattern as the CNN. **B** also shows again that validation loss rises, the training loss decreases.

Both figure 1 and figure 2 clearly indicate that the classifiers were overfitting to the training data. This overfitting effect could not be mitigated by a sharper dropout (up to 30%) or stricter augmentation. This is why we did not further consider supervised methods to solve this issue. Also, because at this point, there might not have been any label information in the data.

Contrastive Learning

By deploying a self-supervised learning regime, we avoided overfitting to labels, because each model is just matched with its image representations, irrespective of label information. For this approach, we aimed to train an eye encoder to match the representation of the trial's shown image of a pretrained image encoder in latent projection space. To this end, we froze a ViT-B-32 model (Dosovitskiy et al., 2020) that was pretrained on ImageNet and used its last encoder output (= model without the head) as image embeddings. This model is also used as an image encoder in other contrastive learning models like CLIP (Cherti et al., 2023). We then trained our previously explained structures without their classification heads as eye-movement encoders. Furthermore, we trained two projection layers to the common space, for each modality respectively. The training objective was to match the mean self-similarity of the two modality-embeddings (Cherti et al., 2023). Also, a cosine annealing was applied, but we did not make use of data augmentation or dropout. As shown in the attention maps interpretations, this was an error and will be adapted in further steps of

this project. The models were trained on 2/3 of the labels, to preserve 1/3 of data for testing. Validation was again one sample per label of the training data. A batch size of 50 was used with a total of ~11k training samples (*contrastive_eye_image/transformer_cl_model.py, cnn_cl_model.py, train_cl_model.py*)

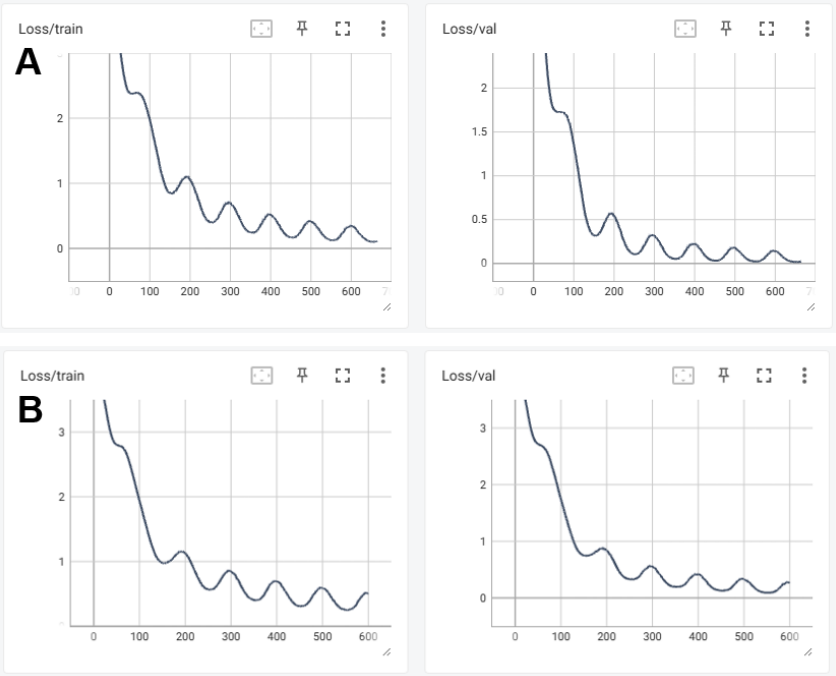

Figure 3. Loss of the contrastive learning models for one subject. **A** shows the loss of the training and validation set for the CNN encoder. **B** shows the loss for the transformer encoder.

Figure 3 shows that the models did not overfit on the data. What might appear odd is that the validation loss is smaller than the training loss. This occurs due to the last batch per validation epoch being smaller than in the rest of the epochs, which decreases the loss systematically. We did not want to shuffle or validation batches, because especially in contrastive learning this leads to a different matching of datasets for each iteration. We also did not want to use drop last, to ensure all test sets are tested. This will be optimized in the future. Also, the cosine annealing might be a bit harsh, as already seen in the first two figures (the waves).

Concepts emerge in self-supervised models

We tested our models with zero shot evaluations. Those were done in a pairwise matter (chance = 50%). So, with one trial of eye movement data, the model must select which of two images was seen in that trial. Furthermore, we established two conditions for this 2AFC task: one between images of the same label and one between images of different labels. If the model is worse in selecting the true image within a label, then there is still label information that the classifiers were unable to learn. We also tested the left-out labels, to observe how well the training data generalizes. Accuracy describes in this case the proportion of higher probability for the correct image.

(*contrastive_eye_image/eval/test_script.py, test_script_left_out.py, zero_shot_pairwise.py, analysis.py*)

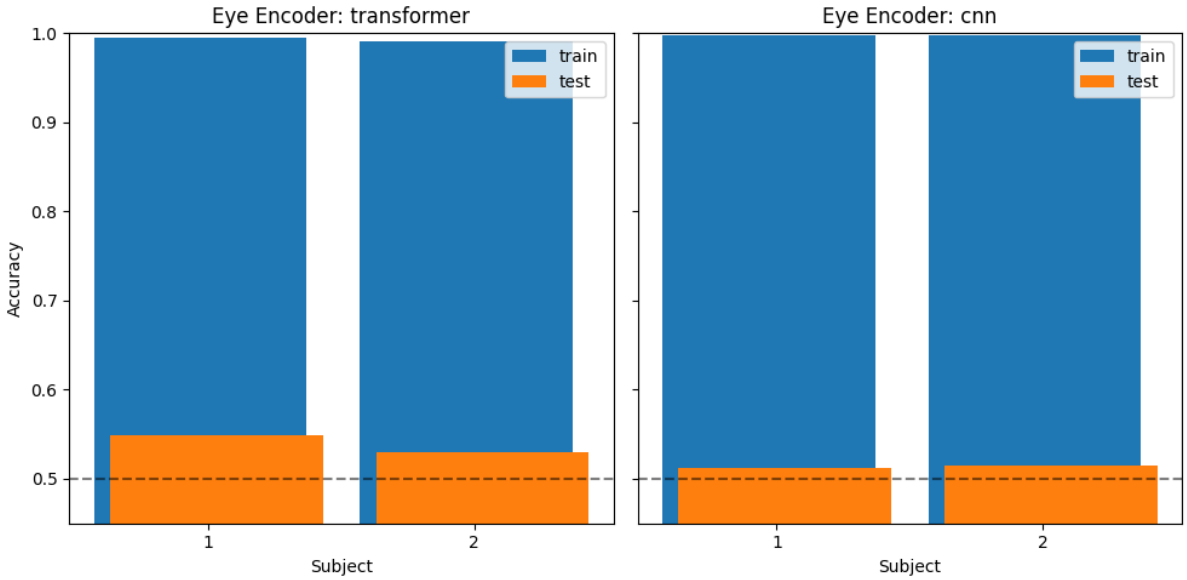

Figure 4. Accuracy of the two encoder models. Per subject and split into test and training sets.

The transformer encoder achieved in both subjects a higher test set accuracy than the CNN (figure 4). Furthermore, the transformer performed better for subject 1. With just two subjects statistical inference is not viable, which is why we skipped this part of our work (at least for the sake of this paper). Still across two participants both models achieved above chance performance in the test set and near perfect performance for the training data.

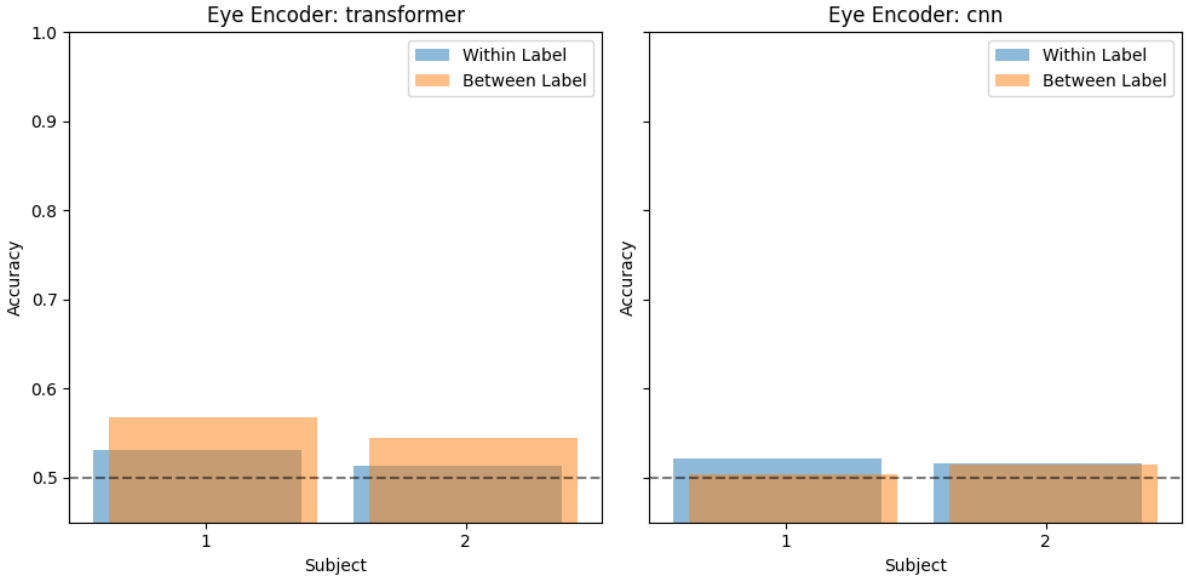

Figure 5. Test set accuracy within and between labels for each model and subject.

The CNN does not show large differences between the conditions (figure 5). For subject 1 it might even appear that the CNN model was worse to distinguish images from different labels. The transformer encoder on the other hand, performs better in both participants between labels than within a label. So, within a label, there might be

common information that decreases decoding performance within label differentiation. All following analyses were only applied in the left-out data.

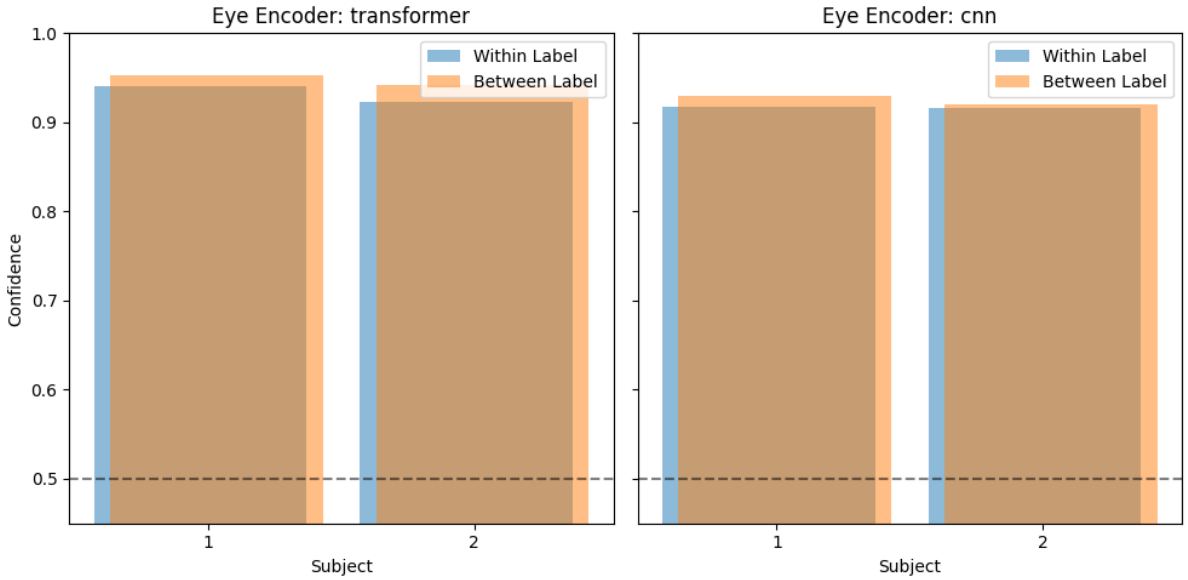

Figure 6. Confidence as the difference between the probability for the correct and the incorrect image, only given for correct answers, for each subject, model and task condition.

To check if there is possibly still an effect of certainty, or confidence in the CNN, we took the mean difference of the probabilities for the correct and incorrect images, for both models, but only in correct trials (figure 6). This aims to answer the question: if the model was correct, was it less certain for the within label tasks? The transformer showed a clear effect across the two subjects, that in correct trials, it was less certain within label. But this effect, albeit smaller, also seems to be present in the CNN. This illustrates that there might also be label information in the CNN. Furthermore, this analysis shows that the models were confident when making correct choices, with the transformer having an edge over the CNN.

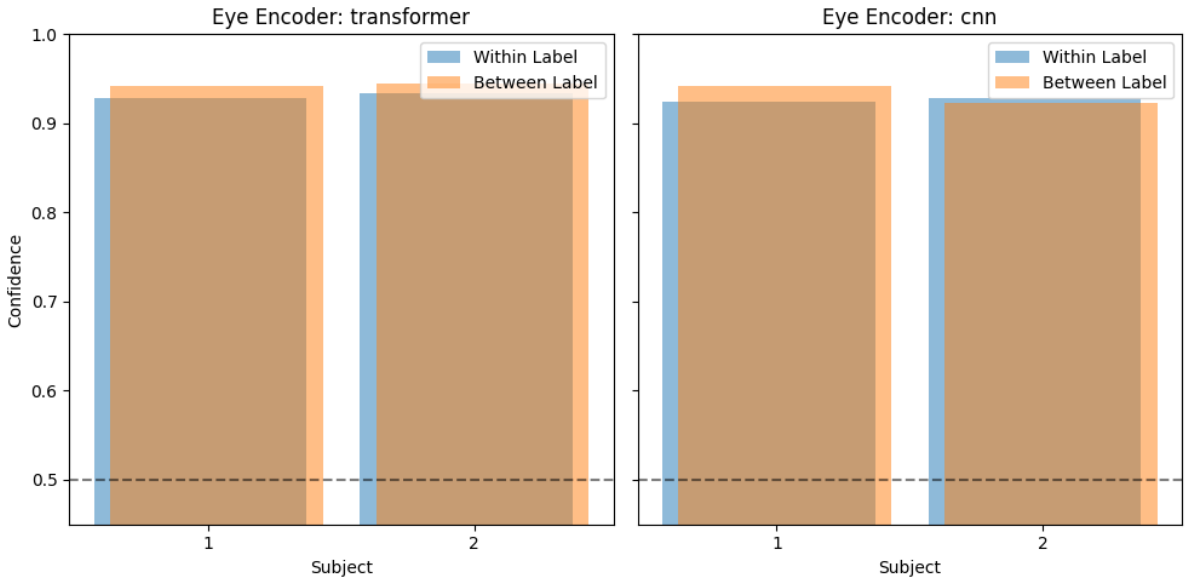

Figure 7. Confidence as the difference between the probability for the correct and the incorrect image, only given for incorrect answers, for each subject, model and task condition

Figure 7 shows that also when being incorrect, the model is quite confident, in its decision. This might indicate that the model did not consider its wrong decisions as difficult cases but was just straight up wrong. This might indicate too hard decision boundaries. For the future of our project we will not use softmax on our cosine similarities in evaluation anymore, to avoid artificially increasing confidence.

Visualizing the eye movements on top of the images is in our case not feasible, since those were most likely micro-saccades or already cleaned by Hebart et al. (2023).

Discussion

We were able to find evidence that there is complex stimulus concept information in fixational eye movements without the participant's ability to free view. We were not able to decode this concept information in a supervised manner. By employing contrastive learning across image and eye movement modalities, we were able to decode images from eye movements. Furthermore, even in complete absence of the labels during the training process, we found label information in eye movements.

Additionally, this information could mostly be retrieved by using transformers instead of CNNs. This might be because the transformer can capture more global dependencies of the signal. We consider exploring this further by incorporating lateral recurrent connections in our CNN to mimic this process on a smaller scale, expecting more of the within label effect on accuracy in CNNs.

Even though the accuracy scores of the models indicate that only the transformer was able to emergently learn a label structure, the confidence score difference when being correct also shows a task effect in the CNN. When it was correct it was less confident within labels. This might indicate a more subtle representation of label information in the model, that does not come to light during the 2AFC.

Is this an issue for neural decoding? The observation that mostly the transformer led to decoding of label specific information, makes this question a matter of context. If we are decoding timepoint wise or with smaller bins from neural data, this might not lead to issues. But decoding from whole trials could lead to issues with some part of the information stemming from eye movements. To this end, one should relate the decoding performance of an eye movement and neural decoder over the same and multiple train-test splits, to observe systematics and thereby non-dissociability between the signals (Jones et al., 2024).

Also, we might make use of our binned data, to see if we are able to observe the interdependency that allows the transformer to decode this information in a shorter period. Furthermore, attention maps and RSAs will be applied in assignment 4. The RSAs showed that the model representations themselves are not similar between encoded modalities, but that this similarity only arises due to the non-linear projections (see Sveas assignment). The attention maps of the transformer models showed that only some timepoints were amplified, which led to highly similar attention maps between different trials. This exemplifies the need for dropout to learn more diverse signal features for the next steps (see Saskias assignment).

References

Cherti, M., Beaumont, R., Wightman, R., Wortsman, M., Ilharco, G., Gordon, C., ... & Jitsev, J. (2023). Reproducible scaling laws for contrastive language-image learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 2818-2829).

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., ... & Houlsby, N. (2020). An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Identity mappings in deep residual networks. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part IV 14* (pp. 630-645). Springer International Publishing.

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 770-778).

Hebart, M. N., Contier, O., Teichmann, L., Rockter, A. H., Zheng, C. Y., Kidder, A., ... & Baker, C. I. (2023). THINGS-data, a multimodal collection of large-scale datasets for investigating object representations in human brain and behavior. *Elife*, *12*, e82580.

Jones, H. M., Thyer, W. S., Suplica, D., & Awh, E. (2024). Cortically disparate visual features evoke content-independent load signals during storage in working memory. *Journal of Neuroscience*, *44*(44).

Kim, B. J., Choi, H., Jang, H., Lee, D., & Kim, S. W. (2023, July). How to use dropout correctly on residual networks with batch normalization. In *Uncertainty in Artificial Intelligence* (pp. 1058-1067). PMLR.

Mostert, P., Albers, A. M., Brinkman, L., Todorova, L., Kok, P., & De Lange, F. P. (2018). Eye movement-related confounds in neural decoding of visual working memory representations. *Eneuro*, *5*(4).

Quax, S. C., Dijkstra, N., van Staveren, M. J., Bosch, S. E., & van Gerven, M. A. (2019). Eye movements explain decodability during perception and cued attention in MEG. *Neuroimage*, *195*, 444-453.

Rommel, C., Paillard, J., Moreau, T., & Gramfort, A. (2022). Data augmentation for learning predictive models on EEG: a systematic comparison. *Journal of Neural Engineering*, *19*(6), 066020.