CoordConformer
Heterogenous EEG datasets decoding using Transformers

Sharat Patil [1] **Robin Tibor Schirrmeister** [2] **Frank Hutter** [1 3] **Tonio Ball** [2]

Abstract

Transfer Learning and meta-learning have been effective in improving performance across multiple domains. It has also been applied successfully to EEG decoding where there is a lack of data. However, there are unique challenges for transfer learning with EEG data across datasets due to differences in experimental setup, like different numbers of electrodes, different positions of the electrodes, and different task definitions. To tackle the issue of cross-dataset training across heterogeneous electrode configuration EEG datasets we introduce a novel method, CoordinateAttention, that uses 3-D coordinates of the electrode sensors to learn the spatial relationship between the electrode's positions to dynamically generate spatial convolution kernels for feature extraction. We show that our model has good performance in EEG decoding across settings and is robust to data corruption. CoordinateAttention is a general-purpose method for feature extraction and data fusion using geometric positional information.

1. Introduction

Electroencephalography (EEG) allows us to record electrical activity in the brain with high temporal resolution for studying brain functions. It is an important method in various fields, including neuroscience, cognitive science, and clinical diagnostics. EEG signals are recorded by electrodes placed on the scalp in a particular configuration called montages. However, EEG data is highly variable and influenced by numerous factors such as subject-specific characteristics, recording conditions, and the experimental paradigms employed. EEG signals also have signal non-stationarities and poor signal-to-noise ratio. This presents significant challenges for effective decoding and analysis. Recording EEG data is an intensive task and therefore the field suffers from data scarcity. In recent years transfer learning methods have been applied to improve EEG decoding (Gu et al., 2023; Kim et al., 2024; Han et al., 2023). There are generally two types of transfer learning in EEG decoding cross subject transfer, where the data for each subject is collected using the same experimental paradigm, and cross-dataset transfer where the datasets have different experimental paradigms like different numbers of electrodes, different positions of the electrodes, and different task definitions.

Current approaches to cross-dataset transfer are learning dataset-specific feature extractors and then using a common feature processing trunk with feature alignments between the heterogeneous datasets (Han et al., 2023). This means that such approaches cannot be generalized easily to new datasets with different electrode setups and require retraining. Some approaches avoid this by training on the common subset of electrodes between the datasets (Guetschel & Tangermann, 2023) however it means a lot of the data is not used for making the predictions and the model fails for datasets that do not have any common electrodes. Other methods use padding to cover all the electrode channels across the datasets but can not generalize to new electrode positions. To overcome these limitations we create a feature extraction module that uses the electrode positions on the scalp to dynamically perform spatial feature extraction during the forward pass, thus not being limited to any specific number of electrodes or configuration.

We base our model, CoordConformer's architecture on the EEGConformer (Song et al., 2023), a transformer encoder model (Vaswani et al., 2017) which has state-of-the-art performance on Motor Imagery decoding tasks. It uses spatial convolutions for feature extraction across the electrode channels. The spatial convolution layer requires the number of electrode channels to be fixed for its kernel size and removing the spatial convolution layer results in an irrecoverable performance drop, so we replace the spatial convolution with our novel dynamic module, CoordinateAttention which can process EEG data with any number and configuration of the electrodes.

[1]Machine Learning Lab, University of Freiburg, Germany [2]Neuromedical AI Lab, University of Freiburg, Germany [3]ELLIS Institute Tübingen, Germany. Correspondence to: Sharat Patil <patilsh@cs.uni-freiburg.de>.

Accepted as an extended abstract for the Geometry-grounded Representation Learning and Generative Modeling Workshop at the 41st International Conference on Machine Learning, ICML 2024, Vienna, Austria.

The CoordinateAttention module takes the 3-D coordinates of the EEG channels as its input and learns a geometric relationship between the electrode positions to generate a convolution kernel. The generated convolution kernel is then used to perform the spatial convolutions.

2. Methods

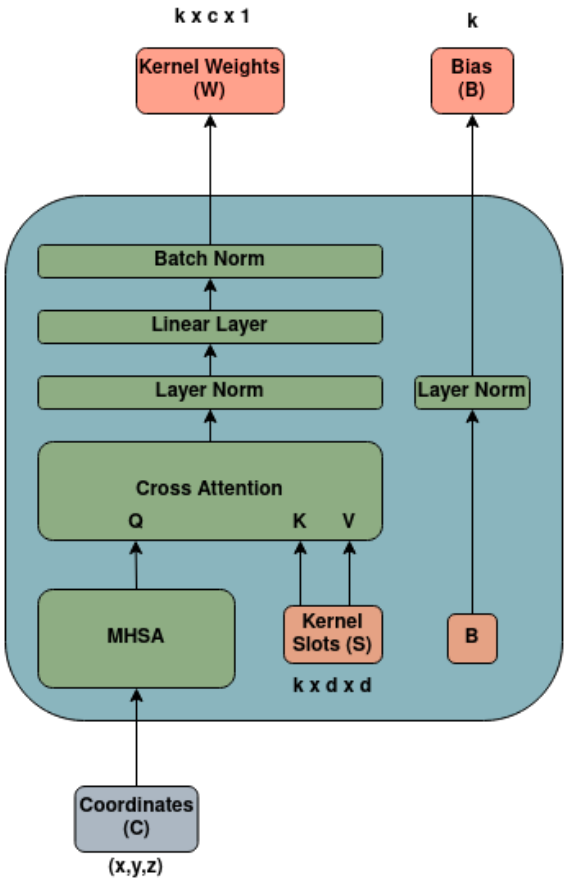

Figure 1. CoordinateAttention Module

Our model, CoordConformer consists of three main modules, i) a feature extraction module, ii) an encoder stack, and iii) a classifier head. The feature extraction module processes the raw EEG input data (trial), $X \in \mathbb{R}^{c \times t}$ where c is the number of electrodes (channels) and t is the number of samples and converts it into patches (tokens) similar to a vision transformer. The encoder stack processes the patches using self-attention (Vaswani et al., 2017) and the classifier head uses the encoder outputs to classify the trial. The feature extraction module applies convolutions along the spatial axis, that is across the electrode channels and the temporal axis for feature extraction, followed by average pooling to obtain the patch tokens. The original EEGConformer applies temporal convolutions followed by spatial convolutions, but we switch the order to avoid having to generate kernels of size $k_s \times k_t \times c \times 1$ instead of 1-D kernels of size $k_s \times c \times 1$. We use per-dataset classifier heads to train across datasets with different classification tasks. We also use adaptive batch instance normalization(Li et al., 2016; Nam & Kim, 2018) for each of the datasets to implicitly handle the feature alignment (Xu et al., 2021; Bakas et al., 2023).

2.1. CoordinateAttention

To tackle the issue of applying convolution on data with varying numbers of channels we dynamically generate the convolution kernel during the forward pass using the electrode positions and attention mechanisms. The CoordinateAttention module uses self-attention and cross-attention to generate the kernels. The module takes as input the idealized 3D coordinates of the electrodes, $C \in \mathbb{R}^{c \times 3}$ where c is the number of electrode channels in the input data, and generates the spatial kernel weights, $W \in \mathbb{R}^{k \times c \times 1}$, where k is the number of kernels to generate, and the bias weight, $B \in \mathbb{R}^{k}$, which are then used to perform spatial feature extraction on the data.

$$
\begin{aligned}
Q &= LinearLayer(C) \\
K &= LinearLayer(S) \\
V &= LinearLayer(S) \\
\\
Q^{'} &= MultiHeadSelfAttention(Q), \\
O &= CrossAttention(Q^{'}, K, V), \\
W &= LinearLayer(O).
\end{aligned}
\tag{1}
$$

The electrode coordinates are replicated k times so that each kernel gets generated independently of each other. We first apply multi-headed self-attention on the input coordinates to learn the relationships between the electrodes. We then apply cross attention between the processed coordinate embeddings as the queries $Q \in \mathbb{R}^{k \times c \times d}$ and learnable parameters called kernel slots $S \in \mathbb{R}^{k \times d \times d}$ as the keys $K \in \mathbb{R}^{k \times d \times d}$ and values $V \in \mathbb{R}^{k \times d \times d}$, where d is the model dimension. The cross-attention output $O \in \mathbb{R}^{k \times c \times d}$ is then passed to a LinearLayer to reduce its dimension. The module returns W and B which are used to perform 1-D convolution on the input data.

2.2. Cropped Decoding

Cropped decoding is an approach for EEG decoding presented in (Tangermann et al., 2012) that averages the predictions of small sub-windows over the trials rather than predicting over the entire trial. (Schirrmeister et al., 2017) introduced a computationally efficient implementation for

Table 1. Performnace on Motor Imagery decoding benchmarks.

DATASET	BNCI2014	HGD
COORDCONFORMER-S	76.1%	93.3%
COORDCONFORMER-F	77.6%	**94.7%**
EEGCONFORMER	**78.6%**	-
DEEP4	72.53	92.9%

Table 2. Performance on the BEETL Competition.

MODEL	ACCURACY
COORDCONFORMER	73.5%
TEAM COGITAT	**76.3%**
TEAM WDUONG	71.3%
TEAM MS01	59.9%

passing all the crops through the network at the same time using dilated convolutions. We extend it for the transformer encoder by masking the attention between the patches so that each patch only attends to patches in its crop neighborhood. This allows us to make multiple predictions for each trial that get averaged to get the final prediction.

2.3. Data Preprocessing and Augmentations

We preprocess the EEG data for all the datasets by resampling the signal to 250hz and normalizing the data together across all the datasets. For all trials, we use the first 4 seconds to make the prediction. Along with the Substitution and Reconstruction (Lotte, 2015) data augmentation used by (Song et al., 2023) for the EEGConformer we also apply random electrode channel dropping and shuffling.

2.4. Training Details

When training on multiple datasets we first train the model on all the datasets jointly and then further finetune it for each test subject independently. For training the models we use a cosine annealing learning rate scheduler with an AdamW optimizer (Loshchilov & Hutter, 2019). We use a per-dataset Cross Entropy loss. Label smoothing (Huang et al., 2021) and center loss (Wen et al., 2016; Huang et al., 2021) are used for regularization. Detailed hyperparameters are listed in Appendix A.

3. Results

We evaluate our model across multiple settings. First, we evaluate its performance on single datasets on Motor Imagery decoding benchmarks (Tangermann et al., 2012; Schirrmeister et al., 2017). Then we evaluate its performance in cross-dataset learning on the 2021 BEETL competition (Wei et al., 2022) Task-2. We then study our model's robustness and anytime performance by testing it on data with missing electrode channels and unseen electrode channels. We report all our performance metrics as the average of three seeds.

3.1. Motor Imagery Decoding

We first evaluate our model's performance on single datasets. We evaluate it on two popular motor imagery benchmark datasets: Dataset IIa from BCI Competition 4 (BNCI2014) (Tangermann et al., 2012) and the High Gamma Dataset (HGD) (Schirrmeister et al., 2017). The BNCI2014 dataset contains EEG data from 9 subjects recorded using 22 electrodes with four classes of imagined movement (left hand, right hand, both feet and tongue). The High Gamma Dataset contains EEG data from 14 subjects recorded using 128 electrodes with four classes of imagined movement (left hand, right hand, both feet and rest). For the BNCI2014 dataset, we train on all 22 electrodes and for the High gamma dataset we train on the 44 electrodes that cover the sensorimotor cortical area. We train two sets of models for each of the datasets: one trained on a single subject (CoordConformer-S) and the other first trained on all subjects together and then finetuned for each subject (CoordConformer-F). We compare our performance using the average accuracy across all subjects against the EEGConformer and the Deep4 (Schirrmeister et al., 2017), a CNN based EEG classification model. [1]

From Table 1 we see that we are able to get comparable performance to other SOTA methods on both of the datasets indicating it is a good model for EEG decoding even with only one training dataset. We also see that joint training on all subjects and then finetuning for each subject increases the accuracy of our model on both datasets indicating that the model has good cross-subject training performance.

3.2. Cross-Dataset Training

We evaluate our model for cross-dataset learning on the 2021 BEETL Competition Task-2 (Wei et al., 2022). The task is a 3 class motor imagery classification challenge. Three full data sets were provided as training data BNCI2014 (Tangermann et al., 2012), Cho2017 (Cho et al., 2017) PhysionetMI (Schalk et al., 2004; Goldberger et al., 2000) and the evaluation was done on two hidden datasets Weibo2014 (Yi et al., 2014) and CyblathonIC (Wei et al., 2021) with a total of five test subjects, from which few samples per test subjects were provided for calibration. All the datasets including the evaluation datasets have different setups, including differences in electrode channels, task definitions, number of subjects, and, number of trials per class. The dataset details are given

[1] Results for both models on the BNCI2014 are taken from their respective papers.

Table 3. Performance with missing electrode channels.

MODEL	LOW		HIGH	
	$P_>$	ACC.	$P_>$	ACC.
COORDCONFORMER	**1.0**	**85.9%**	**0.85**	**69.5%**
DEEP4	0.0	75.1%	0.15	60.5%

Table 4. Performance on Unseen channels.

	16	32	64
TRAIN CHANNELS	90.2%	94.1%	95.3%
UNSEEN CHANNELS	71.8%	78.2%	82.8%
ALL CHANNELS	75.9%	82.7%	90.6%
50% TRAIN CHANNELS	48.2%	64.8%	73.0%
MIXED CHANNELS	63.1%	77.2%	78.1%

in Appendix B.

We first jointly train the model across all of the training datasets including the calibration data from the evaluation datasets with a per-dataset classifier head setup. During the joint training, we balance the sampled batches for class balance within each dataset and also balance the number of batches sampled for each dataset at the epoch level. The trained model is then finetuned for each of the five test subjects separately.

We compare our performance against the top three performing teams of the competition and can get a weighted accuracy of 73.5% beating the second and third place comfortably. The CoordConformer thus is able to learn across datasets and leverage information from larger datasets to perform well on much smaller datasets even when the task definitions are different.

3.3. Robustness to Missing Channels

We also evaluate our model's ability to handle missing electrode channels. We compare it against Deep4 on the High Gamma Dataset. Both models were trained on the 44 cortex electrode channels. We perform two sets of evaluations, in the first we randomly drop between 5-20 electrode channels (Low) from the test data, and in the second we randomly drop between 10-30 electrode channels (High) from the test data. The models are directly evaluated on the test data with missing channels without any re-training. We perform twenty such random evaluations for each of the two sets and repeat it across three seeds for reporting. An average of 27% and 47% of the test channels are dropped in the low and High settings respectively. We report the results as the probability $P_>$ of the model outperforming its competitor in an evaluation and its average accuracy across the evaluations.

From Table 3 we see that our model outperforms Deep4 in 100% of the evaluations for the Low setting and 85% of the evaluations in the High setting. Furthermore, our model has a smaller drop in accuracy on average compared to the Deep4 model when evaluated with electrode channels missing from the test data, getting almost 10% more accuracy in both settings.

3.4. Predicting on Unseen Channels

We now evaluate our model's ability to use information from new electrode channels that it wasn't trained on. We train our model on the High gamma dataset with three different electrode configurations consisting of 16, 32, and 64 electrodes each of which is a subset of the full 128 electrode channels present in the full data. We evaluate it on the test data with five electrode channel configurations 1) With all 128 electrodes (All Channels), 2) only the training electrode channels of the model (Train Channels), 3) the electrode channels not in the training data (Unseen Channels), 4) randomly selected 50% of the training electrode channels(0.5 Train Channel), 5) mixture of the randomly selected 50% of the training channels and randomly selcted 50% of the unseen electrode channels (Mixed Channels). We again report the average accuracy across all subjects.

From Table 4 we see that the model performs best when it is tested using only the training electrode channels indicating that the model might be overfitting to the training electrode configuration. Adding unseen electrode channels reduces the accuracy by adding noise to the model. The model when tested on only the Unseen Channels gets non-trivial accuracy especially in the 64-electrode model, showing that it is capable of good anytime performance on data with a completely new configuration that it has never seen before. we also see that when we test with a mixture of partial training channels and unseen channels the model can use the information from unseen channels to recover the performance drop due to the missing training channels. The CoordConformer can leverage information from the unseen electrode channels to improve its accuracy, however, it is prone to overfitting and might need more regularization to prevent it.

4. Conclusion

We introduce CoordConformer a model capable of training on EEG data with different electrode configurations. The CoordinateAttention module uses attention mechanisms to learn the geometric relationships between the electrode positions to dynamically apply spatial convolutions. Our approach is general and can be extended to other problems and domains in a straightforward manner. CoordinateAttention's

modular nature allows us to use it to replace any convolutional layer where the data has a varying number of input channels. We have shown that our model has good performance in EEG decoding across various settings. The model needs to be further evaluated on more cross-dataset settings and the differences in the generated kernels across electrode configurations need to be studied for interpretability.

5. Acknowledgments

We acknowledge funding by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under SFB 1597 (SmallData), grant number 499552394.

References

Bakas, S., Ludwig, S., Adamos, D. A., Laskaris, N., Panagakis, Y., and Zafeiriou, S. Latent alignment with deep set eeg decoders, 2023.

Cho, H., Ahn, M., Ahn, S., Kwon, M., and Jun, S. C. EEG datasets for motor imagery brain–computer interface. *GigaScience*, 6(7):gix034, 05 2017. ISSN 2047-217X. doi: 10.1093/gigascience/gix034. URL https://doi.org/10.1093/gigascience/gix034.

Goldberger, A. L., Amaral, L. A., Glass, L., Hausdorff, J. M., Ivanov, P. C., Mark, R. G., Mietus, J. E., Moody, G. B., Peng, C.-K., and Stanley, H. E. Physiobank, physiotoolkit, and physionet: components of a new research resource for complex physiologic signals. *circulation*, 101(23): e215–e220, 2000.

Gu, X., Han, J., Yang, G.-Z., and Lo, B. Generalizable movement intention recognition with multiple heterogeneous eeg datasets. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 9858–9864, 2023. doi: 10.1109/ICRA48891.2023.10160462.

Guetschel, P. and Tangermann, M. Transfer learning between motor imagery datasets using deep learning–validation of framework and comparison of datasets. *arXiv preprint arXiv:2311.16109*, 2023.

Han, J., Wei, X., and Faisal, A. A. Eeg decoding for datasets with heterogenous electrode configurations using transfer learning graph neural networks. *Journal of Neural Engineering*, 20(6):066027, 2023.

Huang, X., Zhou, N., and Choi, K.-S. A generalizable and discriminative learning method for deep eeg-based motor imagery classification. *Frontiers in Neuroscience*, 15: 760979, 2021.

Kim, J.-M., Bak, S., Nam, H., Choi, W., and Kam, T.-E. Meta-learning-based cross-dataset motor imagery brain-computer interface. In *2024 12th International Winter Conference on Brain-Computer Interface (BCI)*, pp. 1–4, 2024. doi: 10.1109/BCI60775.2024.10480445.

Li, Y., Wang, N., Shi, J., Liu, J., and Hou, X. Revisiting batch normalization for practical domain adaptation, 2016.

Loshchilov, I. and Hutter, F. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019.

Lotte, F. Signal processing approaches to minimize or suppress calibration time in oscillatory activity-based brain–computer interfaces. *Proceedings of the IEEE*, 103(6): 871–890, 2015.

Nam, H. and Kim, H.-E. Batch-instance normalization for adaptively style-invariant neural networks. *Advances in Neural Information Processing Systems*, 31, 2018.

Schalk, G., McFarland, D., Hinterberger, T., Birbaumer, N., and Wolpaw, J. Bci2000: a general-purpose brain-computer interface (bci) system. *IEEE Transactions on Biomedical Engineering*, 51(6):1034–1043, 2004. doi: 10.1109/TBME.2004.827072.

Schirrmeister, R. T., Springenberg, J. T., Fiederer, L. D. J., Glasstetter, M., Eggensperger, K., Tangermann, M., Hutter, F., Burgard, W., and Ball, T. Deep learning with convolutional neural networks for eeg decoding and visualization. *Human Brain Mapping*, 38(11): 5391–5420, 2017. doi: https://doi.org/10.1002/hbm.23730. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/hbm.23730.

Song, Y., Zheng, Q., Liu, B., and Gao, X. Eeg conformer: Convolutional transformer for eeg decoding and visualization. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 31:710–719, 2023. doi: 10.1109/TNSRE.2022.3230250.

Tangermann, M., Müller, K.-R., Aertsen, A., Birbaumer, N., Braun, C., Brunner, C., Leeb, R., Mehring, C., Miller, K. J., Müller-Putz, G. R., et al. Review of the bci competition iv. *Frontiers in neuroscience*, 6:55, 2012.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L. u., and Polosukhin, I. Attention is all you need. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

Wei, X., Ortega, P., and Faisal, A. A. Inter-subject deep transfer learning for motor imagery eeg decoding. In *2021 10th International IEEE/EMBS Conference on Neural Engineering (NER)*, pp. 21–24, 2021. doi: 10.1109/NER49283.2021.9441085.

Wei, X., Faisal, A. A., Grosse-Wentrup, M., Gramfort, A., Chevallier, S., Jayaram, V., Jeunet, C., Bakas, S., Ludwig, S., Barmpas, K., Bahri, M., Panagakis, Y., Laskaris, N., Adamos, D. A., Zafeiriou, S., Duong, W. C., Gordon, S. M., Lawhern, V. J., Śliwowski, M., Rouanne, V., and Tempczyk, P. 2021 beetl competition: Advancing transfer learning for subject independence and heterogenous eeg data sets. In Kiela, D., Ciccone, M., and Caputo, B. (eds.), *Proceedings of the NeurIPS 2021 Competitions and Demonstrations Track*, volume 176 of *Proceedings of Machine Learning Research*, pp. 205–219. PMLR, 06–14 Dec 2022. URL https://proceedings.mlr.press/v176/wei22a.html.

Wen, Y., Zhang, K., Li, Z., and Qiao, Y. A discriminative feature learning approach for deep face recognition. In *Computer vision–ECCV 2016: 14th European conference, amsterdam, the netherlands, October 11–14, 2016, proceedings, part VII 14*, pp. 499–515. Springer, 2016.

Xu, L., Ma, Z., Meng, J., Xu, M., Jung, T.-P., and Ming, D. Improving transfer performance of deep learning with adaptive batch normalization for brain-computer interfaces. In *2021 43rd Annual International Conference of the IEEE Engineering in Medicine Biology Society (EMBC)*, pp. 5800–5803, 2021. doi: 10.1109/EMBC46164.2021.9629529.

Yi, W., Qiu, S., Wang, K., Qi, H., Zhang, L., Zhou, P., He, F., and Ming, D. Evaluation of eeg oscillatory patterns and cognitive process during simple and compound limb motor imagery. *PLOS ONE*, 9(12):1–19, 12 2014. doi: 10.1371/journal.pone.0114853. URL https://doi.org/10.1371/journal.pone.0114853.

A. Training Details

Table 5. Hyperparmeters for RNAinformer training.

Group	Parameter	Value
Optimizer	Lr	0.0005
	Finetune Lr	0.0005
	weight decay	0.01
	betas	0.9,0.98
	LR schedule	Cosine Annealing
	LR decay factor	0.1
Regularization	Center Loss	0.01
	Center Loss Lr	0.0001
	Label Smoothing	0.50
	Channel Dropout	02-0.3
Model	Model Dimension	80
	Layers	12
	Num head	8
	FeedForward factor	4
	Dropout	0.3-0.5
	Cropped Decoding Window	30

B. Dataset Details

Table 6. MI data sets

MI Data set	Subjects	Channels	Tasks
HGD (Schirrmeister et al., 2017)	14	128	Left/Right hand/Feet/Rest
Cho2017 (Cho et al., 2017)	52	64	Left/Right hand
BNCI2014 (Tangermann et al., 2012)	9	22	Left/Right hand/Feet/Tongue
PhysionetMI (Schalk et al., 2004; Goldberger et al., 2000)	109	64	Left/Right hand/Feet/Both hands/Rest
Weibo2014 (Yi et al., 2014)	10	60	Left/Right hand/Feet/Rest
Cybathlon2020IC (Wei et al., 2021)	5	63	Left/Right hand/Feet/Rest