1. ALGORITHM IMPLEMENTATION

The detail steps of seq-IIB is presented in Algorithm 1, where we train the feature extractor Φ, classifier ω in the first environment and update only g (Φ, ω is frozen) in the subsequent environments.

$$\mathcal{L} = \mathbb{CE}(Y, \hat{Y}) + \lambda_1 I(X; Z) + \lambda_2 I(Y; E|Z) + \mathcal{L}_f, \quad (1)$$

Eq. (1) corresponds to the loss function of seq-IIB in the main text.

Algorithm 1 Sequential Invariant Information Bottleneck (seq-IIB)

Training datasets: $\{D_e|e = 1, \ldots, n\} \in \mathcal{E}_{tr}$
Parameters: λ_1,λ_2: the lagrangian parameter, η: adjust the degree of compression, n_e: number of epochs for each environment's data.

$g \leftarrow 0, \{\Phi, \omega\} \leftarrow Initialize()$
if $e = 1$ **then**
  while $epoch \leq n_e$ **do**
    $\{g, \Phi, \omega\} \leftarrow$ Eq. (1)
  end while
else
  $\{\Phi, \omega\} \leftarrow Freeze(), g \leftarrow$ Eq. (1)
end if

Output: Network Φ, ω and mask parameters g.

2. EXPERIMENTAL SETTINGS

In the experiment, a three-layer fully connected network is used, and the number of neurons in the hidden layer is 200. We repeat each experiment 5 times and record the average results.

We first gradually increase the environmental data on ColorMNIST to verify the effectiveness of our method. In each training environment, the task is to classify whether the data is greater than or equal to 5, with less than 5 marked as 0 and the rest marked as 1. As in prior work, we add label noise by randomly flipping the original labels with 25% probability. The color of the images is determined according to the variable $color\_red$, which is a noisy label flipped with probability $p_c \in [0, 0.4]$. When there are more than 2 environments, a linear probability is used for each environment, i.e., $p_c(i) = (0.4 * i)/n$ where i is the environment index and n is the total number of environments. The color of the digit is red if $color\_red$ is True and green if $color\_red$ is False. Each train environment contains $50,000/n$ images of size 28×28 pixels, while the test environment contains $10,000$ images with the probability $p_c = 0.9$. In this case, the ideal performance of the model on the training and test sets is 75%

Similar to the setting of ColorMNIST, We set the labels of the first five classes of FashionMNIST and KMNIST [1] to

Fig. 1. Colored MNIST dataset

0 and the labels of the last five classes to 1 and add 25% label noise. Each experiment is set up with four training environments, and the p_c of each environment is different. The p_c of the test set is fixed at 0.9.

3. PARAMETER SETTING OF CONTRAST METHOD

For the comparison method we chose IRM, IRMG, IBIRM, IIB, GIB. IRM and IRMG used the best parameters suggested in the original. We use λ to control the IRM term and β to control the mutual information term or information entropy term. For IIB and IBIRM we set $\lambda \in \left[1, 10^2\right], \beta \in \left[10^{-4}, 10^{-3}\right]$, because the original get the best results in this interval. For GIB, we set $\beta \in \left[10^{-4}, 10^{-3}\right]$ and the rest of the parameters are the same as the default values in the original. All experiments are computed on a GPU with 12 Gb RAM.

4. REFERENCES

[1] Alex Lamb, Asanobu Kitamoto, David Ha, Kazuaki Yamamoto, Mikel Bober-Irizar, and Tarin Clanuwat, "Deep learning for classical japanese literature," *arXiv: Computer Vision and Pattern Recognition*, 2018.