SEQUENTIAL INVARIANT INFORMATION BOTTLENECK

Yichen Zhang[1], *Shujian Yu*[2*], *Badong Chen*[1*†]
[1]Xi'an Jiaotong University, Xi'an 710049, Shanxi, China
[2]Vrije Universiteit Amsterdam, 1081 HV Amsterdam, The Netherlands

ABSTRACT

Previous approaches to the problem of generalization for out-of-distribution (OOD) data usually assume that data from each environment is available simultaneously, which is unrealistic in real-world applications. In this paper, we develop a new framework termed the sequential invariant information bottleneck (seq-IIB) to improve the generalization ability of learning agents in sequential environments. Our main idea is to combine the merits of the famed Information Bottleneck (IB) principle with the Invariant Risk Minimization (IRM), such that the learning agent can gradually remove spurious features and remain *invariant* and *compact* task-relevant information in a sequential manner. Experimental results on three MNIST-like datasets show the effectiveness of our method.

Index Terms— Out-of-distribution generalization, sequential environments, IRM, Information Bottleneck.

1. INTRODUCTION

Despite significant progress in deep learning, the generalization of networks to out-of-distribution (OOD) data is still a primary challenge [1]. Learning *invariant* features across different environments is a dominating idea for most of existing approaches for OOD generalization. Given a network with a Markov chain $X \to Z \to Y$, in which X, Y and Z refer to respectively the input, class label and learned features, an intuitive idea to learn domain-invariant features is to seek the invariance of the covariate distribution $p(Z)$ [2][3]. However, minimizing the discrepancy of $p(Z)$ is not sufficient to guarantee generalization [4]. Hence, subsequent studies associate domain-invariant features with labels, expecting the network to learn the corresponding domain-invariant features for each class, which can be understood as finding the conditional domain-invariance $p(Z|Y)$ [5][6].

Compared with domain-invariant features, causal learning, such as the Invariant Causal Prediction (ICP) [7], expects the network to find the causal variables for prediction. Recently, Invariant Risk Minimization (IRM) [8] extends ICP to more practical settings. Essentially, both methods seek invariance in the conditional distribution $p(Y|Z)$.

However, all of the above methods require that data from multiple environments to be obtained simultaneously. In practical scenarios, we usually collect data from different environments sequentially [9][10]. For example, a patient's chest X-ray, usually sequentially obtained from different hospitals in different time periods [11]. In this scenario, learning invariant features becomes more difficult and remains a challenge.

In this paper, we extend IRM to sequential environment scenarios and develop sequential invariant information bottleneck (seq-IIB) that is able to gradually remove spurious features and remain only the compact and causal features in a sequential manner. Specifically, our main contributions are as follows:

- We develop the Sequential Invariant Information Bottleneck (seq-IIB), a novel framework for generalization in sequential environments that combines the merits of both IRM (to ensure *invariant*) and Information Bottleneck [12] (to ensure *compactness*).
- We demonstrate the invariance regularization term in IRM and IIB [13] can be simply estimated with matrix-based Rényi's α-order entropy function [14][15], without variational approximation and distributional assumption.
- Empirical result suggests that our method outperforms IRM, IIB, IBIRM [16] with a large margin under sequential environments. Moreover, it also performs better than GIB [17].

2. PRELIMINARIES

2.1. Problem Setup

Suppose that the training datasets $D_e := \{x_i^e, y_i^e\}_{i=1}^{n_e}$ is collected in multiple environments $e \in \mathcal{E}_{tr}$, and its samples obey the distribution $P_e = (X^e, Y^e)$. Our goal is to use these data to train a predictor $Y = f(X)$ that can perform well in the test set D_t, where the environment $\mathcal{E}_{te} \notin \mathcal{E}_{tr}$ of D_t cannot be seen, i.e., we wish to minimize:

$$\mathcal{R}^e(f), e \in \mathcal{E}_{te}, \tag{1}$$

where $\mathcal{R}^e := \mathbb{E}_{X^e,Y^e}[\ell(f(X^e), Y^e)]$ is the risk under environment e, ℓ denotes a loss function. This problem is also known as out-of-distribution (OOD) generalization.

In previous work, it was often assumed that all training environments $\{e_1, e_2, \ldots, e_i\} \in \mathcal{E}_{tr}$ were given simultane-

[*] To whom correspondence should be addressed (yusj9011@gmail.com; chenbd@mail.xjtu.edu.cn).

[†] B. Chen was funded by the National Natural Science Foundation of China with grant numbers (U21A20485, 61976175)

ously, which is unrealistic in practical scenario. In this work, we require that the environments are acquired sequentially.

2.2. Invariant Risk Minimization: An Information Theoretic View

IRM proposes the invariance of feature conditional label distribution. Specifically, it expects to find an invariant causal predictor $f = \omega \circ \Phi$. The objective of IRM is given by

$$\begin{gathered}\min_{\omega,\Phi} \sum_{e \in \varepsilon_{tr}} \mathcal{R}^e(\omega \circ \Phi), \\ \text{s.t.}, \omega \in \arg\min_{\hat{\omega}} \mathcal{R}^e(\hat{\omega} \circ \Phi),\end{gathered} \tag{2}$$

where Φ is a data representation function , ω is a linear classifier. Arjovsky *et al.* [8] instantiate IRM into the practical version IRMv1:

$$\min_{\Phi} \sum_{e \in \mathcal{E}_{tr}} \mathcal{R}^e(\Phi) + \lambda \cdot \left\| \nabla_{\omega|\omega=1.0} \mathcal{R}^e(\omega \circ \Phi) \right\|, \tag{3}$$

where $\omega = 1.0$ is a scalar and fixed *dummy* classifier.

However, some studies show that IRM is defective [18]. Ahuja *et al.* [16] indicate that invariant features require additional constraints. Hence, the authors combine IRM with Information bottleneck (IB) and propose IBIRM. IB requires the network to compress the representation of input X as much as possible while ensuring the predictive power to label Y. Let Z denote the feature extracted from X and $I(\cdot;\cdot)$ denote the mutual information, IB can be formulated by $\max\{I(Y;Z) - \lambda I(X,Z)\}$, in which λ is a Lagrange multiplier.

Recently, F. Huszár [19] gives the understanding of IRM from an information-theoretic perspective. Specifically, we can interpret IRM as finding an invariant feature $Z = \Phi(X)$ by:

$$\max_{\Phi} \{I[Y;Z] - \beta I[Y;E|Z]\}, \tag{4}$$

where E is the environment index. By applying IB to Eq. (6), the optimization objective can be written as:

$$\max_{\Phi} \{I[Y;Z] - \lambda_1 I[X;Z] - \lambda_2 I(Y;E \mid Z)\}, \tag{5}$$

we want to find compact and causal features by Eq. (5). Li *et al.* [13] implement this idea and propose IIB.

The estimation of mutual information in Eq. (5) is a challenge. To solve this problem, Li *et al.* [13] are inspired by VIB [20] and write the loss of IB as:

$$\begin{gathered}I(Z;Y) - \lambda I(Z;X) \\ \geq \mathbb{E}_{p_{x,y,z}}[\log q(y|z)] - \lambda \mathbb{E}_{p_{x,z}}[\log \frac{p(z|x)}{r(z)}]\end{gathered} \tag{6}$$

where $r(z)$ is the approximation to true marginal $p(z)$, and $q(y|z)$ to $p(y|z)$. They use an encoder of the form $p(z|x) = \mathcal{N}(z|f_e^{\mu}(x), f_e^{\Sigma}(x))$, where f_e is an MLP that outputs both the K-dimensional mean μ of z as well as the $K \times K$ covariance matrix Σ. Then, by the reparameterization trick, they get $q(z|x)\mathrm{d}(z) = q(\varepsilon)\mathrm{d}\varepsilon$, where $z = g(x, \varepsilon), \varepsilon \sim \mathcal{N}(0,1)$, so they optimize Eq. (6) by optimizing

$$\mathcal{L}_i(g, f_i) + \lambda \mathcal{L}_z(g), \tag{7}$$

where $\mathcal{L}_i = \min_{g,f_i} \mathbb{E}_{x,y}\left[L(y, f_i(g(x)))\right]$ and $\mathcal{L}_z = \min_g \mathbb{E}_x\left[KL\left[q(z|x;g)||r(z)\right]\right]$, g is the feature extractor, f_i is the classifier, and L is the *Cross-Entropy* loss.

They follow the rule of variational approximation [21] and decompose $I[Y;E|Z]$ into the difference of two cross-entropy losses:

$$\begin{aligned}I(Y;E|Z) &= H(Y|Z) - H(Y|E,Z) \\ &= \min_{f_i,\Phi} \mathbb{E}_{x,y}[L(y, f_i(\Phi(x)))] \\ &\quad - \min_{f_e,\Phi} \mathbb{E}_{x,y,e}[L(y, f_e(\Phi(x), e)]\end{aligned} \tag{8}$$

where f_i and f_e are classifiers, f_i take the feature Z as the input, f_e take the feature Z and the index of the environment e as the input.

However, the above is just an approximation. We will then introduce a much more accurate estimation to both mutual information and conditional mutual information terms.

3. METHODOLOGY

To extend Eq. (5) to sequential environments, we design a trainable soft mask that uses information from subsequent environments to gradually filter out spurious correlations learned from the first environment. During the filtering process, IRM and IB ensure the *invariant* and *compactness* of features respectively. We name this method Sequential Invariant Information Bottleneck (seq-IIB). The general architecture is illustrated in Fig. 1.

The implementation of seq-IIB has two important ingredients: the first one is the calculation of $I(X;Z)$ and $I(Y;E|Z)$, and the second one is the design of the mask.

3.1. Calculation of Mutual Information and Conditional Mutual Information

We use the matrix-based Rényi′s α-order entropy function to approximate $I(X;Z)$ and $I(Y;E|Z)$. Specifically, given N pairs of samples in e-th environment $\{\mathbf{x}^m, \mathbf{z}^m, \mathbf{y}^m, e\}_{m=1}^N$, $\mathbf{x}^m$ denotes the input sample, $\mathbf{z}^m$ denotes the feature of $\mathbf{x}^m$, and $\mathbf{y}^m$ denotes the target vector. We can view both $\mathbf{x}$, $\mathbf{z}$, $\mathbf{y}$ and $\mathbf{e}$ as random vectors. According to [14], the entropy of $\mathbf{x}$ can be defined over the eigen spectrum of a (normalized) Gram matrix $K_{\mathbf{x}} \in \mathbb{R}^{N \times N}(K_{\mathbf{x}}(m,n) = \kappa(\mathbf{x}^m, \mathbf{x}^n)$ and κ is a Gaussian kernel) as:

$$\begin{aligned}H_\alpha(A_{\mathbf{x}}) &= \frac{1}{1-\alpha}\log_2(\mathrm{tr}(A_{\mathbf{x}}^\alpha)) \\ &= \frac{1}{1-\alpha}\log_2(\sum_{m=1}^{N} \lambda_m(A_{\mathbf{x}})^\alpha),\end{aligned} \tag{9}$$

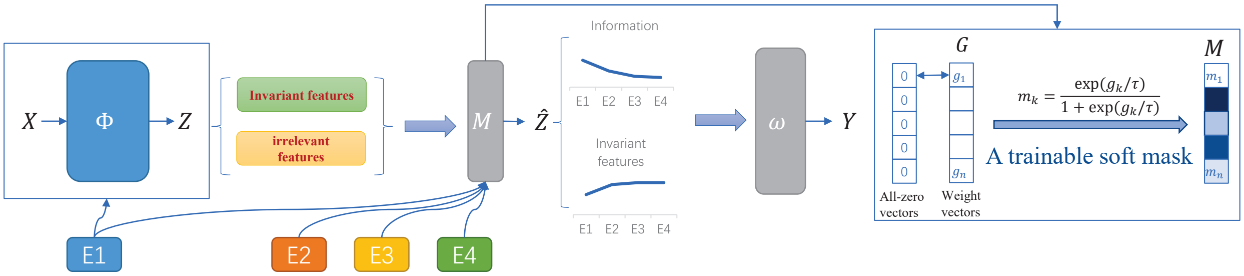

Fig. 1. General architecture. $E_i \in \mathcal{E}_{tr}$, M is a trainable soft mask with weight G. The network in the first environment E_1 extracts invariant features and irrelevant features by Φ. In the subsequent environments, the network gradually filters out irrelevant features by M and retains invariant features. The features $\hat{Z}$ filtered by M are entered into the classifier ω to get the prediction result Y.

where $\alpha \in (0,1) \cup (1,\infty)$. $A_\mathbf{x}$ is the normalized version of $K_\mathbf{x}$, i.e., $A_\mathbf{x} = \frac{K_\mathbf{x}}{\mathrm{tr}(K_\mathbf{x})}$. $\lambda_m(A_\mathbf{x})$ denotes the m-th eigenvalue of $A_\mathbf{x}$.

The entropy of $\mathbf{z}$ can be measured similarly by $A_\mathbf{z} \in \mathbb{R}^{N\times N}$. The joint entropy of $\mathbf{x}$ and $\mathbf{z}$ can be obtained by the following equation:

$$H_\alpha(A_\mathbf{x}, A_\mathbf{z}) = H_\alpha\left(\frac{A_\mathbf{x} \circ A_\mathbf{z}}{\mathrm{tr}(A_\mathbf{x} \circ A_\mathbf{z})}\right), \tag{10}$$

where $\circ$ denotes Hadamard (or element-wise) product.

Similarly, we can calculate the joint entropy between $\mathbf{y}$, $\mathbf{e}$ and $\mathbf{z}$ by the following equation:

$$H_\alpha(A_\mathbf{y}, A_\mathbf{e}, A_\mathbf{z}) = H_\alpha\left(\frac{A_\mathbf{y} \circ A_\mathbf{e} \circ A_\mathbf{z}}{\mathrm{tr}(A_\mathbf{y} \circ A_\mathbf{e} \circ A_\mathbf{z})}\right), \tag{11}$$

where $A_i \in \mathbb{R}^{N\times N}$ (when $\alpha \to 1$, H_α approximates Shannon entropy).

According to Shannon's chain rule [22], $I(X;Z)$ and $I(Y;E|Z)$ can be decomposed as:

$$I(X;Z) = H(A_\mathbf{x}) + H(A_\mathbf{z}) - H(A_\mathbf{x}, A_\mathbf{z}), \tag{12}$$

$$I(Y;E|Z) = H(Y,Z) + H(E,Z) - H(Z) - H(Y,E,Z), \tag{13}$$

in which H denotes entropy or joint entropy. The differentiability of matrix-based Rényi′s α-order entropy functional has been derived in [15].

3.2. A Trainable Soft Mask

We design the mask using the *gumbel-softmax* trick [23][24], as shown in Fig.1, in which:

$$\hat{Z} = M \odot Z, m_k = \frac{\exp(g_k/\tau)}{1+\exp(g_k/\tau)}, \tag{14}$$

where $M = (m_1 \dots m_n)$ is the mask matrix, $Z = \Phi(X)$ is the characteristic matrix, Φ denotes feature extractor, ω denotes classifier, τ is a hyperparameter to control the randomness in *gumbel-softmax*. g_k is a learnable weight that determines the value of m_k. After getting feature Z from $\Phi(X)$, Z will be filtered by mask M. The filtered feature $\hat{Z}$ enter the classifier ω and predict labels.

We can consider the value of the M as the probability p of the feature being selected. After training of the first environment, the network learns both invariant features and spurious or irrelevant features. In the subsequent environments, Φ and ω are frozen and we only train g. The network gradually decreases the p of the irrelevant features and increases the p of the invariant features.

We additionally add a simple regularization to avoid trivial solution:

$$\mathcal{L}_f = \|n_r/n_t - \eta\|, \tag{15}$$

where n_r and n_t denote respectively the number of remained features and all features (after training in the first environment). η is a hyperparameter that controls the degree of compression. Combining Eq. (5) and Eq. (15), the final loss function of seq-IIB becomes:

$$\mathcal{L} = \mathbb{CE}(Y,\hat{Y}) + \lambda_1 I(X;Z) + \lambda_2 I(Y;E|Z) + \mathcal{L}_f, \tag{16}$$

where $\hat{Y}$ is the model prediction, $\mathbb{CE}$ denotes cross-entropy loss. The detailed algorithm of seq-IIB is presented in the supplemental material[1].

4. EXPERIMENTS

In this section, we conduct two groups of experiments: 1) we validate the effectiveness of seq-IIB by comparing the accuracy of OOD data in a sequential environment scenario; 2) we then perform a sanity check to visually demonstrate that our model can learn invariant and compressed features sequentially.

4.1. Datasets and Experimental Setups

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.

[1] https://github.com/SJYuCNEL/seq-IIB

We gradually increase the environmental data on ColorMNIST. As in prior work [8], 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 the 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, while the test environment contains $10,000$ images with the probability $p_c = 0.9$.

Similar to the setting of ColorMNIST, We set the labels of the first five classes of FashionMNIST and KMNIST [25] to 0 and the labels of the last five classes to 1. Each experiment is set up with four training environments.

4.2. Experimental Results

Accuracy (%) of OOD data on ColorMNIST			
Method	2	4	8
ERM	27.9(1.2)	28.3(2.8)	18.8(1.5)
IRM	16.1(1.1)	16.6(1.1)	10.3(0.4)
IRMG	28.2(1.4)	23.9(2.1)	18.3(0.8)
IBIRM	21.6(2.5)	20.8(1.4)	13.4(5.2)
IIB	23.4(1.1)	23.8(0.9)	16.8(1.2)
GIB	59.2(2.2)	51.1(4.4)	48.6(3.9)
seq-IIB	**63.9(2.4)**	**60.9(2.9)**	**57.1(3.3)**

Table 1. Sequential Colored MNIST Dataset in 2, 4, 8 consecutive environments, over 5 evluations, where the probability of the first and last environments are 40%, 10% (5%).

Table 1 shows that the IRM-based methods fail to achieve generalization in the sequential environment scenarios, even lower than random prediction. GIB is lower than seq-IIB in the same settings, indicating that our method learns more invariant features.

Accuracy (%) of OOD data on MNIST-like datasets		
Method	FashionMNIST	KMNIST
ERM	26.2(1.9)	29.2(1.5)
IRM	14.8(0.8)	22.2(2.0)
IRMG	28.8(2.5)	31.7(2.2)
IBIRM	22.8(2.0)	25.7(1.5)
IIB	27.6(5.4)	26.6(1.7)
GIB	43.1(9.1)	47.5(4.3)
SEQ-IIB	**58.9(6.0)**	**54.2(2.4)**

Table 2. Sequential Colored MNIST-like Datasets in 4 consecutive environments, where the probability of the first and last environments are 40%, 10%.

Table 2 shows that our method can achieve the best results in different datasets.

4.3. A Sanity Check

We use the absolute value of the difference between the two Saliency maps [26] to indicate whether the network learns invariant features.

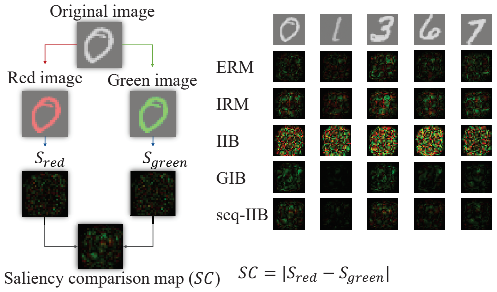

Fig. 2. Saliency comparison map (SC) of ColorMNIST dataset in 4 sequential environments. It is calculated from the Saliency map of the red data (S_{red}) and the Saliency map of the green data (S_{green}), where $SC = |S_{red} - S_{green}|$. The lower its brightness, the smaller the difference between the two Saliency maps and the more invariance is learned by network.

Fig. 2 shows that the average brightness of SC maps of the IRM-based method is higher than that of seq-IIB and even higher than that of ERM. Compared with GIB, seq-IIB does not perform the best in all data, but the results are more stable.

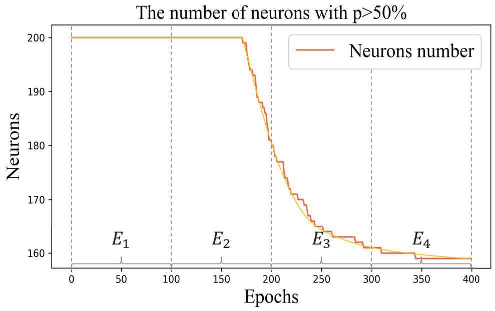

Fig. 3. The number of neurons with $p > 50\%$ during the training process, which can be seen as the change in information with the increase of environment.

Fig. 3 shows that with the increase of environment, the information on the network gradually decreases and eventually stabilizes. We consider that in this process, the network reduces irrelevant features and retains invariant features.

5. CONCLUSION AND FUTURE WORK

In this work, we extend IRM to sequential environment scenarios and propose our algorithm seq-IIB. First, we implement IRM in an information-theoretic way, which facilitates the combination of IRM with IB. Second, we design a trainable mask to turn the learning of subsequent environments into a feature filtering process. In future work, we hope to explore how to learn invariance in scenarios where there is clear boundaries on the environmental partition, which is more valuable in practice.

6. REFERENCES

[1] Zheyan Shen, Jiashuo Liu, Yue He, Xingxuan Zhang, Renzhe Xu, Han Yu, and Peng Cui, "Towards out-of-distribution generalization: A survey," *arXiv preprint arXiv:2108.13624*, 2021.

[2] Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan, "A theory of learning from different domains," *Machine Learning*, 2010.

[3] Krikamol Muandet, David Balduzzi, and Bernhard Schölkopf, "Domain generalization via invariant feature representation," in *International Conference on Machine Learning*. PMLR, 2013, pp. 10–18.

[4] Han Zhao, Remi Tachet Des Combes, Kun Zhang, and Geoffrey Gordon, "On learning invariant representations for domain adaptation," in *International Conference on Machine Learning*. PMLR, 2019, pp. 7523–7532.

[5] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky, "Domain-adversarial training of neural networks," *Journal of Machine Learning Research*, 2015.

[6] Mingsheng Long, Zhangjie Cao, Jianmin Wang, and Michael I Jordan, "Conditional adversarial domain adaptation," *Advances in neural information processing systems*, vol. 31, 2018.

[7] Jonas Peters, Peter Bühlmann, and Nicolai Meinshausen, "Causal inference by using invariant prediction: identification and confidence intervals," *Journal of The Royal Statistical Society Series B-statistical Methodology*, 2016.

[8] Martin Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz, "Invariant risk minimization," *arXiv preprint arXiv:1907.02893*, 2019.

[9] Zhiyuan Chen and Bing Liu, "Lifelong machine learning," *Synthesis Lectures on Artificial Intelligence and Machine Learning*, vol. 12, no. 3, pp. 1–207, 2018.

[10] Zheda Mai, Ruiwen Li, Jihwan Jeong, David Quispe, Hyunwoo Kim, and Scott Sanner, "Online continual learning in image classification: An empirical survey," *Neurocomputing*, vol. 469, pp. 28–51, 2022.

[11] Matthias Lenga, Heinrich Schulz, and Axel Saalbach, "Continual learning for domain adaptation in chest x-ray classification," in *Medical Imaging with Deep Learning*. PMLR, 2020, pp. 413–423.

[12] Naftali Tishby, Fernando C Pereira, and William Bialek, "The information bottleneck method," in *Proc. 37th Annual Allerton Conference on Communications, Control and Computing, 1999*, 1999, pp. 368–377.

[13] Bo Li, Yifei Shen, Yezhen Wang, Wenzhen Zhu, Dongsheng Li, Kurt Keutzer, and Han Zhao, "Invariant information bottleneck for domain generalization," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2022, vol. 36, pp. 7399–7407.

[14] Luis Gonzalo Sanchez Giraldo, Murali Rao, and Jose C Principe, "Measures of entropy from data using infinitely divisible kernels," *IEEE Transactions on Information Theory*, vol. 61, no. 1, pp. 535–548, 2014.

[15] Shujian Yu, Francesco Alesiani, Xi Yu, Robert Jenssen, and Jose Principe, "Measuring dependence with matrix-based entropy functional," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2021, vol. 35, pp. 10781–10789.

[16] Kartik Ahuja, Ethan Caballero, Dinghuai Zhang, Jean-Christophe Gagnon-Audet, Yoshua Bengio, Ioannis Mitliagkas, and Irina Rish, "Invariance principle meets information bottleneck for out-of-distribution generalization," *Advances in Neural Information Processing Systems*, vol. 34, pp. 3438–3450, 2021.

[17] Francesco Alesiani, Shujian Yu, and Xi Yu, "Gated information bottleneck for generalization in sequential environments," in *2021 IEEE International Conference on Data Mining (ICDM)*. IEEE, 2021, pp. 1–10.

[18] Elan Rosenfeld, Pradeep Ravikumar, and Andrej Risteski, "The risks of invariant risk minimization," *arXiv preprint arXiv:2010.05761*, 2020.

[19] Ferenc Huszár, "Invariant risk minimization: An information theoretic view," `https://www.inference.vc/invariant-risk-minimization/`.

[20] Alexander A Alemi, Ian Fischer, Joshua V Dillon, and Kevin Murphy, "Deep variational information bottleneck," *arXiv preprint arXiv:1612.00410*, 2016.

[21] Farzan Farnia and David Tse, "A minimax approach to supervised learning," *Advances in Neural Information Processing Systems*, vol. 29, 2016.

[22] David JC MacKay, David JC Mac Kay, et al., *Information theory, inference and learning algorithms*, Cambridge university press, 2003.

[23] Eric Jang, Shixiang Gu, and Ben Poole, "Categorical reparameterization with gumbel-softmax," *arXiv preprint arXiv:1611.01144*, 2016.

[24] Robin Dupont, Mohammed Amine Alaoui, Hichem Sahbi, and Alice Lebois, "Extracting effective subnetworks with gumebel-softmax," *arXiv preprint arXiv:2202.12986*, 2022.

[25] 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.

[26] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman, "Deep inside convolutional networks: Visualising image classification models and saliency maps," *arXiv preprint arXiv:1312.6034*, 2013.