

Appendix

A Reproducibility

All source codes, figures, models, etc. are available from https://github.com/PavlicLab/debiasing_global_workspace.

B Background

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 (Fodor and Pylyshyn, 1988; Goyal and Bengio, 2022; Greff et al., 2020). Pursuing representations that generalize compositionally has been a significant research topic, with object-centric representation learning (Burgess et al., 2019; Greff et al., 2019; Locatello et al., 2020; Chang et al., 2022; Jia et al., 2022) 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) (Locatello et al., 2020) 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 (Locatello et al., 2020). Some recent works on implementing a cognitive architecture using object-centric methods have been proposed (Hong et al., 2024; Didolkar et al., 2023). 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 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.

C.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)

C.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 (Nam et al., 2020; Kim et al., 2019; Li and Vasconcelos, 2019; Bahng et al., 2020; Darlow et al., 2020; Lee et al., 2021), 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 (Hendrycks and Dietterich, 2018), 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 (Krizhevsky and Hinton, 2009). We used the most severe level of corruption for the dataset, with the following bias-aligned and bias-conflicting samples:

- 0.5%: (44832:228)
- 1%: (44527:442)
- 2%: (44145:887)
- 5%: (42820:2242)

BFFHQ. The dataset is created by using the Flickr-Faces-HQ (FFHQ) Dataset (Karras et al., 2019), 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.

C.3 Image Preprocessing

Following Lee et al. (2021), 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.

C.4 Performance Evaluation

Training Details. For training, we use the Adam (Kingma and Ba, 2014) optimizer with default parameters (i.e., betas = (0.9, 0.999) and weight decay = 0.0) provided in the PyTorchTM framework. We define two different learning rates: LR_{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 LR_{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 LR_{DGW} is 0.0001. For BFFHQ, LR is 0.0001 and 0.0002 is for LR_{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 (Lee et al., 2021), we adjust the learning rate after performing feature augmentation.

We set the hyperparameters (λ_{re} , λ_{swap_b} , λ_{swap} , λ_{ent}) for our proposed loss functions (Section 3.3 in the main text). (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.

C.5 Analysis for Interpretable Attribute Representation

Initialization of Concept Slots. The initialization of concept slots is crucial for the performance of our model, 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

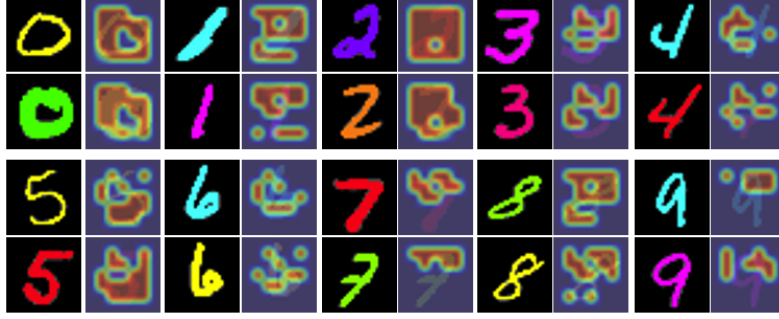


Figure A-1: Visualization of attention masks \mathbf{A}^i for the C-MNIST dataset

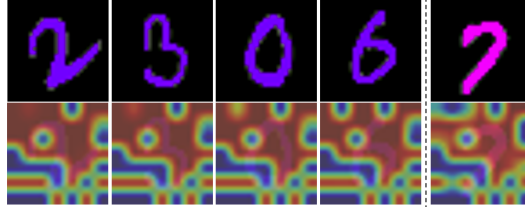


Figure A-2: Visualization from the C-MNIST dataset showing attention masks \mathbf{A}^b , highlighting color patterns. Digits in similar colors (e.g., 2, 3, 0, and 6) share similar attention mask patterns.

Additional Visualization on C-MNIST dataset. Figure A-1 displays the attention masks $\mathbf{A}^i = \mathbf{A}(\mathbf{S}_{\text{latent}}^i, \mathbf{E}^i)$ generated by eq. 3 in the main text for C-MNIST, showing the model focuses on digit shapes, ignoring color. Fig. A-2 shows the attention masks $\mathbf{A}^b = \mathbf{A}(\mathbf{S}_{\text{latent}}^b, \mathbf{E}^b)$ generated by eq. 3 in the main text, highlighting how the model responds 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 BFFHQ dataset. Figure A-3 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.

C.6 Quantitative and Qualitative Analysis

t-SNE and Clustering. We provide more results with t-SNE plots and clustering scores with V-Score (Rosenberg and Hirschberg, 2007) as illustrated in Fig. A-4 and A-5. 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-4, 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-5, 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 separation. Overall, our method outperforms baselines, demonstrating robust separation of intrinsic and bias attributes to improve debiasing process.

Model Similarity. We use Centered Kernel Alignment (CKA) (Raghu et al., 2021; Kornblith et al., 2019; Cortes et al., 2012) 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.

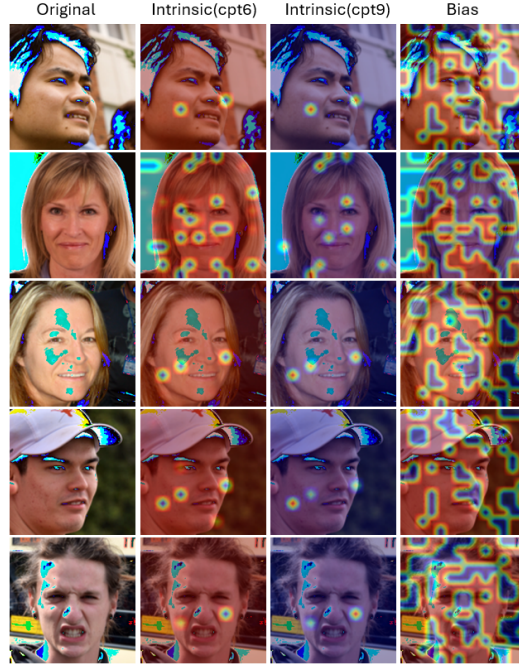


Figure A-3: Face images with attention masks. The first column shows the original image, the next two columns show attention masks \mathbf{A}^i from concept slots 6 and 9, and the last column shows masks \mathbf{A}^b .

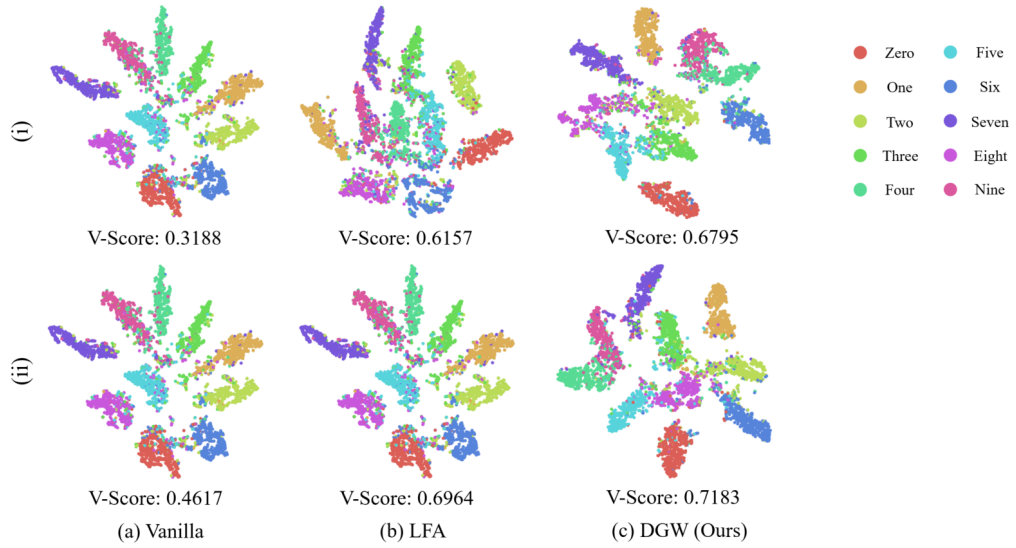


Figure A-4: t-SNE plots for intrinsic features on C-MNIST (with (i) 1.0% and (ii) 2.0% settings).

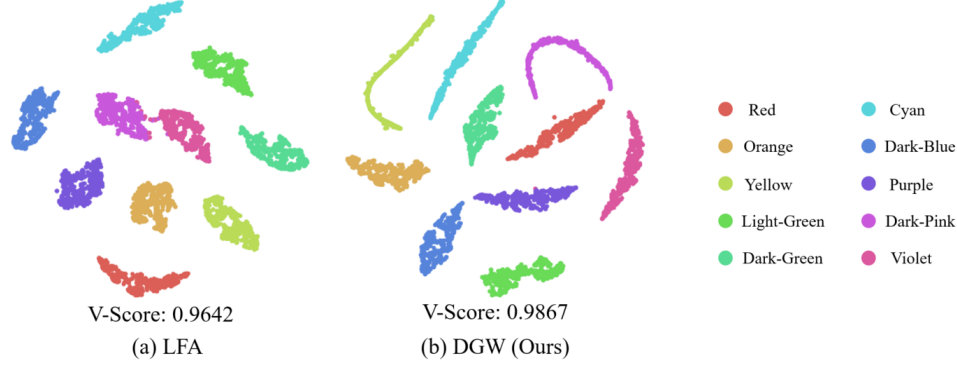


Figure A-5: t-SNE plots for bias features on C-MNIST (with 0.5% setting).

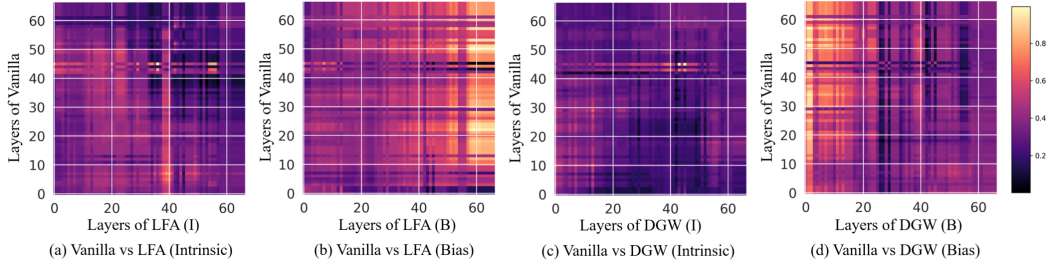


Figure A-6: Representations of similarities for vanilla model and different methods with all pairs of layers on C-CIFAR-10 (5.0% setting). A high similarity score denotes high values.

In Fig. A-6 and Fig. A-7, 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.

Model Reliability. To evaluate the generalizability of models, we measure Expected Calibration Error (ECE) and Negative Log Likelihood (NLL) (Guo et al., 2017), 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 (Guo et al., 2017).

In Table A-1, 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.

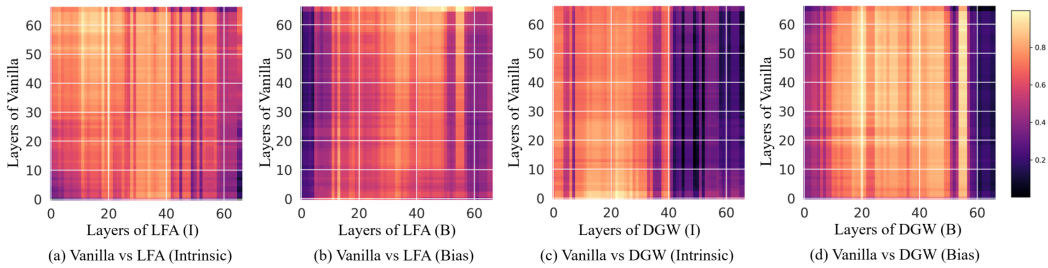


Figure A-7: Representations of similarities for vanilla model and different methods with all pairs of layers on BFFHQ (0.5% setting). A high similarity score denotes high values.

Table A-1: 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