A. Appendix

In this supplementary material, we provide additional details and results that were not included in the main paper but we evaluated during the study. This appendix contains the following items,

- Appendix A.1 Detailed description of the FedMEM algorithm. (Algorithm 3 and Algorithm 4)
- Appendix A.2 Additional details about the experimental setup.
- Appendix A.3 Additional details about the distribution of lifelog data across clients (Fig. 10)
- Appendix A.4 Additional details on the confusion matrix based qualitative analysis of lifelogger's personalized models and global model (FedAvg [30]).
- Appendix A.5 Description and the analysis of the effect of λ_1 and λ_2 when lifeloggers have partial participation.
- Appendix A.6 Analysis of the effect of different number of clusters on FedMEM+MSC.
- Appendix A.7 Additional metrics for the performance analysis of FedMEM with other federated learning approaches (Tab. 3)
- Appendix A.8 Complexity analysis of FedMEM cluster formation.

A.1. Description of FedMEM algorithm

FedMEM algorithm is the personalized federated learning algorithm. We have discussed two type of clustering strategy (1) Memorability Score Distribution based Clustering (MSDC) of lifeloggers, (2) Algorithm 2 Model Similarity-based Clustering (MSC) of lifeloggers.

A.1.1 FedMEM+MSDC

In this Algorithm 3, all available life-loggers first send their memory score distribution (CD_i) to the server for clustering. The server performs KL divergence to determine the memory score similarity between pairs of clients, producing a similarity matrix $KL[,]$. Hierarchical clustering is then performed on this similarity matrix KL, dividing the clients into C_k clusters. Each client initializes their model from their respective cluster, such that if life-logger i belongs to cluster C, the initial model for life-logger i would be W_C. For each cluster, the $Personalized\_Local\_Update$ operation is performed, followed by the aggregation of local models within each cluster. FedMEM+MSDC provides a clustered federated learning approach in which lifeloggers remain in their designated clusters without migrating between them, ensuring that clusters are fully isolated from one another with no cross-cluster movements. Therefore, no cross-cluster knowledge sharing occurs.

Algorithm 3 FedMEM+MSDC

```
Initialize: $\omega_0^0 = \omega_1^0 = \ldots = \omega_C^0 = \omega^0$, $T$, $L$
Server receives distribution of the event memory scores $\{CD_0, CD_1, \ldots, CD_{\mathcal{N}-1}\}$
form $\mathcal{N}$ lifeloggers.
C ← MSDC($CD_0$, $CD_2$, …, $CD_{\mathcal{N}-1}$)                          ▷ Call Algorithm 1
for $t \leftarrow 0$ to $T$ do
    $M \leftarrow Select\_clients(\mathcal{N})$
    Server sends the $\omega_c^t$ to the selected lifeloggers        ▷ $\omega^t$ to the new lifelogger
    for $M$ lifeloggers in parallel do        ▷ Lifelogger assigned to any of the $k$
clusters. c ∈ C
        $\theta_i$ = Personalized_Local_Update($\omega_c^t$)
    end for
    All lifeloggers send $\theta_i^t$ to their respective clusters.
    for $Lg_i \in M$ do in all $C$ clusters perform the cluster update
        $\omega_c^t \leftarrow \omega_c^t + r_i \theta_i^t$                                   ▷ Equation (8)
    end for
end for
procedure PERSONALIZED_LOCAL_UPDATE($\bar{\omega}$)
    for $l \leftarrow 0$ to $L$ do
        $\theta_i^{t,l+1} = \theta_i^{t,l} - \alpha_i \nabla f_i(\theta_i^{t,l}) - \alpha_i \eta (\theta_i^{t,l} - \omega_c^t)$.      ▷ Equation (6)
    end for
end procedure
```

A.1.2 FedMEM+MSC

In Algorithm 4, initially, all available life-loggers do not have any information about the clusters. They start by initializing their local models with the same global model. In the initial round (t=0), the server sends the global model ω^0 to the selected life-loggers. In subsequent rounds, if a life-logger has been assigned to a cluster, the server sends the cluster model to that life-logger. If a new life-logger joins in a later round and does not belong to any cluster, their model is initialized with the global model. Similar to Algorithm 3, clients perform the $Personalized\_local\_update$ operation to train their local models. Each life-logger then sends their trained local model to the server, where a similarity matrix is created using Algorithm 2. Spectral clustering is then performed to identify clusters. After cluster assignments are made, a cluster update is performed, and a new global model is created. This process continues until the personalized models of all life-loggers converge.

A.2. Experimental setup

For each lifelogger, we split the data into train:val:test in 6:3:1 ratios. This was carried out for all the lifeloggers. During FL training iterations, only the train set from each lifelogger is utilized. After training is over, the personalized model for each individual lifelogger is employed to infer the memorability of images in the lifelogger's own test set images. The overall F1-score is calculated by considering the inferred and ground-truth memorability on all the test set images from all lifeloggers. Cross-cluster/Cross-user test-

Algorithm 4 FedMEM+MSC

```
Initialize: $\omega_0^0 = \omega_1^0 = \ldots = \omega_C^0 = \omega^0, T, L$
for $t \leftarrow 0$ to $T$ do
    $M \leftarrow Select\_clients(\mathcal{N})$
    Server sends the $\omega_c^t$ or $\omega^t$ to the selected lifeloggers    ▷ $\omega^t$ to the new lifelogger
    for $M$ lifeloggers in parallel do
        if $Lg_i \notin C$ then    ▷ Lifelogger not assigned to any cluster
            $\theta_i$ = Personalized_Local_Update($\omega^t$)
        else    ▷ Lifelogger assigned to any of the $k$ clusters. c ∈ C
            $\theta_i$ = Personalized_Local_Update($\omega_c^t$)
        end if
    end for
    All lifeloggers send $\theta_i^t$ to the server.
    $C \leftarrow MSC(\theta_1^t, \theta_2^t, \ldots, \theta_M^t)$    ▷ Call Algorithm 2
    for $Lg_i \in M$ do in all $C$ clusters perform the cluster update
        $\omega_c^t \leftarrow \omega_c^t + r_i \theta_i^t$    ▷ Equation (8)
    end for
    for all $c \in C$ in parallel do    ▷ Only for MSC
        $\omega^t = \omega^t + r_c \omega_c^t$    ▷ Equation (10)
    end for
end for
procedure PERSONALIZED_LOCAL_UPDATE($\bar{\omega}$)
    for $l \leftarrow 0$ to $L$ do
        $\theta_i^{t,l+1} = \theta_i^{t,l} - \alpha_i \nabla f_i(\theta_i^{t,l}) - \alpha_i \eta (\theta_i^{t,l} - \omega_c^t)$.    ▷ Equation (6)
    end for
end procedure
```

ing was not carried out as our personalized approach produced individual models for every lifelogger.

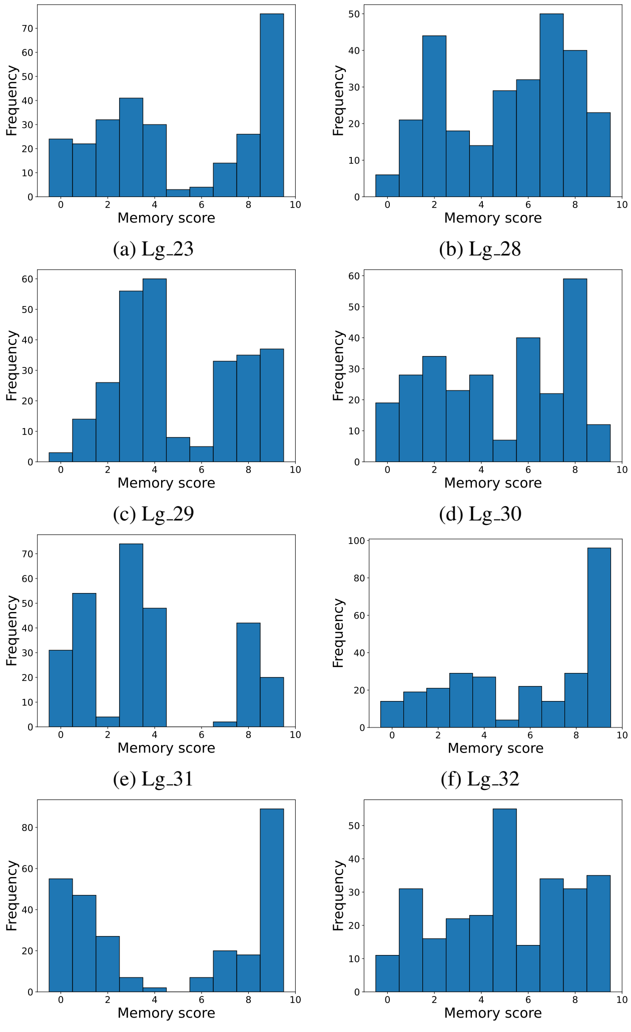
We utilized the same frozen ResNet50 as the baseline CEMNET [40]. Because we want to demonstrate the advantage of our personalized federated learning approach over the centralized CEMNET model.

A.3. Distribution of lifeloggers data

In this supplementary copy we provide the data distributions of some more lifeloggers Lg_{23} (Figure 10a), Lg_{28} (Figure 10b) to Lg_{34} (Figure 10h). We observed the data distribution across lifeloggers, and we noted that some lifeloggers do not have any data for certain memory scores (such as, for Lg_{31}, none of their images had memory scores of 5 and 6). This showed us that the memorability scores provided by the lifeloggers do not follow the same distribution. Therefore, we believe that there is case of non-IID (not independent and identical) distribution of data across lifeloggers.

A.4. Qualitative analysis of lifelogger's personalized models and global model (FedAvg)

We also provide a confusion matrix-based analysis for each qualitative analysis of lifeloggers in Figure 11 . The color intensity represents the classifier's predictions. Darker colors usually indicate higher numbers. This means that areas in the matrix where the model made more predictions (correct or incorrect) will be highlighted with a darker shade. Lighter colors typically represent lower numbers. These areas indicate fewer predictions for those class combinations. The y-axis (rows) represents the actual classes, and the x-axis (columns) represents the predicted classes.



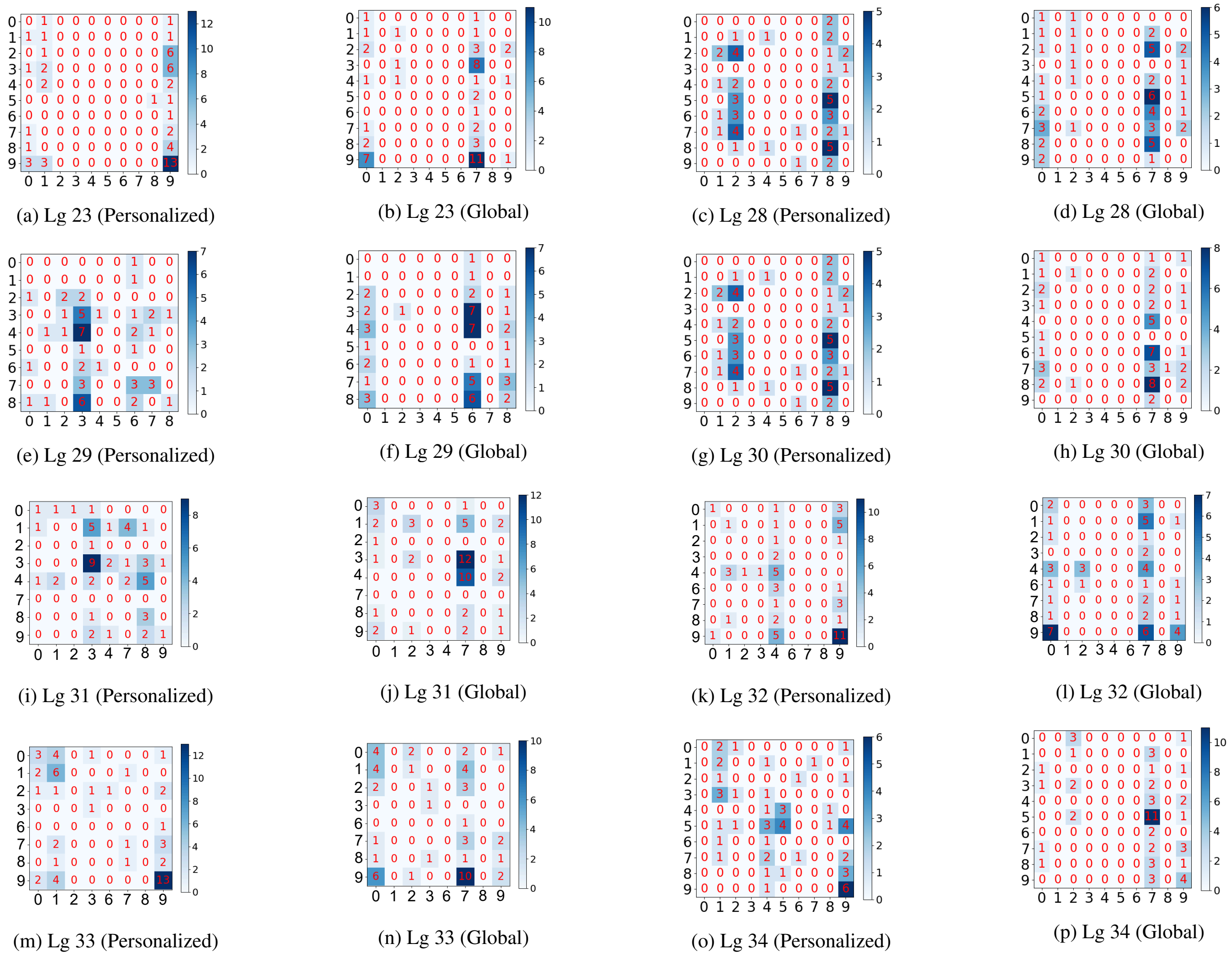
(a) Lg_23 (b) Lg_28 (c) Lg_29 (d) Lg_30 (e) Lg_31 (f) Lg_32 (g) Lg_33 (h) Lg_34

Figure 10. Difference in histograms of memory score distribution in individual lifeloggers (Lg)

We observed from the Figure 11 that the personalized model produces more true positive predictions, whereas the global model makes more false positive predictions. This further supports our claim that the personalized model is more effective than the global model for all the lifeloggers.

A.5. Ablation of the value of λ_1 and λ_2

We trained both FedMEM+MSC and MSDC with 5 clusters. We defined hyperparameters λ_1 and λ_2 in the range [0,1], ensuring $\lambda_1 + \lambda_2 \leq 1$. These parameters regularize the similarity measure. Setting both to 0 focuses on client similarity. Increasing λ_1 emphasized the global model while increasing λ_2 emphasized the clustered model. For equal consideration of both cluster and global models, λ_1 and λ_2 are set equally. In the experiments in the submitted paper we prioritized client similarity and set λ_1 and λ_2 to 0.25.



(a) Lg 23 (Personalized) (b) Lg 23 (Global) (c) Lg 28 (Personalized) (d) Lg 28 (Global)

(e) Lg 29 (Personalized) (f) Lg 29 (Global) (g) Lg 30 (Personalized) (h) Lg 30 (Global)

(i) Lg 31 (Personalized) (j) Lg 31 (Global) (k) Lg 32 (Personalized) (l) Lg 32 (Global)

(m) Lg 33 (Personalized) (n) Lg 33 (Global) (o) Lg 34 (Personalized) (p) Lg 34 (Global)

Figure 11. Personalized model (FedMEM) compared to global model (FedAvg [30]) for selected lifeloggers (Lg). Global model favors higher memorability scores but FedMEM is able to capture the characteristics of each lifelogger's memorability score distribution resulting in improved F1 score.

In Figure 12, we performed a comparative analysis of λ_1 and λ_2 to examine the impact of global and cluster models on clustering, which is reflected in the effectiveness of FedMEM. The model was trained for 30 global rounds, with 50% of the lifeloggers participating in each round. We found that increasing the value of λ_1 improves the performance of FedMEM. Furthermore, when λ_1 and λ_2 are in the range of $(0.25, 0.75)$, the performance of FedMEM remains relatively consistent.

A.6. Ablation on the effect of clusters on FedMEM+MSC

In Figure 13, we performed the experiment across 40 life loggers, but 50% of them were available at each global round. We train for 30 global rounds. The values of the hyperparameters λ_1 and λ_2 are 0.25. We observed if we increase the number of clusters, the performance of FedMEM decreases. But the decrease is not significant.

A.7. Performance analysis with the state-of-the-art

In Table 3, we present a comparative analysis of the performance of FedMEM with different state-of-the-art Federated Learning algorithms. In the main article, we had given only one performance metric, which is the F1-score. Therefore, in this supplementary, we provide additional metrics here for comparison such as the Precision, Recall, weighted-F1 score, and the Mean Absolute Error (MAE). Since the memory score is an ordinal number, we obtain the absolute value of the error between the predicted and ground-truth memorability scores. Then we calculate the average value of this absolute error during inference time over the images in the testset, which is presented as MAE.

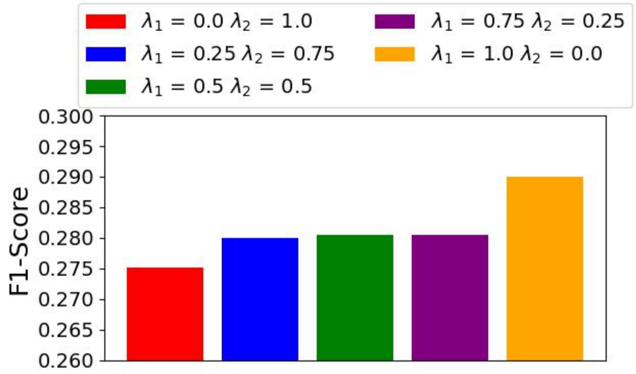


Figure 12. Ablation on value of λ_1 and λ_2

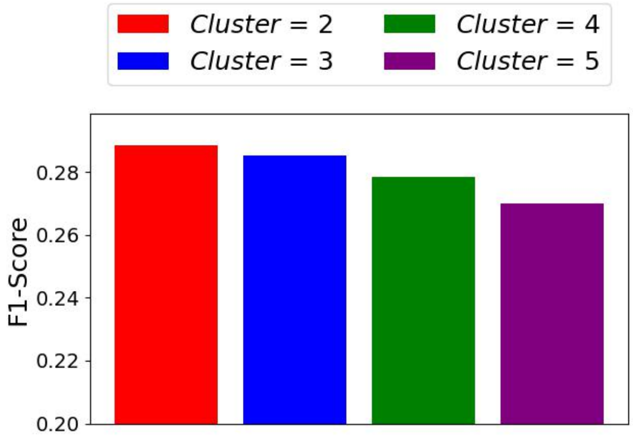


Figure 13. Ablation on the number of clusters

From Table 3 we observed FedMEM+MSC performance is better than the other FL algorithms.

A.8. Complexity analysis

FedMEM algorithm has three parts: local update, cluster formation, and global update.

MSDC: When we are using MSDC, the cluster formation is pre-defined before the start of any global update. We used agglomeration clustering to cluster clients. Agglomerative hierarchical clustering starts with each lifelogger as its own cluster and iteratively merges the closest clusters until all points are in a single cluster or until a desired number of clusters is reached. The agglomerative clustering algorithm can be broken down into three main parts:

1. Initial pairwise distance calculations At the start, the algorithm calculates pairwise distances between all N lifelogger's memory score to create a distance matrix. The number of unique pairs of points is

$$\binom{N}{2} = \frac{N(N-1)}{2},$$

Calculating the distance for each pair takes constant time. Time Complexity for the initial pairwise distance calculations is $O(N^2)$

2. Finding the closest clusters to merge In each iteration, the algorithm identifies the two closest clusters based on the computed distance matrix. For this, it scans the entire distance matrix, which contains $O(N^2)$ elements in the first iteration. After the first merge, the number of clusters decreases by one, and so on, until only one cluster remains. There are $N-1$ iterations, one for each merge step. Searching for the closest pair at each step requires scanning the distance matrix, which takes $O(N^2)$ time. Therefore, for $N-1$ steps, finding the closest clusters will take:

$$O(N^2) + O((N-1)^2) + O((N-2)^2) + \cdots + O(1^2)$$

This results in an overall complexity of $O(N^3)$.

3. Updating the distance matrix After merging two clusters, the algorithm updates the distance matrix to reflect the distances between the newly formed cluster and all other clusters. This involves recalculating the distances for the new cluster, which takes $O(N)$ operations.

There are $N-1$ merge steps, so updating the distance matrix requires:

$$O(N) + O(N-1) + \cdots + O(1) = O(N^2)$$

Total time complexity of MSDC The total complexity of basic agglomerative clustering is the sum of the complexities from the three components is $O(N^2)$, $O(N^3)$, and $O(N^2)$. Therefore, the overall time complexity of agglomerative clustering is dominated by the complexity of finding the closest clusters in each step, which results in $O(n^3)$.

MSC: When we are using MSC, cluster formation happens after every global iteration.

1. Similarity-matrix computation: After N lifeloggersperforms personalized local updates, they compute the similarity matrix using Equation (1) and apply spectral clustering to form clusters. Computing the pairwise distances between N lifeloggers requires $O(N^2)$ operations, as it involves calculating $\binom{N}{2}$ distances. Therefore, the complexity of constructing the similarity matrix is, $O(N^2)$.

Table 3. Comparison of FedMEM with different FL algorithms

Method					**Precision**	**Recall**	**F1-Score (weighted)**	**Mean Absolute Error**
Non Federated	Centralized Model [40] (Image-only baseline)				17.10	17.90	-	3.03
	Siloed models per lifelogger				21.10	29.9	24.74	2.30
Federated	Non Clustered		Non Personalized	FedAvg [30]	12.27	16.49	14.07	3.69
				FedProx [25]	12.22	16.31	13.97	3.78
			Personalized	pFedMe [38]	15.60	19.80	17.45	3.39
	Clustered	Apriori	Non Personalized	h-SGD [26]	07.00	20.39	10.42	3.96
		Dynamic		FeSEM [27]	03.37	11.46	05.21	4.06
		Dynamic	Personalized	DemLearn [32]	**21.54**	28.86	24.67	2.59
		Apriori		FedMEM+ MSDC [Ours]	21.30	28.69	24.45	2.68
		Dynamic		FedMEM+ MSC [Ours]	19.86	**34.47**	**25.20**	**2.28**

2. Constructing the Laplacian Matrix: Once the similarity matrix is done, the **graph Laplacian** L is computed. $L = D - W$, where D is the degree matrix and W is the similarity matrix. The time complexity to calculate the degree matrix D is simply the row sums of W, so constructing L takes $O(N^2)$ time, since we are just performing element-wise operations on the similarity matrix W. Overall Complexity for this Step is $O(N^2)$.

3. Eigenvalue Decomposition Spectral clustering requires computing the first C eigenvectors of the Laplacian matrix, where C is the number of clusters. For a dense $N \times N$ matrix where all lifeloggers are participating, the cost of eigenvalue decomposition is $O(N^3)$ in the worst case. eigenvalue decomposition can be reduced up to $O(C \cdot N^2)$, if we used iterative methods such as Lanczos algorithm [22]

4. Clustering in the Reduced Space After computing the top C eigenvectors, spectral clustering applies a standard clustering algorithm such as k-means to the eigenvector matrix $U \in \mathbb{R}^{N \times C}$, which reduces the original data to a lower-dimensional space. Running k-means on N lifeloggers in C-dimensional space takes $O(N \cdot C \cdot t)$, where t is the number of iterations until convergence. In practice, t is small, so this step is relatively efficient compared to the eigenvalue decomposition step. Overall Complexity for this Step is $O(N \cdot C \cdot t)$

Total time Complexity of MSC The time complexity of spectral clustering is the sum of the complexities of the above steps: $O(N^2)$, $O(N^2)$, $O(C \cdot N^2)$ or $O(N^3)$ and $O(N \cdot C \cdot t)$. The dominant term is the eigenvalue decomposition, making the overall complexity $O(N^3)$ or $O(C \cdot N^2)$ with iterative methods.

Comparison of the time complexity of FedMEM cluster formation with state-of-the-art In the worst case, both MSC and MSDC of FedMEM have time complexity $O(N^3)$. If we can optimize the clustering in MSC by using [22] that reduces time complexity up to $O(C \cdot N^2)$. [32] uses a hierarchical method which produces the overall complexity of $O(N^3)$. Cluster assignment in FeSEM [27] has time complexity of $O(N \cdot C \cdot p)$ where p is the number of parameters in the model.

References

[1] Safaa Azzakhnini, Olfa Ben-Ahmed, and Christine Fernandez-Maloigne. Video memorability prediction using deep features and loss-based memorability distribution estimation. In *MediaEval Benchmarking Initiative for Multimedia Evaluation 2022*, 2022. 3

[2] Wilma A Bainbridge, Phillip Isola, and Aude Oliva. The intrinsic memorability of face photographs. *Journal of Experimental Psychology: General*, 142(4):1323, 2013. 3

[3] Wilma A Bainbridge and Jesse Rissman. Dissociating neural markers of stimulus memorability and subjective recognition during episodic retrieval. *Scientific Reports*, 8(1):8679, 2018. 3

[4] Sourasekhar Banerjee, Ali Dadras, Alp Yurtsever, and Monowar Bhuyan. Personalized multi-tier federated learning. *arXiv preprint arXiv:2407.14251*, 2024. 3

[5] Christopher Briggs, Zhong Fan, and Peter Andras. Federated learning with hierarchical clustering of local updates to improve training on non-iid data. In *2020 International Joint Conference on Neural Networks (IJCNN)*, pages 1–9. IEEE, 2020. 3

[6] Romain Cohendet, Claire-Hélène Demarty, Ngoc QK Duong, and Martin Engilberge. Videomem: Constructing, analyzing, predicting short-term and long-term video memorability. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2531–2540, 2019. 3

[7] Aiden R Doherty, Katalin Pauly-Takacs, Niamh Caprani, Cathal Gurrin, Chris JA Moulin, Noel E O'Connor, and Alan F Smeaton. Experiences of aiding autobiographical memory using the sensecam. *Human–Computer Interaction*, 27(1-2):151–174, 2012. 3

[8] Moming Duan, Duo Liu, Xinyuan Ji, Renping Liu, Liang Liang, Xianzhang Chen, and Yujuan Tan. Fedgroup: Efficient federated learning via decomposed similarity-based clustering. In *2021 IEEE Intl Conf on Parallel & Distributed Processing with Applications, Big Data & Cloud Computing,*

Sustainable Computing & Communications, Social Computing & Networking (ISPA/BDCloud/SocialCom/SustainCom), pages 228–237. IEEE, 2021. 3

[9] Jiri Fajtl, Vasileios Argyriou, Dorothy Monekosso, and Paolo Remagnino. Amnet: Memorability estimation with attention. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6363–6372. IEEE, 2018. 3

[10] Md Sadek Ferdous, Soumyadeb Chowdhury, and Joemon M Jose. Analysing privacy in visual lifelogging. *Pervasive and Mobile Computing*, 40:430–449, 2017. 3

[11] Ana Garcia del Molino, Joo-Hwee Lim, and Ah-Hwee Tan. Predicting visual context for unsupervised event segmentation in continuous photo-streams. In *Proceedings of the 26th ACM international conference on Multimedia*, pages 10–17, 2018. 3

[12] Avishek Ghosh, Jichan Chung, Dong Yin, and Kannan Ramchandran. An efficient framework for clustered federated learning. *Advances in Neural Information Processing Systems*, 33:19586–19597, 2020. 3

[13] Morgan Harvey, Marc Langheinrich, and Geoff Ward. Remembering through lifelogging: A survey of human memory augmentation. *Pervasive and Mobile Computing*, 27:14–26, 2016. 1, 3

[14] Li Huang, Andrew L Shea, Huining Qian, Aditya Masurkar, Hao Deng, and Dianbo Liu. Patient clustering improves efficiency of federated machine learning to predict mortality and hospital stay time using distributed electronic medical records. *Journal of biomedical informatics*, 99:103291, 2019. 3

[15] Phillip Isola, Devi Parikh, Antonio Torralba, and Aude Oliva. Understanding the intrinsic memorability of images. *Advances in neural information processing systems*, 24, 2011. 3

[16] Phillip Isola, Jianxiong Xiao, Devi Parikh, Antonio Torralba, and Aude Oliva. What makes a photograph memorable? *IEEE transactions on pattern analysis and machine intelligence*, 36(7):1469–1482, 2013. 3

[17] Peiguang Jing, Yuting Su, Liqiang Nie, and Huimin Gu. Predicting image memorability through adaptive transfer learning from external sources. *IEEE Transactions on Multimedia*, 19(5):1050–1062, 2016. 3

[18] Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank Reddi, Sebastian Stich, and Ananda Theertha Suresh. Scaffold: Stochastic controlled averaging for federated learning. In *International conference on machine learning*, pages 5132–5143. PMLR, 2020. 3

[19] Aditya Khosla, Akhil S Raju, Antonio Torralba, and Aude Oliva. Understanding and predicting image memorability at a large scale. In *Proceedings of the IEEE international conference on computer vision*, pages 2390–2398. IEEE, 2015. 3

[20] Griffin E Koch, Essang Akpan, and Marc N Coutanche. Image memorability is predicted by discriminability and similarity in different stages of a convolutional neural network. *Learning & Memory*, 27(12):503–509, 2020. 3

[21] Dmitry Kuzovkin, Tania Pouli, Rémi Cozot, Olivier Le Meur, Jonathan Kervec, and Kadi Bouatouch. Image selection in photo albums. In *Proceedings of the 2018 ACM on International Conference on Multimedia Retrieval*, pages 397–404, 2018. 1

[22] Cornelius Lanczos. An iteration method for the solution of the eigenvalue problem of linear differential and integral operators. *Journal of Research of the National Bureau of Standards*, 45:255–282, 1950. 13

[23] Matthew L Lee and Anind K Dey. Providing good memory cues for people with episodic memory impairment. In *Proceedings of the 9th international ACM SIGACCESS conference on Computers and accessibility*, pages 131–138. ACM, 2007. 1

[24] Qinbin Li, Bingsheng He, and Dawn Song. Model-contrastive federated learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10713–10722, 2021. 3, 6, 7, 8

[25] Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. Federated optimization in heterogeneous networks. *Proceedings of Machine learning and systems*, 2:429–450, 2020. 3, 6, 7, 8, 13

[26] Lumin Liu, Jun Zhang, SH Song, and Khaled B Letaief. Client-edge-cloud hierarchical federated learning. In *ICC 2020-2020 IEEE international conference on communications (ICC)*, pages 1–6. IEEE, 2020. 2, 6, 7, 8, 13

[27] Guodong Long, Ming Xie, Tao Shen, Tianyi Zhou, Xianzhi Wang, and Jing Jiang. Multi-center federated learning: clients clustering for better personalization. *World Wide Web*, 26(1):481–500, 2023. 2, 3, 5, 7, 8, 13

[28] Jiaxin Lu, Mai Xu, Ren Yang, and Zulin Wang. Understanding and predicting the memorability of outdoor natural scenes. *IEEE Transactions on Image Processing*, 29:4927–4941, 2020. 3

[29] Yishay Mansour, Mehryar Mohri, Jae Ro, and Ananda Theertha Suresh. Three approaches for personalization with applications to federated learning. *arXiv preprint arXiv:2002.10619*, 2020. 3

[30] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, pages 1273–1282. PMLR, 2017. 2, 3, 6, 7, 8, 9, 11, 13

[31] F Milton, Nils Muhlert, CR Butler, A Benattayallah, and Adam Z Zeman. The neural correlates of everyday recognition memory. *Brain and Cognition*, 76(3):369–381, 2011. 3

[32] Minh NH Nguyen, Shashi Raj Pandey, Tri Nguyen Dang, Eui-Nam Huh, Nguyen H Tran, Walid Saad, and Choong Seon Hong. Self-organizing democratized learning: Toward large-scale distributed learning systems. *IEEE Transactions on Neural Networks and Learning Systems*, 2022. 2, 3, 5, 6, 7, 8, 13

[33] Jesse Rissman, Tiffany E Chow, Nicco Reggente, and Anthony D Wagner. Decoding fmri signatures of real-world autobiographical memory retrieval. *Journal of cognitive neuroscience*, 28(4):604–620, 2016. 3

[34] Felix Sattler, Klaus-Robert Müller, and Wojciech Samek. Clustered federated learning: Model-agnostic distributed multitask optimization under privacy constraints. *IEEE transactions on neural networks and learning systems*, 32(8):3710–3722, 2020. 3

[35] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *IEEE Transactions on pattern analysis and machine intelligence*, 22(8):888–905, 2000. 5

[36] AR Silva, Maria Salomé Pinho, Luis Macedo, and CJA Moulin. A critical review of the effects of wearable cameras on memory. *Neuropsychological Rehabilitation*, 28(1):117–141, 2018. 1

[37] Hammad Squalli-Houssaini, Ngoc QK Duong, Marquant Gwenaëlle, and Claire-Hélène Demarty. Deep learning for predicting image memorability. In *2018 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 2371–2375. IEEE, 2018. 3

[38] Canh T Dinh, Nguyen Tran, and Josh Nguyen. Personalized federated learning with moreau envelopes. *Advances in Neural Information Processing Systems*, 33:21394–21405, 2020. 2, 3, 6, 7, 8, 13

[39] Alice Thudt, Dominikus Baur, Samuel Huron, and Sheelagh Carpendale. Visual mementos: Reflecting memories with personal data. *IEEE transactions on visualization and computer graphics*, 22(1):369–378, 2015. 1

[40] Qianli Xu, Fen Fang, Ana Molino, Vigneshwaran Subbaraju, and Joo-Hwee Lim. Predicting event memorability from contextual visual semantics. *Advances in Neural Information Processing Systems*, 34:22431–22442, 2021. 1, 2, 3, 5, 6, 7, 8, 10, 13

[41] Qianli Xu, Vigneshwaran Subbaraju, Chee How Cheong, Aijing Wang, Kathleen Kang, Munirah Bashir, Yanhong Dong, Liyuan Li, and Joo-Hwee Lim. Personalized serious games for cognitive intervention with lifelog visual analytics. In *Proceedings of the 26th ACM international conference on Multimedia*, pages 328–336, 2018. 1