Quantum Cognition-Inspired EEG-based Recommendation via Graph Neural Networks

Anonymous Author(s)

ACM Reference Format:
Anonymous Author(s). 2023. Quantum Cognition-Inspired EEG-based Recommendation via Graph Neural Networks. In *Proceedings of the 33nd ACM International Conference on Information and Knowledge Management (CIKM '24), October 21-25, 2023, Boise, Idaho, USA*. ACM, New York, NY, USA, 3 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

0.1 Interference between Two Events

An orthogonal matrix $B_{m,i}$ has the property $B_{m,i}B_{m,i}^{\dagger} = \left[|b_1\rangle, |b_2\rangle, \cdots, |b_{|B_{m,i}|}\rangle\right]\left[|b_1\rangle, |b_2\rangle, \cdots, |b_{|B_{m,i}|}\rangle\right]^T = \sum_{j\in|B_{m,i}|}|b_j\rangle\langle b_j| = I = (o_{\Theta_{m,1}} + o_{\bar{\Theta}_{m,1}})$ with conjugate transpose $\dagger$, identity matrix I and $b_j \in \mathbb{R}^{\Lambda\times 1}$. Therefore, the probability $p(\Theta_{m,2})$ can be presented as Eq. (1a)

$$p(\Theta_{m,2}) = \left\|o_{\Theta_{m,2}}|x\rangle\right\|^2 = \left\|o_{\Theta_{m,2}}I|x\rangle\right\|^2 = \left\|o_{\Theta_{m,2}}(o_{\Theta_{m,1}} + o_{\bar{\Theta}_{m,1}})|x\rangle\right\|^2 \tag{1a}$$

$$= \langle x|(o_{\Theta_{m,1}} + o_{\bar{\Theta}_{m,1}})o_{\Theta_{m,2}}o_{\Theta_{m,2}}(o_{\Theta_{m,1}} + o_{\bar{\Theta}_{m,1}})|x\rangle$$

$$= \langle x|o_{\Theta_{m,1}}o_{\Theta_{m,2}}o_{\Theta_{m,2}}o_{\Theta_{m,1}}|x\rangle + \langle x|o_{\bar{\Theta}_{m,1}}o_{\Theta_{m,2}}o_{\Theta_{m,2}}o_{\bar{\Theta}_{m,1}}|x\rangle$$
$$\langle x|o_{\Theta_{m,1}}o_{\Theta_{m,2}}o_{\Theta_{m,2}}o_{\bar{\Theta}_{m,1}}|x\rangle + \langle x|o_{\bar{\Theta}_{m,1}}o_{\Theta_{m,2}}o_{\Theta_{m,2}}o_{\Theta_{m,1}}|x\rangle$$

$$= \left\|o_{\Theta_{m,2}}o_{\Theta_{m,1}}|x\rangle\right\|^2 + \left\|o_{\Theta_{m,2}}o_{\bar{\Theta}_{m,1}}|x\rangle\right\|^2$$
$$+ \langle x|o_{\Theta_{m,1}}o_{\Theta_{m,2}}o_{\Theta_{m,2}}o_{\bar{\Theta}_{m,1}}|x\rangle + \langle x|o_{\bar{\Theta}_{m,1}}o_{\Theta_{m,2}}o_{\Theta_{m,2}}o_{\Theta_{m,1}}|x\rangle \tag{1b}$$

$$= p(\Theta_{m,2})p(\Theta_{m,1}) + p(\Theta_{m,2})p(\bar{\Theta}_{m,1}) \tag{1c}$$

$$+ 2\times\langle x|o_{\bar{\Theta}_{m,1}}o_{\Theta_{m,2}}o_{\Theta_{m,2}}o_{\Theta_{m,1}}|x\rangle \tag{1d}$$

$$= p'(\Theta_{m,2}) + 2\times\tau\langle x|o_{\bar{\Theta}_{m,1}}o_{\Theta_{m,2}}o_{\Theta_{m,2}}o_{\Theta_{m,1}}|x\rangle \tag{1e}$$

In Eq. (1c), $p(\Theta_{m,2})p(\Theta_{m,1})+p(\Theta_{m,2})p(\bar{\Theta}_{m,1})$ that equals $p'(\Theta_{m,2})$ in Eq. (1e) is the probability that represents $\left\|o_{\Theta_{m,2}}o_{\Theta_{m,1}}|x\rangle\right\|^2 + \left\|o_{\Theta_{m,2}}o_{\bar{\Theta}_{m,1}}|x\rangle\right\|^2$ by assuming that events $\Theta_{m,1}$ and $\Theta_{m,2}$ are independent just as used in CPT, while the rest term in Eq. (1d) is an extra value that breaks the assumption of independence, which is reformulated from $\langle x|o_{\Theta_{m,1}}o_{\Theta_{m,2}}o_{\Theta_{m,2}}o_{\bar{\Theta}_{m,1}}|x\rangle + \langle x|o_{\bar{\Theta}_{m,1}}o_{\Theta_{m,2}}o_{\Theta_{m,2}}o_{\Theta_{m,1}}|x\rangle$ because $o_{\Theta_{m,1}}o_{\Theta_{m,2}}o_{\Theta_{m,2}}o_{\bar{\Theta}_{m,1}}$ is the conjugation of $o_{\bar{\Theta}_{m,1}}o_{\Theta_{m,2}}o_{\Theta_{m,2}}o_{\Theta_{m,1}}$ in Eq. (1b). Therefore, a parameter $\tau = 0$ is involved in maintaining independence in Eq. (1e) or $\tau = \pm 1$ reveals the interference between two events. In EEG processing, past thoughts definitely interfere with future thoughts, so τ is set to 1.

CIKM '24, October 21-25, 2024, Boise, Idaho, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM
https://doi.org/10.1145/nnnnnnn.nnnnnnn

0.2 Hyperparameter Analysis

In this section, we discuss the selection and settings of hyperparameters in our experiments. As the datasets exhibit precision@10 disparities, plotting the changes of a hyperparameter together may result in plots resembling mere horizontal lines. To address this issue, we adopt the lowest precision of each dataset in a plot as 0. The remaining precisions are adjusted by subtracting the corresponding lowest precision from each dataset plotted. This approach allows for clearer visualization of hyperparameter effects. Refer to Figure 1 for a visual representation of these hyperparameter impacts.

0.2.1 Impacts of window size Λ. The window size represents the amount of information (*i.e.*,EEG data in this paper) contained in a user's thoughts. In Figure 1(a), there are some suitable window sizes – 10 and 15 for long-tail distribution and normal distribution, respectively. When the window size is too small, the amount of information in the user's thought is insufficient for learning. With the precision difference increasing from the window size 5 to 10, more EEG data are contained in a window to represent a latent thought and contain enough data as a state to learn user preference. When the window size continues to increase to 35, too much EEG data is contained to reflect more complicated thoughts, which makes QUARK hard to learn what thinking factors are in the user's thoughts.

0.2.2 Impacts of sliding size Δ. The sliding size represents the sample rate of the EEG. If the sliding size is smaller than the suitable window size, some overlapped EEG data are sampled as inputs and redundant EEG information impairs learning. On the contrary, if the sliding size is bigger than the suitable window size, some EEG data fragments are not used as inputs to learn the user's thoughts and needs. In Figure 1(b), the suitable sliding sizes are 20 and 25 for long-tail distribution and normal distribution, respectively, which are both bigger than the window sizes. QUARK is not that sensitive to the sliding size as 20 to 25 is a good selection for eight datasets with different distributions and different recommendation granularities. This is because QUARK self-adaptively learns that not all the EEG signals contribute to learning the user's thoughts and needs.

0.2.3 Impacts of the number of the basis vectors $|B_{m,i}|$. The number of basis vectors, $|B_{m,i}|$, shows how many latent thinking factors are contained in a user's thought. In Figure 1(c), the suitable number of basis vectors are 10 and 15 for long-tail distribution and normal distribution, respectively. The number of basis vectors less than the suitable values can not provide enough choices for thought to collapse, while too many basis vectors may make thought collapse to wrong basis vectors.

0.2.4 Impacts of the number c of the selected basis vectors. The number of the selected basis vectors, c, reveals how many latent factors each thought finally decides to collapse. The suitable value of c is 2 for both distributions in Figure 1(d). One thinking factor may not be able to describe a user's feelings, while the number

of thinking factors more than the suitable value may make EEG representation over-smooth technically when computing a new state.

0.2.5 Filter ratios α and β. The filter ratio, α, decides how many weights should be kept in the continuity adjacency matrix. As it is applied to the ReLU function, some weights on the matrices are 0. In Figure 1(e), horizontal lines are shown when the ratios are relatively small because zeros are filtered out with ratio α. Thus, there is no influence on model performance. The suitable values of filter ratios for long-tail distribution and normal distribution are 0.9 and 0.8, respectively. When the values are smaller than the suitable values, some weights revealing irrelevant connections of two events are still used in the learning process. When the values are bigger than the suitable values, not enough connections are kept to learn data patterns. Similarly, the filter ratio, β, decides how many weights should be kept in the interference adjacency matrix showing the same trends in Figure 1(f). The suitable values of filter ratios for long-tail distribution and normal distribution are 0.7, which is a trade-off between more connections and more accurate information.

0.2.6 Impacts of graph depth d. In Figure 1(g), there is one suitable value, $d = 5$, for the graph convolutional network. From 1 to 5, the convolutional depth is not deep enough so some high-order relations and information are not well-studied. From 5 to 9, the deep graph convolutional network makes the representation over-smooth.

0.2.7 Impacts of the ratio ξ. The value of ξ controls the portion of original EEG information x° when the EEG data is involved in graph convolutional computing. The suitable value is 0.3 for both distributions. When the value is smaller than 0.3, more neighbor information is added to the original EEG information so the original EEG information loses its unique data pattern. On the other hand, when the value is from 0.3 to 0.9, too much original EEG information is kept while losing the advantage of capturing the influence of neighbor events.

0.2.8 Conclusion of hyperparameter analysis. We present a detailed analysis of each tunable hyperparameter. In general, QUARK is not a parameter-sensitive model, because the magnitude of the precision difference is 1e-3 to 1e-4. Even though the changes in parameters influence the performance of QUARK, it is not the main reason that our model performs better than baseline models in the comparison section **??**.

0.3 Metrics for Feeling and Style Detection

0.3.1 Content Similarity. The learned representations of images from the image recognition model, ViTAE2, can effectively represent the image contents. So, the similarity between the representation of the original image y_x (a seen image whose corresponding EEG signal is learned by QUARK) and the representation of a recommended image y_k indicates how similar the image content is. The averaged content similarity for all test cases is computed by Eq. (2). To easily filter the recommended images whose similarity scores are lower than a threshold, the Frobenius normalization strategy is considered to scale the values.

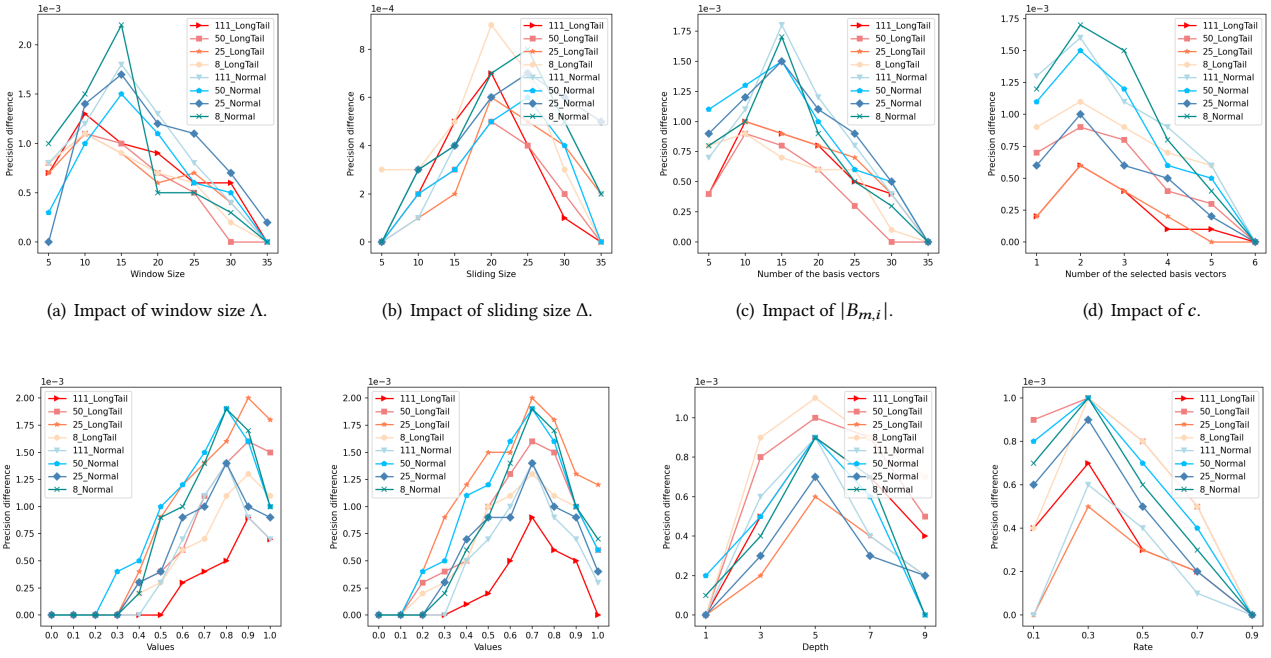
$$ConSim = \frac{1}{|X|}\sum_{x\in X}(\frac{1}{k}\sum_{i\le k}\frac{y_x}{\|y_x\|_F}\frac{(y_k)^T}{\|y_k\|_F}) \quad (2)$$

0.3.2 Color Similarity. An image is composed of three colors, red, green, and blue. Color histogram distribution similarity method [1] compares the distance of two pixels from unlearned image matrices , $y_x^{raw} \in \mathbb{R}^{d\times d}$ and $y_k^{raw} \in \mathbb{R}^{d\times d}$. If the distance of two pixels is large, then there is a small similarity. The averaged color similarity for all test cases is given in Eq. (3).

$$CorSim = \frac{1}{|X|}\sum_{x\in X}\left\{\frac{1}{k}\sum_{i\le k}\sum_{i\le d}\sum_{j\le d}(1-\frac{|(y_x^{raw})_{i,j}-(y_k^{raw})_{i,j}|}{max((y_x^{raw})_{i,j},(y_k^{raw})_{i,j})})\right\} \quad (3)$$

0.3.3 Structural Similarity. Edge detection is a simple but useful way to explore the structure information of an image. Following Vincent's work [2], a white-black image that only contains detected edges with white color (value 255) and non-edges with black color (value 0) can be generated. For the unlearned images $y_x^{raw} \in \mathbb{R}^{d\times d}$ and $y_k^{raw} \in \mathbb{R}^{d\times d}$, edge detection method generates y_x^{edge} and y_k^{edge}. Then, the exclusive OR operation is applied to find the non-overlapped areas, and the summation of the non-overlapped area is normalized by the total area to range $[0, 1]$. The averaged structural similarity is obtained in Eq (4).

$$StrcSim = \frac{1}{|X|}\sum_{x\in X}\left\{\frac{1}{k}\sum_{i\le k}(1-\frac{sum(XOR(y_x^{edge},y_k^{edge}))}{d\times d})\right\} \quad (4)$$



(a) Impact of window size Λ. (b) Impact of sliding size Δ. (c) Impact of $|B_{m,i}|$. (d) Impact of c.

(e) Impact of α. (f) Impact of β. (g) Impact of graph depth d. (h) Impact of ξ.

Figure 1: Plots of hyperparameter analysis. (Note: Axes are labeled in scientific notation.)

REFERENCES

[1] Carol L Novak, Steven A Shafer, et al. 1992. Anatomy of a color histogram.. In *CVPR*, Vol. 92. 599–605.

[2] O Rebecca Vincent, Olusegun Folorunso, et al. 2009. A descriptive algorithm for sobel image edge detection. In *Proceedings of informing science & IT education conference (InSITE)*, Vol. 40. 97–107.