Supplementary Materials of Manuscript Entitled "Domain Generalization Study of Empirical Risk Minimization from Causal Perspectives"

VII. Proofs of Theories

In this section, the proofs of theories in the paper are offered. Note that equations, tables, and figures, etc., in the Supplementary Materials are all numbered consecutively to those in the paper, while notations and references are consistent with those in the paper.

A. Proof of Lemma 1

First, it is known that the information flow is non-negative, i.e., $\mathcal{I}\left(A \mapsto B \mid do(C)\right) \geq 0$, since the relative entropy is non-negative [25].

Then, we are going to prove the lemma by contradiction. Assume that, given a DAG and its three disjoint sets of nodes A, B, and C, if there is a directed path from A to B unblocked by C, then $\mathcal{I}\left(A \to B \mid do(C)\right) = 0$. By Proposition 1, it implies that $A \perp_{ud} B \mid C$, which means C blocks all directed paths from A to B. Thus, it contradicts that there is a directed path from A to B unlocked by C. □

B. Proof of Proposition 2

1) *Proof of P1*

When C is a null set, we can reformulate (5) as follows.

$$\begin{aligned} &A \perp_{ud} B \Leftrightarrow A \perp_{do} B \Leftrightarrow \\ &p\left(b \mid do(a)\right) = \sum_{a \in A} p(a)\, p\left(b \mid do(a)\right), \\ &\forall a \in A, \forall b \in B \end{aligned} \tag{12}$$

In addition, we can also rewrite the information flow in (6) as the full information flow as follows.

$$\begin{aligned} &\mathcal{I}\left(A \mapsto B\right) \\ &= \sum_{a \in A} p(a) \sum_{b \in B} p\left(b \mid do(a)\right) \\ &\log \frac{p\left(b \mid do(a)\right)}{\sum_{a \in A} p(a)\, p\left(b \mid do(a)\right)} \end{aligned} \tag{13}$$

Thus, by the equality condition of relative entropy [25], we have

$$\mathcal{I}\left(A \mapsto B\right) = 0 \Leftrightarrow A \perp_{do} B \Leftrightarrow A \perp_{ud} B \tag{14}$$

Then, by the definition of *ud*-separation, there are no directed paths from A to B, which is equivalent to removing all directed paths from A to B without changing the causal graph. Hence, the Markov condition of the causal graph is intact. □

2) *Proof of P2*

We will employ Rule 2 of *do* Calculus (Theorem 3.4.1 of [21]). Hence, we summarize Rule 2 as follows. Given a DAG G and two disjoint sets of nodes A and B, we denote $G(\underline{B})$ as a DAG by deleting all the arrows emitting from nodes of B. The Rule 2 says that if A is *d*-separated from B in $G(\underline{B})$, i.e., $\left(A \perp_d B\right)_{G(\underline{B})}$, then

$$p\left(A \mid do(B)\right) = p\left(A \mid B\right) \tag{15}$$

Thus, if the causal graph is $a \to b$, then $\left(b \perp_d a\right)_{G(\underline{a})}$. Let a' and b' denote all possible values of nodes a and b, respectively. It follows that

$$\begin{aligned} &\mathcal{I}\left(a \mapsto b\right) \\ &= \sum_{a'} p(a) \sum_{b'} p\left(b \mid do(a)\right) \\ &\log \frac{p\left(b \mid do(a)\right)}{\sum_{a'} p(a)\, p\left(b \mid do(a)\right)} \\ &= \sum_{a'} p(a) \sum_{b'} p\left(b \mid a\right) \\ &\log \frac{p\left(b \mid a\right)}{\sum_{a'} p(a)\, p\left(b \mid a\right)} \\ &= \mathcal{I}\left(a : b\right) \end{aligned} \tag{16}$$

□

3) *Proof of P3*

It is obvious that a has no ancestors and b has no descendants since a is the root and b is the sink. $\mathcal{I}\left(a \mapsto b\right)$ is only dependent on a and b. Hence, this condition is trivially satisfied. □

4) *Proof of P4*

First, the setting is that A are roots connected via directed paths with sinks B. In addition, $a \in A$ and $b \in B$. The setting implies the followings:

(1) $a \perp_d Pa\left(b \setminus a \mapsto b\right)$ due to colliders formed with b $\Rightarrow$ $\mathcal{I}\left(a : b\right) = \mathcal{I}\left(a : b \mid Pa\left(b \setminus a \mapsto b\right)\right)$.

(2) If there are directed paths from a to b, then $\left(b \perp_d a\right)_{G(\underline{a})}$. Thus, by (16), we have $\mathcal{I}\left(a \mapsto b\right) = \mathcal{I}\left(a : b\right)$. □

5) *Proof of P5*

$\mathcal{I}\left(A \mapsto B\right) = 0 \Leftrightarrow A \perp_{do} B \Leftrightarrow A \perp_{ud} B$. Hence, there are no directed paths from A to B. For $a \in A$ and $b \in B$, $\mathcal{I}\left(a \mapsto b\right) = 0$. □

C. Proof of Lemma 2

We continue to follow the same rule in (15). Let G be the DAG shown in Fig. 2. Since $\left(X_O^e \perp_d Y\right)_{G(\underline{Y})}$, we have

$$\begin{aligned}
&\mathcal{I}\left(Y \mapsto X_O^e \mid C^e\right) \\
&= \sum_{c^e \in C^e} p\left(c^e\right) \sum_{\mathrm{y} \in Y} p\left(\mathrm{y} \mid c^e\right) \sum_{x_O^e \in X_O^e} p\left(x_O^e \mid do(y), c^e\right) \\
&\log \frac{p\left(x_O^e \mid do(y), c^e\right)}{\sum_{\mathrm{y} \in Y} p\left(\mathrm{y} \mid c^e\right) p\left(x_O^e \mid do(\mathrm{y}), c^e\right)} \\
&= \sum_{c^e \in C^e} p\left(c^e\right) \sum_{\mathrm{y} \in Y} p\left(\mathrm{y} \mid c^e\right) \sum_{x_O^e \in X_O^e} p\left(x_O^e \mid y, c^e\right) \\
&\log \frac{p\left(x_O^e \mid y, c^e\right)}{\sum_{\mathrm{y} \in Y} p\left(\mathrm{y} \mid c^e\right) p\left(x_O^e \mid \mathrm{y}, c^e\right)} \\
&= \mathcal{I}\left(Y : X_O^e \mid C^e\right)
\end{aligned} \tag{17}$$

□

D. Proof of Corollary 2

Recall Rule 2 of *do* Calculus (Theorem 3.4.1 of [21]), we have $\left(X_N^e \perp_d Y\right)_{G(\underline{Y})}$. It then follows similarly as (17). □

E. Proof of Theorem 1

First, let C be a null set, i.e., $C = \varnothing$, in Lemma 1. Then, we know that both $\mathcal{I}\left(Y \mapsto X_O^e\right)$ and $\mathcal{I}\left(Y \mapsto X_N^e\right)$ are non-negative, which implies that X_O^e and X_N^e both contain information of Y. However, the causal information X_N^e receives from Y is only mediated by C^e. By Corollary 2,

$$\mathcal{I}\left(Y \mapsto X_N^e \mid C^e\right) = \mathcal{I}\left(Y : X_N^e \mid C^e\right) \tag{18}$$

Since $Y \perp_d X_N^e \mid C^e$, $\mathcal{I}\left(Y : X_N^e \mid C^e\right) = 0$. Hence,

$$\mathcal{I}\left(Y \mapsto X_N^e \mid C^e\right) = 0 \tag{19}$$

Similarly, since $Y \not\perp_d X_O^e \mid C^e$, $\mathcal{I}\left(Y : X_O^e \mid C^e\right) > 0$. By Lemma 2, we have

$$\mathcal{I}\left(Y \mapsto X_O^e \mid C^e\right) = \mathcal{I}\left(Y : X_O^e \mid C^e\right) > 0 \tag{20}$$

Therefore,

$$\mathcal{I}\left(Y \mapsto X_O^e \mid C^e\right) > \mathcal{I}\left(Y \mapsto X_N^e \mid C^e\right) \tag{21}$$

From the causal perspective, it implies that, when observing C^e, X_O^e still has more causal information about Y than that X_N^e has about Y. Thus, the model may favor X_O^e rather than X_N^e.

Next, we offer a more formal proof from the information theoretic perspective. From the above, we know that

$$\mathcal{I}\left(Y : X_O^e \mid C^e\right) > \mathcal{I}\left(Y : X_N^e \mid C^e\right) \tag{22}$$

Let X_Z be the feature to be learned by an encoder $X_Z = h_\theta\left(X^e\right)$ and then be used to predict labels by a classifier $\hat{Y} = g_\omega\left(X_Z\right)$, where X^e is the input data, θ and ω are model parameters, and $\hat{Y}$ is the predicted label.

The mutual information between X_Z and Y given C^e is decomposed as follows.

$$\begin{aligned}
&\mathcal{I}\left(Y : X_Z \mid C^e\right) \\
&= \mathcal{H}\left(Y \mid C^e\right) - \mathcal{H}\left(Y \mid X_Z, C^e\right) \\
&\geq \mathcal{H}\left(Y \mid C^e\right) - \mathcal{H}\left(Y \mid X_Z\right)
\end{aligned} \tag{23}$$

where $\mathcal{H}$ denotes the Shannon's entropy and the inequality comes from the fact that the conditioning reduces entropy. Note that $\mathcal{H}\left(Y \mid C^e\right)$ is a constant since Y and C^e are fixed samples in the model learning process [23]. Then, minimizing the objective $\mathcal{H}\left(Y \mid X_Z\right)$ will have the similar effect of maximizing the objective $\mathcal{I}\left(Y : X_Z \mid C^e\right)$ for model learning since both objectives only matter on the conditional value of X_Z and C^e. In addition, C^e represents fixed samples and X_Z varies due to variational functionals in the model learning process. Therefore, if $X_Z = X_O^e$ is a better choice than $X_Z = X_N^e$ in terms of the objective $\mathcal{I}\left(Y : X_Z \mid C^e\right)$, it can be speculated that $X_Z = X_O^e$ is also a better choice than $X_Z = X_N^e$ in terms of the objective $\mathcal{H}\left(Y \mid X_Z\right)$. This reasoning can be also understood by considering the information Venn diagram of three variables Y, X_Z and C^e when two circles of Y and C^e are fixed and the left one varies in position [39].

Note that the cross-entropy used for prediction can be decomposed into two terms as follows.

$$\mathcal{H}\left(Y; \hat{Y} \mid X_Z\right) = \mathcal{H}\left(Y \mid X_Z\right) + D_{KL}\left(Y \,\|\, \hat{Y} \mid X_Z\right) \tag{24}$$

Based on the two-step model learning perspective (Lemma 2 of [23]), $\mathcal{H}\left(Y \mid X_Z\right)$ corresponds to the learning of the encoder $h_\theta(\cdot)$, while the KL divergence $D_{KL}\left(Y \,\|\, \hat{Y} \mid X_Z\right)$ corresponds to the learning of the classifier $g_\omega(\cdot)$. Note that X_O^e and X_N^e both contain information of Y as stated at the beginning of the proof. Hence, it is possible to make the corresponding KL divergence in (24) similarly approximate to zero given an ideal classifier [23]. Finally, we would expect that X_O^e, compared to X_N^e, will be selected by the encoder and X_O^e also offers lower cross-entropy. □

F. Proof of Corollary 3

It is known from Theorem 1 that X_O^e will be learned. If C^e and X_O^e is not *ud*-separated, the cross-entropy $\mathcal{H}\left(Y; \hat{Y} \mid X_Z = X_O^e\right)$ will be domain-dependent. It implies that the model fails at DG. □

G. Proof of Corollary 4

If C^e and X_O^e is *ud*-separated, we can show that X_O (i.e., X_O^e without the domain influence) will be learned. This can be seen as follows.

First, due to the symmetric property of mutual information and the correspondence between X_O and Y assumed in

TABLE III
HYPERPARAMETER SEARCH GRIDS USED IN EACH EXPERIMENT

Hyperparameter	Default value	Search grid
Training step	1000	[500, 1000, …, 5000]
Learning rate	0.001	$10^r, r \sim U(-4.5,-2.5)$
Weight decay	0	$10^r, r \sim U(-8,-4)$
Batch size	64	$\text{int}(10^r), r \sim U(3,9)$

The $10^r, r \sim U(-4.5,-2.5)$ means drawing r uniformly at random from the range $(-4.5,-2.5)$. int() means taking the integer value.

Assumption 1, the mutual information will be equal to the entropy as follows.

$$\mathcal{I}\left(X_O : Y\right) = \mathcal{I}\left(Y : X_O\right) = \mathcal{I}\left(Y : L\left(X_O\right)\right) = \mathcal{H}\left(Y\right) \quad (25)$$

Then, we consider the mutual information between Y and C^e as follows.

$$\begin{aligned} &\mathcal{I}\left(Y : C^e\right) \\ &= \mathcal{H}\left(Y\right) - \mathcal{H}\left(Y \mid C^e\right) \\ &\le \mathcal{H}\left(Y\right) \end{aligned} \quad (26)$$

where the inequality comes from the fact that the conditional entropy $\mathcal{H}\left(Y \mid C^e\right) \ge 0$.

Note that the path from Y to X_N^e is a Markov chain, by data processing inequality [25], we have

$$I\left(Y : C^e\right) \ge I\left(Y : X_N^e\right) \quad (27)$$

The equality in (27) holds if and only if $I\left(Y : C^e \mid X_N^e\right) = 0$ i.e., $C^e \to X_N^e \to Y$ forms a Markov chain ($Y \perp_d C^e \mid X_N^e$), which is not true based on the causal graph in Assumption 1. Considering the results in (25), (26), and (27), we obtain

$$\mathcal{I}\left(Y : X_O\right) > \mathcal{I}\left(Y : X_N^e\right) \quad (28)$$

Then, we can follow the same proof starting from (23) of Theorem 1 without conditioning on C^e, which results in that X_O will be learned and X_N^e will be ignored by the encoder. Therefore, $\mathcal{H}\left(Y; \hat{Y} \mid X_Z = X_O\right)$ is the same across domains. In addition, $\mathcal{H}\left(Y \mid X_Z = X_O\right) = 0$ and $D_{KL}\left(Y \,\|\, \hat{Y} \mid X_Z = X_O\right) = 0$ since X_O is the casual feature. Hence, $\mathcal{H}\left(Y; \hat{Y} \mid X_Z = X_O\right)$ is the smallest across domains. It implies that the model succeeds at DG. □

VIII. EXPERIMENT DETAIL

A. DomainBed Experiment Principle and Simulation Principle

Our experiments follow the DomainBed experiment principle proposed in [14]. For each model on each generalization task, each hyperparameter is selected based on conducting random searches 20 times over a given search grid. Each set of hyperparameters is tested 3 times independently. The model is selected based on the training domain validation and the test domain validation. The training domain validation means that 20% training domain data is reserved as the validation set. The test domain validation means that 20 % test domain data is reserved as the validation set. The similar search grids of hyperparameters employed in this study are presented in TABLE III.

TABLE IV
VARIATIONAL AUTOENCODER-CLASSIFIER FOR MNIST BASED EXPERIMENTS

Subnetwork	Layer	Layer function
Encoder	E1	Flatten()
	E2	$\text{Fc_Bn_ReLu}\left(in = x_{dim}, out = x_{dim}/2\right)$
	E3	$\text{Fc_Bn_ReLu}\left(in = x_{dim}/2, out = x_{dim}/4\right)$
	E4-1	$\text{Fc}\left(in = x_{dim}/4, out = x_{dim}/8\right)$
	E4-2	$\text{Fc}\left(in = x_{dim}/4, out = x_{dim}/8\right)$
Decoder	D1	Reparam()
	D2	$\text{Fc_Bn_ReLu}\left(in = x_{dim}/8, out = x_{dim}/4\right)$
	D3	$\text{Fc_Bn_ReLu}\left(in = x_{dim}/4, out = x_{dim}/2\right)$
	D4	$\text{Fc}\left(in = x_{dim}/2, out = x_{dim}\right)$
	D5	Sigmoid(UnFlatten())
Classifier	C1	$\text{Fc_Bn_ReLu}\left(in = x_{dim}/8, out = x_{dim}/16\right)$
	C2	$\text{Fc_Bn_ReLu}\left(in = x_{dim}/16, out = x_{dim}/32\right)$
	C3	$\text{Fc_Bn_ReLu}\left(in = x_{dim}/32, out = 2\right)$
	C4	Softmax()

Flatten() is an flatterning operator. The reverse is UnFlatten(). Fc_Bn_ReLu() means the compostion of a layer of fully connected network Fc(), a batch norm operation, and a relu function. $x_{\text{dim}} = 1568$. Sigmoid() is a sigmoid funciotn. Softmax() is the softmax function. E4-1 and E4-2 correspond to the mean and variance latent codes. Reparam() is the reparameterization of the latent codes.

The simulation principle is detailed as follows based on the proposed causal graph and the base datasets.

(1) There is a correspondence between the causal feature X_O (i.e., the interested yellow object) and the binary label Y (0 or 1) via the labeling mechanism.

$$Y \doteq L\left(X_O\right) \quad (29)$$

(2) The label then interacts with the domain noise N_1^e to produce spurious influencer C^e.

$$C^e \leftarrow Y \oplus N_1^e \quad (30)$$

$$N_1^e \sim \mathcal{B}\left(e\right) \quad (31)$$

where $\oplus$ denotes exclusive OR (i.e., XOR). N_1^e follows a Bernoulli distribution $\mathcal{B}\left(e\right)$ with probability e in giving the values of 1s. This process can be regarded as flipping the label with probability e to obtain C^e.

(3) The spurious influencer C^e is then applied to transform the causal feature X_O and the domain noise N_2^e into the transformed causal feature X_O^e (i.e., the re-colored object) and the transformed spurious feature X_N^e (i.e., the colored noise), respectively.

$$X_O^e \leftarrow T\left(X_O, C^e\right) \quad (32)$$

$$X_N^e \leftarrow T\left(N_2^e, C^e\right) \quad (33)$$

$$N_2^e = \left(1-e\right) \times 10 N\left(0,1\right) \quad (34)$$

where $T\left(\cdot\right)$ presents a function setting the red channel to zero if $C^e = 1$ and setting the green channel to zero if $C^e = 0$. N_2^e is the Gaussian noise with the domain dependent variance. $N\left(0,1\right)$ denotes the standard Gaussian variable.

TABLE V
DOMAIN GENERALIZATION RESULTS BASED ON TRAINING DOMAIN VALIDATION

Base	**Data settings**	**TS 1** $e_{test}=+90\%$	**TS 2** $e_{test}=+80\%$	**TS 3** $e_{test}=-90\%$	**DG:** Yes/No/N.A.	Δ^*	**Avg**
Data Base: MNIST	DS 1 $\not\rightarrow X_O \not\rightarrow$	94.4 +/- 0.5	95.2 +/- 0.4	94.6 +/- 0.4	N.A. (Baseline)	0.8	94.7
	DS 2 $X_N^e \leftarrow C^e \rightarrow X_O^e$	90.7 +/- 1.0	91.3 +/- 0.8	72.6 +/- 1.7	No (Thm.1 & Cllry.3)	18.7	84.8
Model Base: VAEC	DS 3 $X_N^e \not\leftarrow C^e \rightarrow X_O^e$	89.9 +/- 0.4	91.6 +/- 0.8	76.5 +/- 1.9	No (Cllry.3)	15.1	86.0
	DS 4 $X_N^e \leftarrow C^e \not\rightarrow X_O^e$	94.9 +/- 0.6	95.2 +/- 0.2	94.4 +/- 0.2	Yes (Cllry.4)	0.8	94.8
Data Base: Fashion MNIST	DS 1 $\not\rightarrow X_O \not\rightarrow$	91.9 +/- 0.3	93.4 +/- 0.1	92.2 +/- 0.3	N.A. (Baseline)	1.5	92.5
	DS 2 $X_N^e \leftarrow C^e \rightarrow X_O^e$	93.6 +/- 0.5	93.6 +/- 0.3	79.4 +/- 0.8	No (Thm.1 & Cllry.3)	14.2	88.9
Model Base: MNIST_CNN [14]	DS 3 $X_N^e \not\leftarrow C^e \rightarrow X_O^e$	95.0 +/- 0.2	92.6 +/- 0.2	80.6 +/- 0.3	No (Cllry.3)	14.4	89.4
	DS 4 $X_N^e \leftarrow C^e \not\rightarrow X_O^e$	92.6 +/- 0.3	93.9 +/- 0.2	91.5 +/- 0.6	Yes (Cllry.4)	2.4	92.7
Data Base: CIFAR10	DS 1 $\not\rightarrow X_O \not\rightarrow$	96.1 +/- 0.4	96.1 +/- 0.4	96.4 +/- 0.2	N.A. (Baseline)	0.3	96.2
	DS 2 $X_N^e \leftarrow C^e \rightarrow X_O^e$	94.2 +/- 0.2	92.9 +/- 0.3	79.6 +/- 0.5	No (Thm.1 & Cllry.3)	14.6	88.9
Model Base: Wide_ResNet [33]	DS 3 $X_N^e \not\leftarrow C^e \rightarrow X_O^e$	94.8 +/- 0.3	92.2 +/- 0.8	79.2 +/- 0.4	No (Cllry.3)	15.6	88.7
	DS 4 $X_N^e \leftarrow C^e \not\rightarrow X_O^e$	94.9 +/- 0.2	94.2 +/- 0.4	94.1 +/- 0.4	Yes (Cllry.4)	0.9	94.4

(4) The observational data X^e is the composition of X_O^e and X_N^e, i.e., $X^e=\left(X_O^e, X_N^e\right)$. For simulation simplicity, we let $X^e = X_O^e + X_N^e$.

B. Simulated Visualization Experiments

The neural network model applied is the variational autoencoder-classifier. The structure of this model is detailed in TABLE IV. The model has three losses, the reconstruction loss, the KL divergence, and the classification loss (cross-entropy), which are assigned with weights 10, 1, and 1, respectively. For the MNIST data, the training step is set to 3000. As for other parameters, they are set to just the default values in TABLE III.

C. Simulated Numerical Experiments

We follow the same experiment procedure and the similar random search scheme of the DomainBed [14] to train and select the models (the grid search settings used are described in TABLE III). Regarding the variational autoencoder classifier, the weights of the reconstruction loss, the KL divergence, and the classification loss are selected as 10^{n1}, 10^{n2}, and 10^{n3}, respectively, where $n1 \sim U(0,2)$, $n2 \sim U(0,2)$, and $n3 \sim U(0,3)$ describe values randomly generated from three uniform distributions of respective ranges.

The results of the simulated domain generalization experiment based on the training domain validation are presented in TABLE V. Essentially, the results are in line with the results based on the test domain validation presented in TABLE I of the paper.

D. Experiments Based on Real-World Datasets

The experiment procedure also follows the DomainBed experiment principle [14]. The setting of the grid search applied are the same as those described in TABLE III except for the gird of the batch size, which is set to {8, 16, 32, 64} to obtain a good training efficiency.

Regarding the PU bearing fault dataset [34], at each speed, there are three kinds of data, the normal, the inner race fault, and the outer race fault. Here, we employ data from two speed conditions, 900 rpm and 1500 rpm. In the time domain input setting and the frequency domain input setting, TS 1 is training the model on data of 900 rpm and testing the model on data of 1500 rpm, namely, generalizing the model from 900 rpm to 1500 rpm, while TS 2 describes the reverse direction. The baseline training and test data are created from two independent trials of data at 900 rpm.

Regarding the the EPdeM belt fault dataset [35], at each speed, there are two types of data, the normal and the belt fault. TS 1 is to generalize the model from the data of the lower speed group (speed: 400-900 rpm) to the data of the higher speed group (speed: 1500-2000 rpm). TS 2 is in the opposite direction. The baseline is built by collecting the data of all speeds, randomizing speeds, and splitting data into two datasets.

Regarding the BU human activity dataset [36], the data is collected using the mobile phone vibration sensor when different ages of people (both male and female) are sitting and walking. We know that older people are often less active in movement and have relatively lower speeds than younger people. Thus, the age dependent domain data can be treated as speed dependent domain data. We then divide the data into the younger group (around 20 years old) and older group (around 50 years old) to create two domains of data. TS 1 is generalizing the model from the young group to the old group, while TS 2 considers the reverse direction. Baseline datasets are built by randomizing the ages and then splitting the data into two sets.

Regarding the JUST lung dataset [37], we also segment the data into the younger group (mostly round 30 years old) and the older group (mostly around 60 years old) to obtain two domains of data. TS 1 and TS 2 are similarly defined as the above human activity dataset. However, the change of speed (in terms of frequency shift) in the lung sound is not significant between the young and the old [38]. Baseline datasets are established by randomizing the ages and splitting the data into two sets.