Debiasing Global Workspace: A Cognitive Neural Framework for Learning Debiased and Interpretable Representations

Jinyung Hong[1] **Eun Som Jeon**[2] **Changhoon Kim**[1] **Keun Hee Park**[1]

Utkarsh Nath[1] **Yezhou Yang**[1] **Pavan Turaga**[3] **Theodore P. Pavlic**[1,4]

[1]**School of Computing and Augmented Intelligence**
[3]**School of Arts, Media, and Engineering**
[4]**School of Life Sciences**
Arizona State University, Tempe, AZ 85281, USA

[2]**Department of Computer Science and Engineering**
Seoul National University of Science and Technology, Seoul, 01811, Korea

Abstract

Deep Neural Networks (DNNs) often make predictions based on "spurious" attributes when trained on biased datasets, where most samples have features spuriously correlated with the target labels. This can be problematic if irrelevant features are easier for the model to learn than the truly relevant ones. Existing debiasing methods require predefined bias labels and entail computational complexity with additional networks. We propose an alternative approach inspired by cognitive science, called *Debiasing Global Workspace* (DGW). DGW consists of specialized modules and a shared workspace, allowing for increased modularity and improved debiasing performance. Additionally, our method enhances the transparency of decision-making processes through attention masks. We validate DGW across various biased datasets, proving its effectiveness in better debiasing performance.

1 Introduction

Despite their remarkable performance across many domains [23, 66, 64, 32, 44, 70], Deep Neural Networks (DNNs) often show poor performance, lacking generalization capability to out-of-distribution (OOD) data and robustness to biases in their training datasets [58]. These biases occur when irrelevant features, such as background color, correlate with target labels, causing the models to rely on these features for making predictions [14]. Specifically, given the biased datasets that possess many *bias-aligned* samples (irrelevant features strongly correlate with the labels) and a small number of *bias-conflicting* samples (the features that do not align with the labels), models trained on such data indeed tend to focus on the bias-aligned samples, leading to poor generalization [26, 24].

Various debiasing approaches have been proposed to prevent a model from relying on spurious correlations when trained on a biased dataset, such as using auxiliary models[49, 55], re-weighting samples [42, 49], data augmentation [34, 38], and leveraging biased labels [28, 33, 39, 54]. However, they struggle with insufficiently diverse samples and require accurate bias identification with manual labeling [4, 57].

NeurIPS 2024 Workshop on Behavioral Machine Learning.

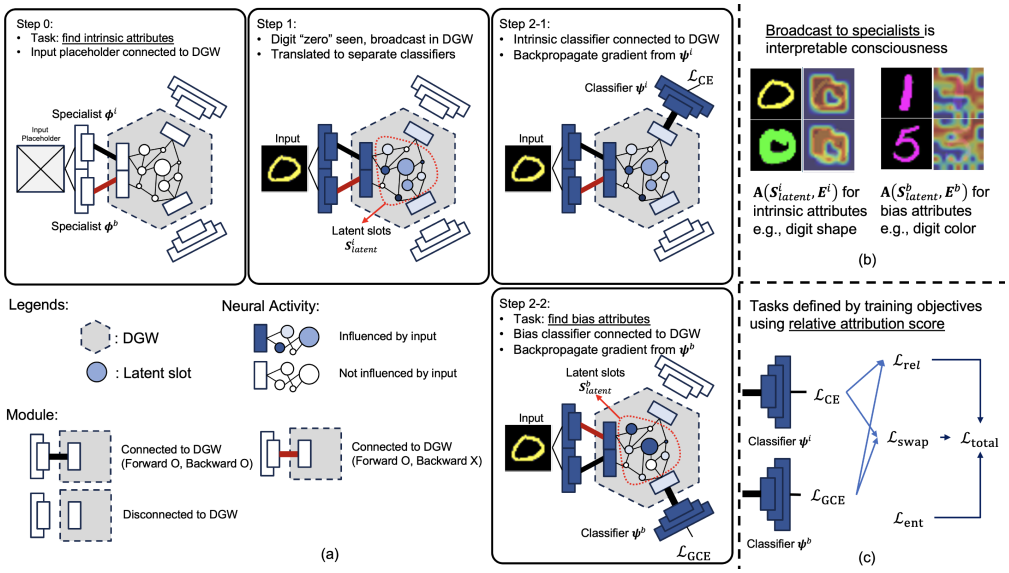


Figure 1: The conceptual framework of Debiasing Global Workspace (DGW). (a) When attention selects inputs from specialists (Step 0), its latent space activation is copied into the DGW and immediately translated into representations suitable for each module (Step 1). We control which module is mobilized into the workspace to receive and process the corresponding data effectively. The classifier ψ^i is initiated for intrinsic attributes (Step 2-1), and the classifier ψ^b is activated for learning bias attributes (Step 2-2). (b) Broadcast: The information broadcast in DGW can demonstrate interpretable representation for attribute learning. (c) Unlike the original GWT, where task definitions can be preset in Step 0, we address them using our training objectives using relative attribution score. The generic figure is inspired by [60, Fig. 3], and more details are described in Appendix C.1.

In this work, we focus on a completely different way to implement a novel debiasing framework, inspired by Global Workspace Theory (GWT). While GWT enables to modeling of human consciousness arising from integrating and broadcasting information across specialized, unconscious processes in the brain [2, 3], many recent studies proposing a deep-learning implementation of GWT [5, 18, 60, 27] have shown their efficacy in allowing a model to have general-purpose functionality, increased modularity, improved performance, and interpretable representation learning. Thus, we propose a novel GWT instantiation for debiasing, *Debiasing Global Workspace* (DGW), to eliminate the negative effect of the misleading correlations. Our debiasing approach involves specialized modules (acting as the specialists in GWT) and an attention-based information bottleneck (acting as the global workspace in GWT), allowing for achieving straightforward, functional modularity and effective debiasing performance while providing interpretable representation by visualizing which attributes are essential for accurate predictions and which are irrelevant and likely to cause errors.

2 Roadmap to Implement Debiasing Global Workspace

This section presents the DGW implementation, which combines existing deep learning components for effective debiasing frameworks while adhering to neuroscience findings. Additional implementation details are in Appendix C.

Two specialized modules and the shared workspace. DGW uses two independent specialists: the intrinsic attribute encoder ϕ^i and the bias attribute encoder ϕ^b, and produce the concatenated features $\mathbf{E} = [\phi^i(\mathbf{x}); \phi^b(\mathbf{x})] \in \mathbb{R}^{L \times D}$. To connect specialists and the shared workspace, we define $\mathbf{e} \in \{\mathbf{E}^i, \mathbf{E}^b\}$, where $\mathbf{E}^i = [\phi^i(\mathbf{x}); \mathrm{sg}(\phi^b(\mathbf{x}))]$ and $\mathbf{E}^b$ is vice versa, with $\mathrm{sg}(\cdot)$ as the stop-gradient operator. Additionally, we introduce the Global Latent Attention (GLA) module, which acts as a shared workspace that encourages the synchronization among the input feature vector $\mathbf{E}$ via a latent feature representation $\mathbf{S}_{\text{latent}}$.

Latent-slot binding specific to each input. The GLA module uses a set number of latent embeddings or latent slots C. These latent slots represent the learnable embedding vectors in the DGW, and perform competitive attention [61] on the input features $\mathbf{e}$. We define $\mathbf{s}_{\text{latent}} \in \{\mathbf{S}^i_{\text{latent}}, \mathbf{S}^b_{\text{latent}}\} \in \mathbb{R}^{C\times D}$ where C^i is number of slots for intrinsic features and C^b for bias features, with $C = C^i + C^b$. The attention mechanism uses the following equation:

$$\mathbf{A}(\mathbf{e}, \mathbf{s}_{\text{latent}}) = \text{softmax}\left(\frac{k(\mathbf{e}) \cdot q(\mathbf{s}_{\text{latent}})^\top}{\sqrt{D}}\right) \in \mathbb{R}^{C\times L}, \tag{1}$$

where, k, q are linear projection matrices, and the softmax function normalizes the slots, creating competition among them. The slots are refined iteratively using the following:

$$\mathbf{s}^{(n+1)}_{\text{latent}} = \text{GRU}\left(\mathbf{s}^{(n)}_{\text{latent}}, \text{Normalize}\left(\mathbf{A}(\mathbf{e}, \mathbf{s}^{(n)}_{\text{latent}})^\top\right) \cdot v\left(\mathbf{E}\right)\right), \tag{2}$$

where, $\mathbf{s}^{(n)}_{\text{latent}}$ is the latent slot representation after n iterations, GRU [8] is a recurrent neural network, and v is another liner projection matrix. The initial slots $\mathbf{s}^{(0)}_{\text{latent}}$ are initialized with learnable queries following [30]. The above computations can be considered to implement a shared global workspace in [18, 27] because they enable different parts of the model to compete for attention, integrating and broadcasting information similar to GWT.

Broadcast updated information to specialists. Specialists update their states using information from the shared workspace. The inverted cross-attention mechanism allows specialists to query and interact with updated latent slots $\mathbf{s}^{(n+1)}_{\text{latent}}$, updating their states through: $\bar{\mathbf{e}} = \mathbf{e} \oplus \left(\mathbf{A}\left(\mathbf{s}^{(n+1)}_{\text{latent}}, \mathbf{e}\right) \cdot v\left(\mathbf{s}^{(n+1)}_{\text{latent}}\right)\right) \in \mathbb{R}^{L\times D}$ where v is a linear projection matrix. Here, as the meaning of information broadcast, $\oplus$ can be instantiated with various computational operations, including a residual connection [22].

In GWT, the information broadcast through the global workspace is a necessary and sufficient condition for conscious perception [60]. Intuitively, the attention mask $\mathbf{A}(\mathbf{s}^{(n+1)}_{\text{latent}}, \mathbf{e})$ can be seen as artificial phenomenal consciousness, indicating the immediate subjective experience of sensations and perceptions.

Training Objectives. We have two linear classifiers $\boldsymbol{\psi}^i$ and $\boldsymbol{\psi}^b$ that take the updated concatenated vector $\bar{\boldsymbol{e}}$ from the previous module as input to predict the target label y. Our training objectives consist of: i) the relative attribute score learning phase, and ii) the attribute composition phase.

For the relative attribute score learning phase, we define two tasks: identifying intrinsic attributes and biased attributes within the conceptual framework. Without specific information about bias types, we utilize the relative difficulty score of each data sample, referring to [49]. Specifically, we train $\boldsymbol{\phi}^b$, $\mathbf{S}^b_{\text{latent}}$, and $\boldsymbol{\psi}^b$ to focus on bias attributes using generalized cross entropy (GCE) [68], while $\boldsymbol{\phi}^i$, $\mathbf{S}^i_{\text{latent}}$ and $\boldsymbol{\psi}^i$ are trained with the cross entropy (CE) loss. Samples with high CE loss from $\boldsymbol{\psi}^b$ are considered bias-conflicting compared to those with low CE loss. Based on the definition of the relevance score function: $\text{Score}(\bar{\boldsymbol{e}}, y) \triangleq CE(\boldsymbol{\psi}^b(\bar{\boldsymbol{e}}), y)/(CE(\boldsymbol{\psi}^i(\bar{\boldsymbol{e}}), y) + CE(\boldsymbol{\psi}^b(\bar{\boldsymbol{e}}), y))$, the objective function $\mathcal{L}_{\text{rel}}$ is defined using the above relative difficulty score of each data sample: $\mathcal{L}_{\text{rel}} = \text{Score}(\bar{\boldsymbol{e}}, y) \cdot CE(\boldsymbol{\psi}^i(\bar{\boldsymbol{e}}), y) + \lambda_{\text{rel}} GCE(\boldsymbol{\psi}^b(\bar{\boldsymbol{e}}), y)$ where λ_{rel} is the weight that adjusts between two loss terms.

In the attribute-composition phase, we swap the disentangled latent vectors among the training sets [38]. We randomly permute the intrinsic and bias features in each mini-batch, creating $\mathbf{E}_{\text{swap}} = [\boldsymbol{\phi}^i(\mathbf{x}); \boldsymbol{\phi}^b_{\text{swap}}(\mathbf{x})]$ where $\boldsymbol{\phi}^b_{\text{swap}}(\mathbf{x})$ denotes the randomly permuted bias attributes. This process produces augmented bias-conflicting latent vectors. As similar as the definition of $\mathbf{e}$, we define $\mathbf{e}_{\text{swap}} \in \{\mathbf{E}^i_{\text{swap}}, \mathbf{E}^b_{\text{swap}}\}$ and generate $\bar{\mathbf{e}}_{\text{swap}}$ following the same process described in eqs 1, 2 and our proposed broadcast scheme. The objective function for this phase is: $\mathcal{L}_{\text{swap}} = \text{Score}(\bar{\boldsymbol{e}}, y) \cdot CE(\boldsymbol{\psi}^i(\bar{\boldsymbol{e}}_{\text{swap}}), y) + \lambda_{\text{swap}} GCE(\boldsymbol{\psi}^b(\bar{\boldsymbol{e}}_{\text{swap}}), \tilde{y})$ where $\tilde{y}$ denotes target labels for permute bias attributes $\boldsymbol{\phi}^b_{\text{swap}}(\mathbf{x})$, and λ_{swap} is the balancing weight between two loss terms.

Therefore, the total loss function is a combination of the above components: $\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{rel}} + \lambda_{\text{swap}} \cdot \mathcal{L}_{\text{swap}}$. Here, λ_{swap} is the weight that adjusts the importance of the feature augmentation.

3 Experiments

This section presents our experimental results, focusing on performance evaluation on various biased datasets and interpretable analysis for attribute-centric representation learning. The additional experimental results, including full performance evaluation and analysis, are provided in Appendix D.

Datasets. Following [38], we used three well-known benchmark datasets for debiasing methods to evaluate DGW's performance and interpretability: Colored MNIST (C-MNIST), Corrupted CIFAR10 (C-CIFAR-10), and Biased FFHQ (BFFHQ). C-MNIST and C-CIFAR-10 are synthetic datasets designed to test model generalization on unbiased test sets by varying the ratio of bias-conflicting samples (0.5%, 1%, 2%, and 5%). BFFHQ is a real-world dataset from FFHQ [31], containing face images annotated with age (intrinsic attribute) and gender (bias attribute).

Table 1: Test accuracy (%) on unbiased test sets of C-MNIST and C-CIFAR-10, and the bias-conflicting test set of BFFHQ with varying ratio of bias-conflicting samples. (†) methods relying on the easy-to-learn heuristic, and (‡) methods combined with GWT. V+CCT indicates the direct integration of Vanilla and CCT. The best-performing results are shown in bold, and the second-best results are underlined.

Dataset	Ratio (%)	Vanilla	LFA†	V+CCT‡	DGW‡
C-MNIST	0.5	$36.2_{\pm 1.8}$	$\underline{67.4}_{\pm 1.7}$	$26.3_{\pm 1.1}$	$\mathbf{70.3}_{\pm 1.2}$
	1.0	$50.8_{\pm 2.3}$	$\mathbf{79.0}_{\pm 1.0}$	$40.1_{\pm 2.1}$	$\underline{77.4}_{\pm 0.4}$
	2.0	$65.2_{\pm 2.1}$	$\underline{85.0}_{\pm 0.8}$	$56.2_{\pm 1.8}$	$\mathbf{85.3}_{\pm 0.7}$
	5.0	$81.6_{\pm 0.6}$	$\underline{88.7}_{\pm 1.3}$	$73.4_{\pm 0.8}$	$\mathbf{89.1}_{\pm 0.6}$
C-CIFAR-10	0.5	$22.8_{\pm 0.3}$	$\underline{27.9}_{\pm 1.0}$	$15.2_{\pm 0.3}$	$\mathbf{30.4}_{\pm 2.2}$
	1.0	$26.2_{\pm 0.5}$	$\mathbf{34.3}_{\pm 0.6}$	$20.6_{\pm 0.4}$	$\underline{33.6}_{\pm 2.4}$
	2.0	$31.1_{\pm 0.6}$	$\underline{40.3}_{\pm 2.4}$	$24.6_{\pm 0.5}$	$\mathbf{42.0}_{\pm 1.9}$
	5.0	$42.0_{\pm 0.3}$	$\underline{50.3}_{\pm 1.1}$	$35.6_{\pm 0.8}$	$\underline{50.3}_{\pm 1.9}$
BFFHQ	0.5	$54.5_{\pm 0.6}$	$\underline{59.5}_{\pm 3.8}$	$52.6_{\pm 1.1}$	$\mathbf{65.6}_{\pm 3.3}$

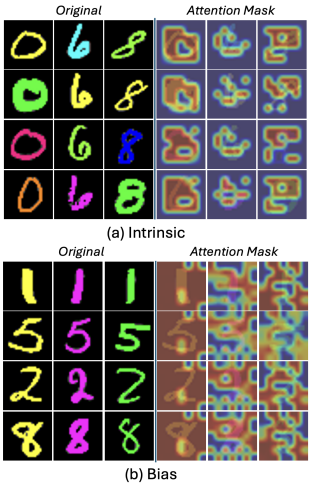


Figure 2: Visualization of $\mathbf{A}^i$ and $\mathbf{A}^b$ for the C-MNIST dataset

Performance Evaluation. Our set of debiasing baselines includes two different approaches: Vanilla network, and LFA [38]. Vanilla refers to the classification model trained only with the original cross-entropy (CE) loss without debiasing strategies. LFA and our method do not require prior knowledge about the bias type. Furthermore, we configure a naive debiasing approach integrated with GWT implementation: V+CCT. CCT [27] proposed an instantiation of GWT applicable to implement an interpretable model. To compare our DGW, we simply configure the direction fusion of the Vanilla network with CCT as a GWT debiasing method. Table 1 shows that DGW performs equivalently or, in some cases, better than LFA, demonstrating its robustness and flexibility in debiasing image classification tasks. Furthermore, the poor performance of V+CCT highlights the importance of finding the proper configuration for debiasing methods, indicating the effectiveness of our DGW configuration as a debiasing method. Full performance evaluation with more baselines is in Appendix D.4.

Analysis for Interpretable Attribute Representation. DGW generates two attention masks: $\mathbf{A}^i = \mathbf{A}(\mathbf{S}^i_{\text{latent}}, \mathbf{E}^i)$ for intrinsic attributes, focusing on essential features like shape, and $\mathbf{A}^b = \mathbf{A}(\mathbf{S}^b_{\text{latent}}, \mathbf{E}^b)$ for biased attributes, capturing non-essential features like color. We visualize both attention masks to demonstrate the interpretable representation of learning in our method.

For the C-MNIST dataset, intrinsic attention masks highlight the shapes of the digits, ignoring colors. For instance, the digits '0', '6', and '8' consistently highlight shape regions (Fig. 2(a)), showing that the model focuses on shape for classification. Conversely, bias attention masks highlight color regions, not shapes. Digits '1', '5', '2', and '8' in yellow/magenta/green show nearly identical masks (Fig. 2(b)), indicating a focus on color. This confirms that the biased components of DGW capture color information, which is irrelevant for digit recognition. More visualization results can be found in Appendix D.5.

4 Conclusion

In this work, we introduced Debiasing Global Workspace (DGW), a framework designed to learn debiased representations of attributes in neural networks. By leveraging attention mechanisms inspired by the Global Workspace Theory, our method effectively differentiates between intrinsic and biased attributes, enhancing both performance and interpretability. Comprehensive evaluations across various biased datasets demonstrated that DGW improves model robustness and generalizability on biased data and provides interpretable insights into the model's decision-making process.

Acknowledgments and Disclosure of Funding

This work was supported by NSF EFMA-2223839.

References

[1] Agarwal, V., Shetty, R., Fritz, M.: Towards causal vqa: Revealing and reducing spurious correlations by invariant and covariant semantic editing. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 9690–9698 (2020)

[2] Baars, B.J.: A cognitive theory of consciousness. Cambridge University Press (1993)

[3] Baars, B.J.: Global workspace theory of consciousness: toward a cognitive neuroscience of human experience. Progress in brain research **150**, 45–53 (2005)

[4] Bahng, H., Chun, S., Yun, S., Choo, J., Oh, S.J.: Learning de-biased representations with biased representations. In: Proceedings of the International Conference on Machine Learning, pp. 528–539, PMLR (2020)

[5] Bengio, Y.: The consciousness prior. CoRR **abs/1709.08568** (2017), URL `http://arxiv.org/abs/1709.08568`

[6] Burgess, C.P., Matthey, L., Watters, N., Kabra, R., Higgins, I., Botvinick, M., Lerchner, A.: Monet: Unsupervised scene decomposition and representation. arXiv preprint arXiv:1901.11390 (2019)

[7] Chang, M., Griffiths, T., Levine, S.: Object representations as fixed points: Training iterative refinement algorithms with implicit differentiation. Advances in Neural Information Processing Systems **35**, 32694–32708 (2022)

[8] Cho, K., van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., Bengio, Y.: Learning phrase representations using RNN encoder–decoder for statistical machine translation. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 1724–1734 (2014)

[9] Cortes, C., Mohri, M., Rostamizadeh, A.: Algorithms for learning kernels based on centered alignment. The Journal of Machine Learning Research **13**(1), 795–828 (2012)

[10] Darlow, L., Jastrzębski, S., Storkey, A.: Latent adversarial debiasing: Mitigating collider bias in deep neural networks. arXiv preprint arXiv:2011.11486 (2020)

[11] Dehaene, S., Changeux, J.P.: Experimental and theoretical approaches to conscious processing. Neuron **70**(2), 200–227 (2011)

[12] Didolkar, A.R., Goyal, A., Bengio, Y.: Cycle consistency driven object discovery. In: The Twelfth International Conference on Learning Representations (2023)

[13] Fodor, J.A., Pylyshyn, Z.W.: Connectionism and cognitive architecture: A critical analysis. Cognition **28**(1-2), 3–71 (1988)

[14] Geirhos, R., Jacobsen, J.H., Michaelis, C., Zemel, R., Brendel, W., Bethge, M., Wichmann, F.A.: Shortcut learning in deep neural networks. Nature Machine Intelligence **2**(11), 665–673 (2020)

[15] Geirhos, R., Rubisch, P., Michaelis, C., Bethge, M., Wichmann, F.A., Brendel, W.: Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness. In: Proceedings of the International Conference on Learning Representations (2018)

[16] Goel, K., Gu, A., Li, Y., Re, C.: Model patching: Closing the subgroup performance gap with data augmentation. In: Proceedings of the International Conference on Learning Representations (2020)

[17] Goyal, A., Bengio, Y.: Inductive biases for deep learning of higher-level cognition. Proceedings of the Royal Society A **478**(2266), 20210068 (2022)

[18] Goyal, A., Didolkar, A.R., Lamb, A., Badola, K., Ke, N.R., Rahaman, N., Binas, J., Blundell, C., Mozer, M.C., Bengio, Y.: Coordination among neural modules through a shared global workspace. In: Proceedings of the International Conference on Learning Representations (2021)

[19] Greff, K., Kaufman, R.L., Kabra, R., Watters, N., Burgess, C., Zoran, D., Matthey, L., Botvinick, M., Lerchner, A.: Multi-object representation learning with iterative variational inference. In: International conference on machine learning, pp. 2424–2433, PMLR (2019)

[20] Greff, K., Van Steenkiste, S., Schmidhuber, J.: On the binding problem in artificial neural networks. arXiv preprint arXiv:2012.05208 (2020)

[21] Guo, C., Pleiss, G., Sun, Y., Weinberger, K.Q.: On calibration of modern neural networks. In: Proceedings of the International Conference on Machine Learning (ICML), pp. 1321–1330 (2017)

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

[23] He, T., Zhang, Z., Zhang, H., Zhang, Z., Xie, J., Li, M.: Bag of tricks for image classification with convolutional neural networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 558–567 (2019)

[24] Hendrycks, D., Basart, S., Mu, N., Kadavath, S., Wang, F., Dorundo, E., Desai, R., Zhu, T., Parajuli, S., Guo, M., Song, D., Steinhardt, J., Gilmer, J.: The many faces of robustness: A critical analysis of out-of-distribution generalization (2021)

[25] Hendrycks, D., Dietterich, T.: Benchmarking neural network robustness to common corruptions and perturbations. In: International Conference on Learning Representations (2018)

[26] Hendrycks, D., Zhao, K., Basart, S., Steinhardt, J., Song, D.: Natural adversarial examples. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp. 15262–15271 (2021)

[27] Hong, J., Park, K.H., Pavlic, T.P.: Concept-centric transformers: Enhancing model interpretability through object-centric concept learning within a shared global workspace. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV), pp. 4880–4891 (January 2024)

[28] Hong, Y., Yang, E.: Unbiased classification through bias-contrastive and bias-balanced learning. Advances in Neural Information Processing Systems **34**, 26449–26461 (2021)

[29] Hwang, I., Lee, S., Kwak, Y., Oh, S.J., Teney, D., Kim, J.H., Zhang, B.T.: Selecmix: Debiased learning by contradicting-pair sampling. Advances in Neural Information Processing Systems **35**, 14345–14357 (2022)

[30] Jia, B., Liu, Y., Huang, S.: Improving object-centric learning with query optimization. In: Proceedings of the International Conference on Learning Representations (2022)

[31] Karras, T., Laine, S., Aila, T.: A style-based generator architecture for generative adversarial networks. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp. 4401–4410 (2019)

[32] Kataoka, Y., Matsubara, T., Uehara, K.: Image generation using generative adversarial networks and attention mechanism. In: Proceedings of the IEEE/ACIS International Conference on Computer and Information Science (ICIS), pp. 1–6, IEEE (2016)

[33] Kim, B., Kim, H., Kim, K., Kim, S., Kim, J.: Learning not to learn: Training deep neural networks with biased data. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp. 9012–9020 (2019)

[34] Kim, E., Lee, J., Choo, J.: Biaswap: Removing dataset bias with bias-tailored swapping augmentation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 14992–15001 (2021)

[35] Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)

[36] Kornblith, S., Norouzi, M., Lee, H., Hinton, G.: Similarity of neural network representations revisited. In: Proceedings of the International Conference on Machine Learning, pp. 3519–3529, PMLR (2019)

[37] Krizhevsky, A., Hinton, G.: Learning multiple layers of features from tiny images. Tech. Rep. TR-2009, University of Toronto, Toronto, Ontario (2009)

[38] Lee, J., Kim, E., Lee, J., Lee, J., Choo, J.: Learning debiased representation via disentangled feature augmentation. Advances in Neural Information Processing Systems **34**, 25123–25133 (2021)

[39] Li, Y., Vasconcelos, N.: Repair: Removing representation bias by dataset resampling. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp. 9572–9581 (2019)

[40] Li, Y., Yu, Q., Tan, M., Mei, J., Tang, P., Shen, W., Yuille, A., Xie, C.: Shape-texture debiased neural network training. arXiv preprint arXiv:2010.05981 (2020)

[41] Lim, J., Kim, Y., Kim, B., Ahn, C., Shin, J., Yang, E., Han, S.: Biasadv: Bias-adversarial augmentation for model debiasing. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 3832–3841 (2023)

[42] Liu, E.Z., Haghgoo, B., Chen, A.S., Raghunathan, A., Koh, P.W., Sagawa, S., Liang, P., Finn, C.: Just train twice: Improving group robustness without training group information. In: International Conference on Machine Learning, pp. 6781–6792, PMLR (2021)

[43] Locatello, F., Weissenborn, D., Unterthiner, T., Mahendran, A., Heigold, G., Uszkoreit, J., Dosovitskiy, A., Kipf, T.: Object-centric learning with slot attention. Advances in Neural Information Processing Systems **33**, 11525–11538 (2020)

[44] Luo, P., Wang, G., Lin, L., Wang, X.: Deep dual learning for semantic image segmentation. In: Proceedings of the IEEE international conference on computer vision, pp. 2718–2726 (2017)

[45] van der Maaten, L., Hinton, G.: Visualizing data using t-sne. Journal of Machine Learning Research **9**(86), 2579–2605 (2008)

[46] Mashour, G.A., Roelfsema, P., Changeux, J.P., Dehaene, S.: Conscious processing and the global neuronal workspace hypothesis. Neuron **105**(5), 776–798 (2020)

[47] Minderer, M., Bachem, O., Houlsby, N., Tschannen, M.: Automatic shortcut removal for self-supervised representation learning. In: Proceedings of the International Conference on Machine Learning, pp. 6927–6937, PMLR (2020)

[48] Minsky, M.: Society of mind. Simon and Schuster (1988)

[49] Nam, J., Cha, H., Ahn, S., Lee, J., Shin, J.: Learning from failure: De-biasing classifier from biased classifier. Advances in Neural Information Processing Systems **33**, 20673–20684 (2020)

[50] Posner, M.I.: Attention: the mechanisms of consciousness. Proceedings of the National Academy of Sciences **91**(16), 7398–7403 (1994)

[51] Raghu, M., Unterthiner, T., Kornblith, S., Zhang, C., Dosovitskiy, A.: Do vision transformers see like convolutional neural networks? Advances in Neural Information Processing Systems **34**, 12116–12128 (2021)

[52] Robbins, P.: Modularity of mind. In: Zalta, E.N. (ed.) The Stanford Encyclopedia of Philosophy, Metaphysics Research Lab, Stanford University, Winter 2017 edn. (2017), URL `https://plato.stanford.edu/archives/win2017/entries/modularity-mind/`

[53] Rosenberg, A., Hirschberg, J.: V-measure: A conditional entropy-based external cluster evaluation measure. In: Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, pp. 410–420 (2007)

[54] Sagawa, S., Koh, P.W., Hashimoto, T.B., Liang, P.: Distributionally robust neural networks. In: International Conference on Learning Representations (2019)

[55] Sanh, V., Wolf, T., Belinkov, Y., Rush, A.M.: Learning from others' mistakes: Avoiding dataset biases without modeling them. In: International Conference on Learning Representations (2020)

[56] Seth, A.K., Bayne, T.: Theories of consciousness. Nature Reviews Neuroscience **23**(7), 439–452 (2022)

[57] Tartaglione, E., Barbano, C.A., Grangetto, M.: End: Entangling and disentangling deep representations for bias correction. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp. 13508–13517 (2021)

[58] Torralba, A., Efros, A.A.: Unbiased look at dataset bias. In: CVPR 2011, pp. 1521–1528, IEEE (2011)

[59] Tu, B., Zhou, C., Kuang, W., Chen, S., Plaza, A.: Multiattribute sample learning for hyperspectral image classification using hierarchical peak attribute propagation. IEEE Transactions on Instrumentation and Measurement **71**, 1–17 (2022)

[60] VanRullen, R., Kanai, R.: Deep learning and the global workspace theory. Trends in Neurosciences **44**(9), 692–704 (2021)

[61] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. Advances in Neural Information Processing Systems **30** (2017)

[62] Verma, V., Lamb, A., Beckham, C., Najafi, A., Mitliagkas, I., Lopez-Paz, D., Bengio, Y.: Manifold mixup: Better representations by interpolating hidden states. In: International conference on machine learning, pp. 6438–6447, PMLR (2019)

[63] Wang, H., He, Z., Lipton, Z.C., Xing, E.P.: Learning robust representations by projecting superficial statistics out. In: Proceedings of the International Conference on Learning Representations (2018)

[64] Wang, X., Gupta, A.: Generative image modeling using style and structure adversarial networks. In: Proceedings of the European Conference on Computer Vision, pp. 318–335, Springer (2016)

[65] Wang, Z., Qinami, K., Karakozis, I.C., Genova, K., Nair, P., Hata, K., Russakovsky, O.: Towards fairness in visual recognition: Effective strategies for bias mitigation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp. 8919–8928 (2020)

[66] Xie, Q., Luong, M.T., Hovy, E., Le, Q.V.: Self-training with noisy student improves imagenet classification. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 10687–10698 (2020)

[67] Zhang, Y.K., Wang, Q.W., Zhan, D.C., Ye, H.J.: Learning debiased representations via conditional attribute interpolation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 7599–7608 (2023)

[68] Zhang, Z., Sabuncu, M.: Generalized cross entropy loss for training deep neural networks with noisy labels. Advances in neural information processing systems **31** (2018)

[69] Zhao, J., Peng, Y., He, X.: Attribute hierarchy based multi-task learning for fine-grained image classification. Neurocomputing **395**, 150–159 (2020)

[70] Zheng, S., Cheng, M.M., Warrell, J., Sturgess, P., Vineet, V., Rother, C., Torr, P.H.: Dense semantic image segmentation with objects and attributes. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 3214–3221 (2014)

Appendix

A Reproducibility

All source codes, figures, models, etc., are available at `https://github.com/jyhong0304/debiasing_global_workspace`.

B Related Work

Debiasing Methods. One of the most well-known existing debiasing methods is those with predefined forms of bias or specific bias labels. This method involves identifying specific biases before training [28, 33, 39, 54]. The model then learns to ignore or correct these biases. While effective, it depends on accurately identifying biases beforehand, which can be challenging. Another approach [4, 57] uses bias labels to tag data, allowing the model to differentiate between biased and unbiased data during training. This improves learning but requires extensive manual labeling.

Debiasing approaches using the easy-to-learn heuristic are other effective methods. Biases are easier for models to learn [49] than intrinsic features. Techniques like dynamic training schemes, re-weighting samples, and data augmentation [15, 38, 47, 39, 41] help models focus on unbiased features. However, these methods struggle with insufficient diverse samples. Complex models can learn invariant features or correct representations but are difficult to design and train [59, 69, 1, 4, 15, 16, 33, 40, 47, 57, 65].

Additionally, SelecMix [29] creates new training samples by mixing pairs with similar labels but different biases, or different labels but similar biases, using an auxiliary contrastive model. While effective, this adds significant training complexity. χ^2 model [67] learns debiased representations by identifying Intermediate Attribute Samples (IAS) and using a χ-structured metric learning objective. However, its reliance on training dynamics to identify IASs makes it different from our approach and out of the scope of our study.

Deep Learning and Global Workspace Theory. In neuroscience and cognitive science, there is an ongoing effort to develop theories of consciousness (ToCs) to identify the neural correlates of consciousness, as reviewed by Seth and Bayne [56]. One such theory is the Global Workspace Theory (GWT) [2, 11, 46], which is inspired by the 'blackboard' architecture used in artificial intelligence. In this architecture, a centralized resource, the blackboard, facilitates information sharing among specialized processors.

Recent studies have aimed to bridge the gap between neuroscience and deep learning, focusing on practical solutions for implementing a GWT using current deep learning components while considering the equivalent brain mechanisms [17, 48, 52, 18, 27]. Bengio [5] emphasized learning high-level concepts by selecting key elements through attention, forming a low-dimensional conscious state similar to language, which aids in better representation learning. Mashour et al. [46] details GWT's implementation in neuroscience, suggesting that consciousness arises from extensive information sharing across brain regions via a central network of neurons.

Inspired by GWT, our Debiasing Global Workspace (DGW) framework manages intrinsic and biased attributes in neural networks. DGW integrates information from intrinsic and bias specialists, ensuring disentangled representations are considered in decision making. Unlike prior works focusing on monolithic architecture or general-purpose learning, our approach uniquely applies these theories to debiasing neural networks.

Object-Centric Representation Learning. Humans outperform sophisticated AI technologies due to our exceptional ability to recombine previously acquired knowledge, allowing us to extrapolate to novel scenarios [13, 17, 20]. Pursuing representations that generalize compositionally has been a significant research topic, with object-centric representation learning [6, 19, 43, 7, 30] emerging as a prominent effort. This approach represents each object in an image with a unique subset of the image's latent code, enabling compositional generalization due to its modular structure.

Due to its simple yet effective design, Slot-Attention (SA) [43] has gained significant attention in unsupervised object-centric representation learning. Its iterative attention mechanism allows SA to

learn and compete between slots for explaining parts of the input, showing a soft clustering effect on visual inputs [43]. Some recent works on implementing a cognitive architecture using object-centric methods have been proposed [27, 12]. Our approach also emphasizes compositional generalization in debiasing learning, using the slot-based method to implement a crucial module. The benefits of this method are noteworthy and deserve further exploration.

C Further Details of Our Method

C.1 The Conceptual Instantiation of Debiasing Global Workspace

Figure 1 in the main text depicts a conceptual overview of our proposed DGW framework. The conceptual flow of the DGW proceeds through a sequence of steps that we describe in detail here.

Step 0. To learn disentangled representations of intrinsic and biased attributes, we introduce two specialists: intrinsic ϕ^i and biased ϕ^b. In the original GWT, the specialists connect to the global workspace before any stimulus appears, coupling their latent spaces bidirectionally with the workspace. We modify this setup to control the connections to backpropagate different information to two specialists separately (black and red connections between specialists and DGW in Step 0 in Fig. 1). Specifically, the intrinsic and bias specialists function identically in the forward pass. However, during the backpropagation stage, only the intrinsic attribute encoder updates its parameters and learns, while the bias attribute encoder remains frozen and does not undergo parameter updates when the task of the model is to find intrinsic attributes.

Step 1. The DGW acts as an independent and intermediate shared latent space trained to perform unsupervised neural translation between the C latent spaces from the specialized modules. The translation system is optimized to ensure that successive translation and back translation (e.g., a cycle from A to B, then back to A) return the original input [18, 60]. We implement specific operations to mimic the translation system by leveraging the residual operations [22] and a variant of mixup [62].

In [50], attention determines what information is consciously perceived and what is discarded in brains. In GWT, attention selects the information that enters the workspace. When a specific module is connected to the workspace through attention, its latent space activation vector is copied into the DGW. This internal copy serves as a bidirectional connection interface between the corresponding module and the DGW.

When a new stimulus, such as the digit 'zero,' appears, its latent activity transfers to the corresponding internal copy inside the workspace, initiating a broadcast to all other domains. This shared latent space ($\mathbf{S}^i_{\text{latent}}$ in Fig. 1) uses translations and back translations from all modules to compute and train via error backpropagation. We introduce a recurrent top-down pathway, and it can sometimes be considered as a key to account for the global ignition property observed in the brain when an input reaches consciousness, and the corresponding module is mobilized into the conscious global workspace [60].

Step 2. The incoming information is then immediately broadcast and translated (via the shared latent space) into the latent space of all other modules. In GWT, this translation process is automatic. However, we modify this to manually enforce learning of intrinsic and biased attribute representation with different loss functions. Specifically, we enforce the classifier ϕ^i to learn intrinsic attributes through error backpropagation from specific training objectives (Step 2-1 in Fig. 1). Step 2-2 simultaneously enforces the connection to the classifier ψ^b and limits backpropagation to the intrinsic specialist ϕ^i to learn the bias attribute representations.

C.2 Other Approach to Broadcast Information

While we introduced a residual connection in the main text as the information broadcast, the other way of operation is a modified version of Manifold Mixup [62], which interpolates feature embeddings to capture higher-level information.

$$\bar{\mathbf{e}} = \text{Mix}_\alpha \left(\mathbf{e}, \left(\mathbf{A} \left(\mathbf{s}^{(n+1)}_{\text{latent}}, \mathbf{e} \right) \cdot v \left(\mathbf{s}^{(n+1)}_{\text{latent}} \right) \right) \right),$$

where $\text{Mix}_\alpha(a, b) = \alpha \cdot a + (1 - \alpha) \cdot b$, and $\alpha \sim \text{Beta}(\beta, \beta)$. The updated feature vector $\bar{\mathbf{e}}$ is then fed to the classifier ψ^i and ψ^b. We compare the performance of using residual connections versus our modified Manifold Mixup in Appendix D.4.

C.3 Entropy regularization.

We empirically incorporate an additional regularization term on the latent slot attention mask to enhance performance:

$$\mathcal{L}_{\text{ent}} = H(\mathbf{A}(\mathbf{s}_{\text{latent}}^{(n)}, \mathbf{e})) + H(\mathbf{A}(\mathbf{s}_{\text{latent}}^{(n)}, \mathbf{e}_{\text{swap}})),$$

where $\mathbf{A}(\mathbf{s}_{\text{latent}}^{(n)}, \mathbf{e})$ and $\mathbf{A}(\mathbf{s}_{\text{latent}}^{(n)}, \mathbf{e}_{\text{swap}})$ are attention masks from the last iteration of eq. 2 in the main text. Minimizing entropy $H(\mathbf{A}) = H(a_1, \ldots, a_{|\mathbf{A}|}) = (1/|\mathbf{A}|) \sum_i -a_i \cdot \log(a_i)$ encourages the attention masks to be consistent over the input features captured by the latent slots. This regularization ensures the model's attention remains focused and interpretable across different input scenarios.

Therefore, the total loss function is a combination of the above components: $\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{rel}} + \lambda_{\text{swap}} \cdot \mathcal{L}_{\text{swap}} + \lambda_{\text{ent}} \cdot \mathcal{L}_{\text{ent}}$. Here, λ_{swap} and λ_{ent} are weights that adjust the importance of the feature augmentation and entropy regularization, respectively. This comprehensive loss function ensures balanced training that enhances the model's ability to learn and generalize effectively while maintaining interpretability and robustness.

D Further Experimental Results and Details

In this section, we explain further experimental results and details. All experiments are conducted with three different random seeds and 95% confidence intervals.

D.1 Hardware Specification of The Server

The hardware specification of the server that we used to experiment is as follows:

- CPU: Intel® Core™ i7-6950X CPU @ 3.00GHz (up to 3.50 GHz)
- RAM: 128 GB (DDR4 2400MHz)
- GPU: NVIDIA GeForce Titan Xp GP102 (Pascal architecture, 3840 CUDA Cores @ 1.6 GHz, 384-bit bus width, 12 GB GDDR G5X memory)

D.2 Datasets

We describe the details of biased datasets, Colored MNIST (C-MNIST), Corrupted CIFAR-10 (C-CIFAR-10), and BFFHQ.

Colored MNIST. Following existing studies [49, 33, 39, 4, 10, 38], this biased dataset comprises two highly correlated attributes: color and digit. We added specific colors to the foreground of each digit, generating bias-aligned and bias-conflicting samples for different ratios of bias-conflicting samples:

- 0.5%: (54751:249)
- 1%: (54509:491)
- 2%: (54014:986)
- 5%: (52551:2449)

Corrupted CIFAR-10. Among 15 different corruptions introduced in the original dataset [25], we selected types including Brightness, Contrast, Gaussian Noise, Frost, Elastic Transform, Gaussian Blur, Defocus Blur, Impulse Noise, Saturate, and Pixelate, related to CIFAR-10 classes [37]. We used the most severe level of corruption for the dataset, with the following bias-aligned and bias-conflicting samples:

- 0.5%: (44832:228)

Table A-1: Test accuracy (%) on unbiased test sets of C-MNIST and C-CIFAR-10, and the bias-conflicting test set of BFFHQ with varying ratio of bias-conflicting samples. ($*$) denotes methods tailored to predefined forms of bias, ($^\circ$) methods using bias labels, ($\dagger$) methods relying on the easy-to-learn heuristic, and ($\ddagger$) methods combined with GWT. V+CCT indicates the direct integration of Vanilla and CCT. DGW+M refers to DGW with our mixup strategy, and DGW+R refers to DGW with residual connection. Performance for HEX and EnD is from [38], while results for Vanilla, ReBias, LfF, LFA, V+CCT and DGW are from our evaluation. The best-performing results are shown in bold, and the second-best results are underlined.

Dataset	Ratio (%)	Vanilla	HEX*	EnD$^\circ$	ReBias*	LfF†	LFA†	V+CCT‡	DGW+M‡	DGW+R‡
C-MNIST	0.5	$36.2_{\pm 1.8}$	$30.3_{\pm 0.8}$	$34.3_{\pm 1.2}$	**72.2**$_{\pm 1.5}$	$47.5_{\pm 3.0}$	$67.4_{\pm 1.7}$	$26.3_{\pm 1.1}$	$68.9_{\pm 2.8}$	$\underline{70.3}_{\pm 1.2}$
	1.0	$50.8_{\pm 2.3}$	$43.7_{\pm 5.5}$	$49.5_{\pm 2.5}$	**86.6**$_{\pm 0.6}$	$64.6_{\pm 2.5}$	$79.0_{\pm 1.0}$	$40.1_{\pm 2.1}$	$\underline{81.3}_{\pm 1.2}$	$77.4_{\pm 0.4}$
	2.0	$65.2_{\pm 2.1}$	$56.9_{\pm 2.6}$	$68.5_{\pm 2.2}$	**92.7**$_{\pm 0.3}$	$74.9_{\pm 3.7}$	$85.0_{\pm 0.8}$	$56.2_{\pm 1.8}$	$84.6_{\pm 1.5}$	$\underline{85.3}_{\pm 0.7}$
	5.0	$81.6_{\pm 0.6}$	$74.6_{\pm 3.2}$	$81.2_{\pm 1.4}$	**97.1**$_{\pm 0.6}$	$80.2_{\pm 0.9}$	$88.7_{\pm 1.3}$	$73.4_{\pm 0.8}$	$88.9_{\pm 0.2}$	$\underline{89.1}_{\pm 0.6}$
C-CIFAR-10	0.5	$22.8_{\pm 0.3}$	$13.9_{\pm 0.1}$	$22.9_{\pm 0.3}$	$20.8_{\pm 0.2}$	$25.0_{\pm 1.5}$	$27.9_{\pm 1.0}$	$15.2_{\pm 0.3}$	$\underline{29.6}_{\pm 0.5}$	**30.4**$_{\pm 2.2}$
	1.0	$26.2_{\pm 0.5}$	$14.8_{\pm 0.4}$	$25.5_{\pm 0.4}$	$24.4_{\pm 0.4}$	$31.0_{\pm 0.4}$	$34.3_{\pm 0.6}$	$20.6_{\pm 0.4}$	**34.9**$_{\pm 0.4}$	$\underline{33.6}_{\pm 2.4}$
	2.0	$31.1_{\pm 0.6}$	$15.2_{\pm 0.5}$	$31.3_{\pm 0.4}$	$29.6_{\pm 2.9}$	$38.3_{\pm 0.4}$	$40.3_{\pm 2.4}$	$24.6_{\pm 0.5}$	$\underline{41.3}_{\pm 1.0}$	**42.0**$_{\pm 1.9}$
	5.0	$42.0_{\pm 0.3}$	$16.0_{\pm 0.6}$	$40.3_{\pm 0.9}$	$41.1_{\pm 0.2}$	$48.8_{\pm 0.9}$	$\underline{50.3}_{\pm 1.1}$	$35.6_{\pm 0.8}$	**52.3**$_{\pm 0.8}$	$\underline{50.3}_{\pm 1.9}$
BFFHQ	0.5	$54.5_{\pm 0.6}$	$52.8_{\pm 0.9}$	$56.9_{\pm 1.4}$	$58.0_{\pm 0.2}$	$63.6_{\pm 2.9}$	$59.5_{\pm 3.8}$	$52.6_{\pm 1.1}$	**66.9**$_{\pm 1.0}$	$\underline{65.6}_{\pm 3.3}$

- 1%: (44527:442)
- 2%: (44145:887)
- 5%: (42820:2242)

BFFHQ. The dataset is created by using the Flickr-Faces-HQ (FFHQ) Dataset [31], focusing on age and gender as two strongly correlated attributes. The dataset includes 19200 training images (19104 bias-aligned and 96 bias-conflicting) and 1000 testing samples.

D.3 Image Preprocessing

Following Lee et al. [38], our model is trained and evaluated using fixed-size images. For C-MNIST, the size is 28×28; for C-CIFAR-10, it is 32×32, and for BFFHQ, it is 224×224. Images for C-CIFAR-10 and BFFHQ are preprocessed using random crop and horizontal flip transformations, as well as normalization along each channel (3, H, W) with a mean of (0.4914, 0.4822, 0.4465) and standard deviation of (0.2023, 0.1994, 0.2010). We do not use augmentation techniques for C-MNIST.

D.4 Performance Evaluation

Full Performance Comparison. Our set of debiasing baselines includes six different approaches: Vanilla network, HEX [63], EnD [57], ReBias [4], LfF [49], and LFA [38]. Vanilla refers to the classification model trained only with the original cross-entropy (CE) loss without debiasing strategies. EnD leverages the explicit bias labels, such as the color labels in the C-MNIST dataset, during the training phase. HEX and ReBias assume an image's texture as a bias type, whereas LfF, LFA, and our method do not require any prior knowledge about the bias type. Furthermore, we configure a naive debiasing approach integrated with GWT implementation: V+CCT. CCT [27] proposed an instantiation of GWT applicable to implement an interpretable model. To compare our DGW, we simply configure the direction fusion of the Vanilla network with CCT as a GWT debiasing method. Table. A-1 shows the full table of test performance evaluation.

Implementation Details. We followed the implementation details from [38]. We used a fully connected network for attribute encoders with three hidden layers for C-MNIST and ResNet-18 for C-CIFAR-10 and BFFHQ. We employed a fully connected classifier with double the hidden units to handle the combined output from the intrinsic attribute encoder $\boldsymbol{\phi}^i$ and the bias attribute encoder $\boldsymbol{\phi}^b$.

During testing, only the intrinsic classifier $\boldsymbol{\psi}^i(\mathbf{e})$ was used for final predictions. We used batch sizes of 256 for C-MNIST and C-CIFAR-10, and 64 for BFFHQ, respectively. 2 concepts and size of 8 were used for C-MNIST, 5 and 16 for C-CIFAR-10, and 10 and 32 for BFFHQ, respectively. We trained our model and baselines with three trials and reported the averaged accuracy and standard deviation.

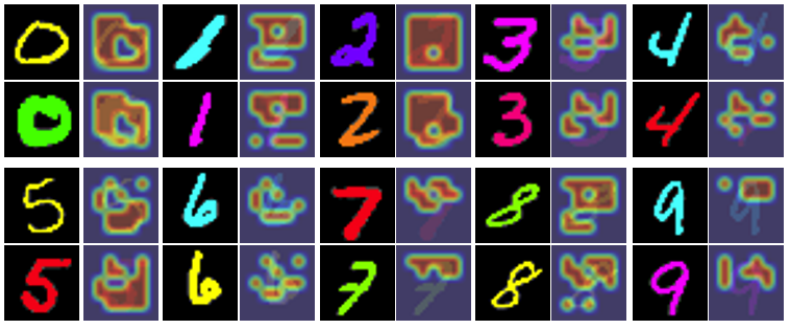


Figure A-1: Visualization of attention masks $\mathbf{A}^i$ for the C-MNIST dataset

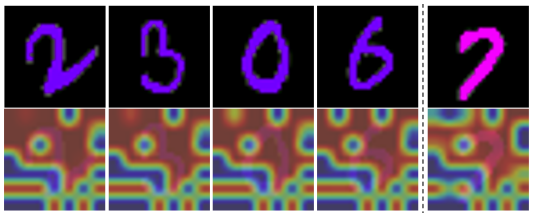


Figure A-2: Visualization from the C-MNIST dataset showing attention masks $\mathbf{A}^b$, highlighting color patterns. Digits in similar colors (e.g., 2, 3, 0, and 6) share similar attention mask patterns.

Training Details. For training, we use the Adam [35] optimizer with default parameters (i.e., betas = (0.9, 0.999) and weight decay = 0.0) provided in the PyTorch™framework. We define two different learning rates: $\texttt{LR}_{\texttt{DGW}}$ for our `DGW` modules, and `LR` for the remaining modules in our method, including encoders and classifiers. For C-MNIST, `LR` is 0.01, while $\texttt{LR}_{\texttt{DGW}}$ is 0.0005 for C-MNIST-2%, 0.002 is for the remaining ratios of datasets. For C-CIFAR-10, `LR` is 0.001, and $\texttt{LR}_{\texttt{DGW}}$ is 0.0001. For BFFHQ, `LR` is 0.0001 and 0.0002 is for $\texttt{LR}_{\texttt{DGW}}$.

We utilize StepLR for learning rate scheduling, with a decaying step set to 10K for all datasets. The decay ratio is 0.5 for both C-MNIST and C-CIFAR-10 and 0.1 for BFFHQ. Following [38], we adjust the learning rate after performing feature augmentation.

We set the hyperparameters $(\lambda_{\text{re}}, \lambda_{\text{swap}_b}, \lambda_{\text{swap}}, \lambda_{\text{ent}})$ for our proposed loss functions. $(10, 10, 1, 0.01)$ is set for the ratio of 0.5% of C-MNIST, and $(15, 15, 1, 0.01)$ for the ratio of $1\%, 2\%$, and 5% of C-MNIST. We set $(1, 1, 1, 0.01)$ for C-CIFAR-10, and $(2, 2, 0.1, 0.01)$ for BFFHQ.

Our proposed mixup strategy uses the hyperparameter β to select the mixing coefficient $\alpha \sim \text{Beta}(\beta, \beta)$. For BFFHQ, we set 0.5, whereas 0.2 for C-MNIST and C-CIFAR-10.

We provide the scripts, including all hyperparameter setups, in our Git repository (Section A) to reproduce our performance evaluation.

D.5 Analysis for Interpretable Attribute Representation

Initialization of Concept Slots. The initialization of concept slots is crucial for our model's performance, tailoring the attention mechanisms to each dataset. We set the initial number of concept slots (C) as follows:

- For C-MNIST, C is set to 2, reflecting its simple attribute composition
- For C-CIFAR-10, C is set to 10, accommodating its diverse features
- For BFFHQ, C is set to 10, capturing a wide range of human facial features

Additional Visualization on C-MNIST dataset. Figure A-1 displays the attention masks $\mathbf{A}^i = \mathbf{A}(\mathbf{S}^i_{\text{latent}}, \mathbf{E}^i)$ generated by our broadcast scheme in the main text for C-MNIST, showing the model focuses on digit shapes, ignoring color. Fig. A-2 shows the attention masks $\mathbf{A}^b = \mathbf{A}(\mathbf{S}^b_{\text{latent}}, \mathbf{E}^b)$ generated by our proposed broadcast method in the main text, highlighting how the model responds

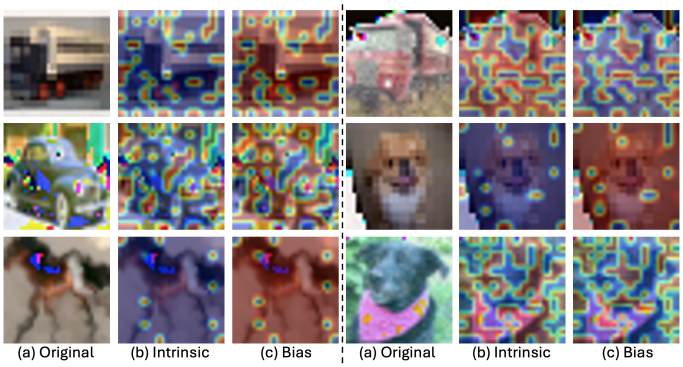


Figure A-3: Visualization of $\mathbf{A}^i$ and $\mathbf{A}^b$ for the C-CIFAR10 dataset

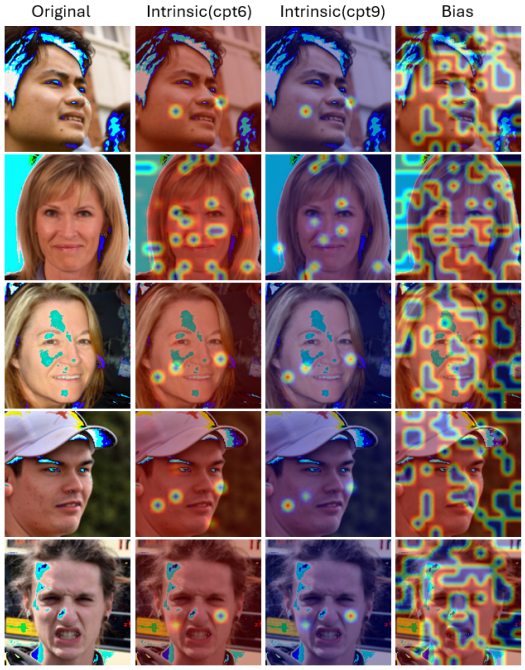


Figure A-4: Face images with attention masks. The first column shows the original image, the next two columns show attention masks $\mathbf{A}^i$ from concept slots 6 and 9, and the last column shows masks $\mathbf{A}^b$.

to color patterns. Similar colors, like the purple digits 2, 3, 0, and 6, have similar attention masks, indicating the model's sensitivity to color.

Visualization on Corrupted CIFAR-10 dataset. For the C-CIFAR-10 dataset, intrinsic masks focus on uncorrupted parts of the images (Fig. A-3(b)), highlighting true object features. For example, masks for a truck, car, dog, and horse highlight uncorrupted areas, avoiding noise. Bias masks, on the other hand, focus on corrupted areas, showing no overlap with intrinsic masks (Fig. A-3(c)). This complementary relationship illustrates the effective segregation of essential (intrinsic) and non-essential (biased) information.

Visualization on BFFHQ dataset. Figure A-4 shows DGW's behavior on the BFFHQ dataset, where the intrinsic components display complementary behavior within themselves (concept slots 6 and 9), focusing on specific facial features like cheeks for gender classification. This behavior is due to BFFHQ's focus on human facial shapes for gender classification, where the model prioritizes critical facial features, filtering out less relevant data.

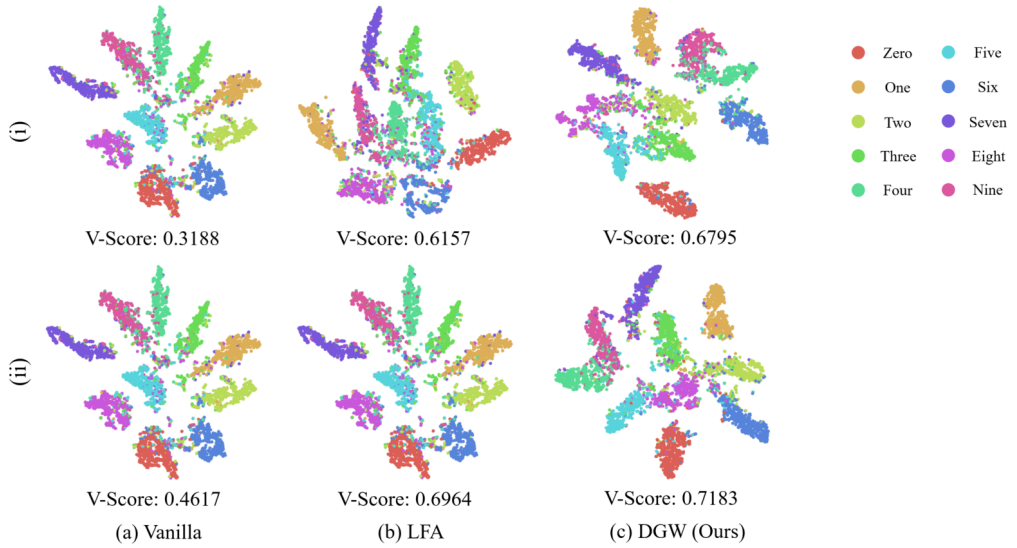


Figure A-5: t-SNE plots for intrinsic features on C-MNIST (with (i) 1.0% and (ii) 2.0% settings).

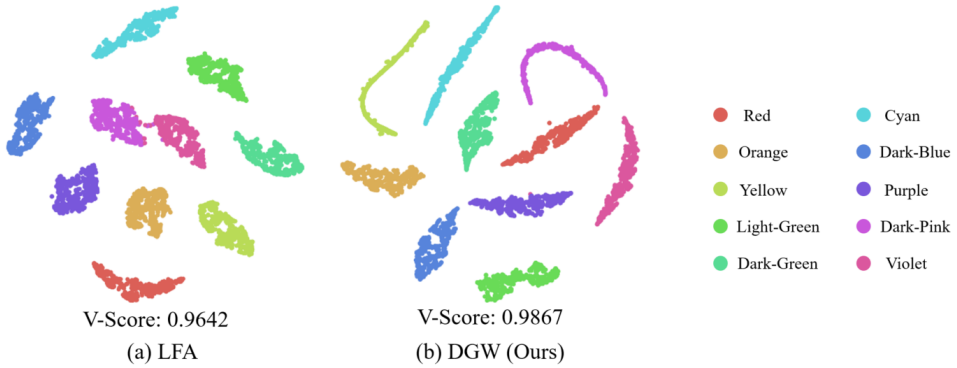


Figure A-6: t-SNE plots for bias features on C-MNIST (with 0.5% setting).

D.6 Quantitative and Qualitative Analysis

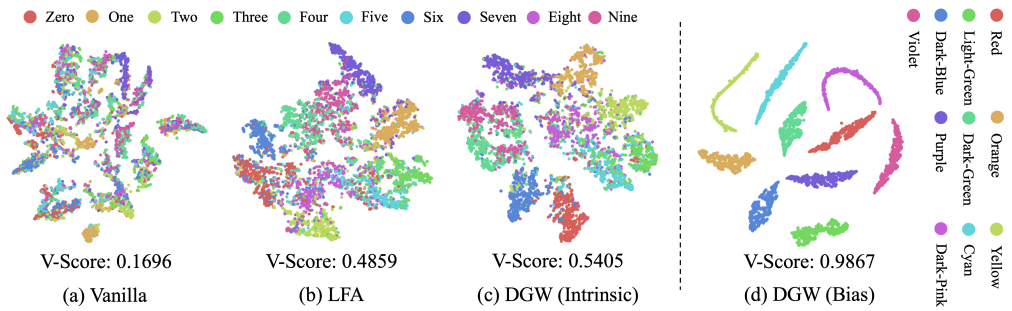
We provide additional analysis to compare our DGW (DGW+M in Table A-1) method with Vanilla and LFA [38].

t-SNE and Clustering. We measure clustering performance using t-SNE [45] and V-Score [53] on features from various models capturing intrinsic and bias attributes on C-MNIST. V-Score represents homogeneity and completeness, with higher values indicating better clustering. In Fig. A-7, our DGW's ϕ^i captures intrinsic attributes effectively, resulting in tighter clusters and better separation, as indicated by the V-Score. Bias attributes are well captured by the ϕ^b, as shown in Fig. A-7(d).

We provide more results with t-SNE plots and clustering scores with V-Score [53] as illustrated in Fig. A-5 and A-6. V-Score, a harmonic mean between homogeneity and completeness, is widely used to evaluate clustering. A higher V-Score indicates tighter intra-class clusters and better inter-class separation.

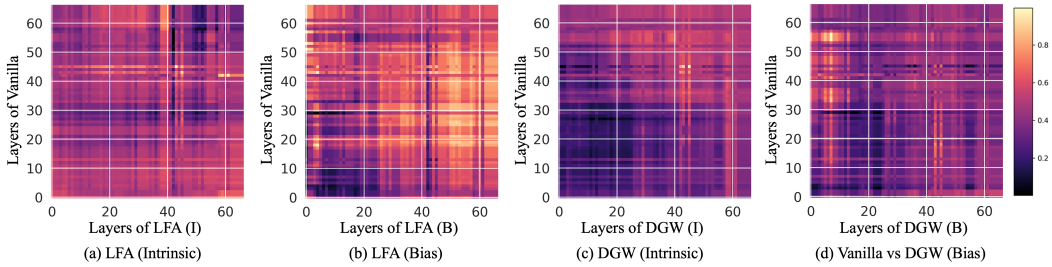
In Fig. A-5, intrinsic features from baselines and the intrinsic attribute encoder ϕ^i are used. It consistently shows a higher V-Score, implying better classification and intrinsic attribute capture compared to baselines. V-Scores are higher in setting (ii) than (i) because more bias-conflicting samples are used for training in setting (ii).

In Fig. A-6, features from the bias attribute capturing layer of LFA and the bias attribute encoder ϕ^b are utilized. It shows a higher V-Score compared to LFA, indicating more effective bias attribute



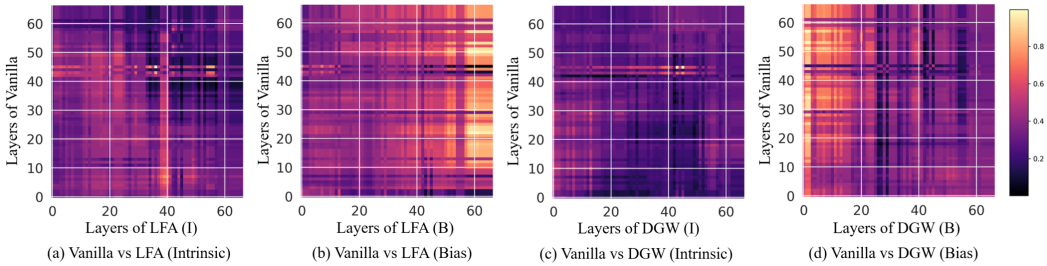
(a) Vanilla (b) LFA (c) DGW (Intrinsic) (d) DGW (Bias)

Figure A-7: t-SNE plots for intrinsic and bias features on C-MNIST (with 0.5% setting).



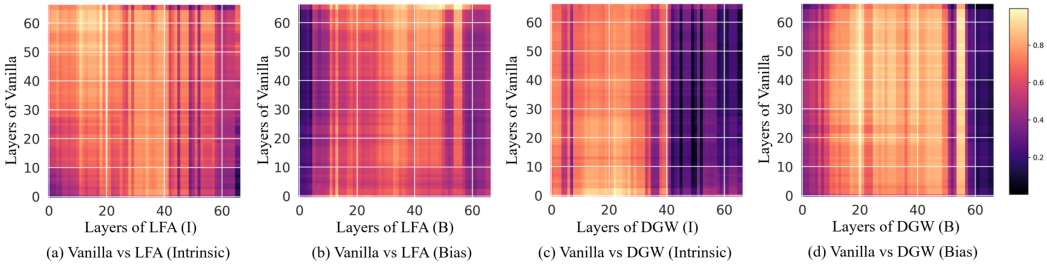
(a) LFA (Intrinsic) (b) LFA (Bias) (c) DGW (Intrinsic) (d) Vanilla vs DGW (Bias)

Figure A-8: Representations of similarities for vanilla and different methods with all pairs of layers on C-CIFAR-10 (0.5% setting). High similarity score denotes high values.



(a) Vanilla vs LFA (Intrinsic) (b) Vanilla vs LFA (Bias) (c) Vanilla vs DGW (Intrinsic) (d) Vanilla vs DGW (Bias)

Figure A-9: Representations of similarities for vanilla model and different methods with all pairs of layers on C-CIFAR-10 (5.0% setting). A high similarity score denotes high values.



(a) Vanilla vs LFA (Intrinsic) (b) Vanilla vs LFA (Bias) (c) Vanilla vs DGW (Intrinsic) (d) Vanilla vs DGW (Bias)

Figure A-10: Representations of similarities for vanilla model and different methods with all pairs of layers on BFFHQ (0.5% setting). A high similarity score denotes high values.

separation. Overall, our method outperforms baselines, demonstrating robust separation of intrinsic and bias attributes to improve debiasing process.

Model Similarity. We visualize model similarity using Centered Kernel Alignment (CKA) [51, 36, 9], comparing similarities between all pairs of layers for different models. In this analysis, I and B denote ϕ^i and ϕ^b. As shown in Fig. A-8, Vanilla and LFA possess similar weights across many layers, while DGW shows fewer similarities in both initial and deeper layers, indicating different behavior across layers compared to baselines.

Table A-3: ECE (%) and NLL under different settings on C-MNIST and C-CIFAR-10.

Dataset	C-MNIST								C-CIFAR-10							
Ratio (%)	0.5		1.0		2.0		5.0		0.5		1.0		2.0		5.0	
	ECE	NLL	ECE	NLL	ECE	NLL	ECE	NLL	ECE	NLL	ECE	NLL	ECE	NLL	ECE	NLL
Vanilla	10.9	**13.17**	7.97	**6.45**	5.70	**5.71**	9.54	4.10	13.75	5.99	13.14	9.87	12.25	6.65	13.76	5.99
LFA	4.35	67.72	2.79	36.46	2.09	18.35	7.59	**3.09**	12.09	5.81	**11.45**	7.27	10.25	5.14	7.56	3.09
DGW	**3.41**	271.71	**2.03**	143.36	**1.73**	41.44	**1.61**	20.19	**11.85**	**5.71**	11.53	**6.88**	**9.96**	**4.41**	**7.55**	**3.01**

We use Centered Kernel Alignment (CKA) [51, 36, 9] to visualize similarities between all pairs of layers in different models, helping us understand model behavior. The bias and intrinsic attribute encoders ϕ^b and ϕ^i in our approach are compared.

In Fig. A-9 and Fig. A-10, Vanilla and LFA models show similar weights in many layers, represented by bright colors. In contrast, our method shows significantly lower similarity values, indicating different weights and behaviors across layers compared to Vanilla and LFA. Our method affects deeper layers more, where the attention module is inserted, suggesting a distinct impact on model behavior.

Table A-2: ECE (%) and NLL under different settings on C-CIFAR-10.

Ratio (%)	0.5		1.0		2.0		5.0	
	ECE	NLL	ECE	NLL	ECE	NLL	ECE	NLL
Vanilla	13.75	5.99	13.14	9.87	12.25	6.65	13.76	5.99
LFA	12.09	5.81	**11.45**	7.27	10.25	5.14	7.56	3.09
DGW (Ours)	**11.85**	**5.71**	11.53	**6.88**	**9.96**	**4.41**	**7.55**	**3.01**

Model Reliability. We evaluate model generalizability using Expected Calibration Error (ECE) and Negative Log Likelihood (NLL) [21]. ECE measures calibration error, and NLL assesses probabilistic quality. As shown in Table A-2, DGW consistently has the lowest ECE and NLL, indicating better generalizability compared to baselines. To evaluate the generalizability of models, we measure Expected Calibration Error (ECE) and Negative Log Likelihood (NLL) [21], where ECE is to measure calibration error and NLL is to calculate the probabilistic quality of a model. In detail, ECE aims to evaluate whether the predictions of a model are reliable and accurate, which is a simple yet sufficient metric for assessing model calibration and reflecting model generalizability [21].

In Table A-3, our method consistently shows the lowest ECE, indicating better calibration and reliability. For C-MNIST, it presents a higher NLL compared to baselines. Since C-MNIST includes color bias only in the training set, it prevents overfitting by being less affected by bias, leading to better overall model performance. This trend is consistent across different settings in C-MNIST, providing insights into analyzing and explaining dataset bias types and complexity characteristics.

E Limitations.

We acknowledge that introducing our modules can increase training complexity, including model size and training time. This represents a trade-off between performance and decision-making transparency. While our additional overhead is minimal, further analysis is necessary to optimize and streamline the process.