Appendix

This appendix is organized as follows:

- For preliminaries on structural causal model and do-calculus, we refer readers to Section 2 of [10].
- Section A.1 gives the proofs and derivations for Section 3, where we first prove the Counterfactual Faithfulness theorem in Section A.1.1, and then prove the sufficient condition used to establish the correspondence between DCMs and disentangled generative causal factors in Section A.1.2.
- Section A.2 the proofs and derivations for Section 4, where we prove the Proxy Function theorem and its corollary in Section A.2.1, and then derive Eq. (7) in Section A.2.2.
- Section A.3 provides implementation details. Specifically, in Section A.3.1, we provide the network architecture of DCMs, the implementation of CycleGAN loss $\mathcal{L}^i_{CycleGAN}$ and DCM training details. In Section A.3.2, we show the network architectures of our backbone, the VAE and discriminators, together with their training details. In Section A.3.3, we attend to some details in the experiment.
- Section A.4 shows additional generated images from our DCMs and additional CAM results.

A.1. Proof and Derivation for Section 3

In this section, we will first derive the *Counterfactual Faithfulness theorem*. Then we will prove the sufficient condition in Section 3.1.

A.1.1. Counterfactual Faithfulness Theorem

We will first provide a brief introduction to the concept of counterfactual and disentanglement. Causality allows to compute how an outcome would have changed, had some variables taken different values, referred to as a counterfactual. In Section 3.1, we refer to each DCM (M_i, M_i^{-1}) in $\{(M_i, M_i^{-1})\}_{i=1}^k$ as a counterfactual mapping, where each M_i (or M_i^{-1}) essentially follows the three steps of computing counterfactuals [12] (conceptually): given a sample $X = \mathbf{x}$, 1) In abduction, $(U_1 = u_1, \dots, U_i = u_i, \dots, U_k = u_k)$ is inferred from $\mathbf{x}$ through $P(U|X)$; 2) In action, the attribute U_i is intervened by setting it to u_i' drawn from $P(U_i|S = t)$ (or $P(U_i|S = s)$), while the values of other attributes are fixed; 3) In prediction, the modified $(U_1 = u_1, \dots, U_i = u_i', \dots, U_k = u_k)$ is fed to the generative process $P(X|U)$ to obtain the output of the DCM $M_i(\mathbf{x})$ (or $M_i^{-1}(\mathbf{x})$). More details regarding counterfactual can be found in [11].

Our definition of disentanglement is based on [5] of group theory. Let $\mathcal{U}$ be a set of (unknown) generative factors, *e.g.*, such as shape and background. There is a set of *independent causal mechanisms* $\varphi : \mathcal{U} \to \mathcal{X}$, generating images from $\mathcal{U}$. Let $\mathcal{G}$ be the group acting on $\mathcal{U}$, *i.e.*, $g \circ u$ transforms $u \in \mathcal{U}$ using $g \in \mathcal{G}$ (*e.g.*, changing background "cluttered" to "pure"). When there exists a direct product decomposition $\mathcal{G} = \prod_{i=1}^k \mathcal{G}_i$ and $\mathcal{U} = \prod_{i=1}^k \mathcal{U}_i$ such that $\mathcal{G}_i$ acts on $\mathcal{U}_i$, we say that each $\mathcal{U}_i$ is the space of a disentangled factor. The causal mechanism (M_i, M_i^{-1}) is disentangled when its transformation in $\mathcal{X}$ corresponds to the action of $\mathcal{G}_i$ on $\mathcal{U}_i$.

We use $\mathcal{U}$ and $\mathcal{X}$ to denote the vector space of U and X respectively. We denote the generative process $P(X|U)$ $(U \to X)$ as a function $g : \mathcal{U} \to \mathcal{X}$. Note that we consider the function g as an embedded function [2], *i.e.*, a continuous injective function with continuous inversion, which generally holds for convolution-based networks as shown in [14]. Without loss of generality, we will consider the $S = s \to S = t$ mapping M_i for the analysis below, which can be easily extended to M_i^{-1}. Our definition of disentangled intervention follows the intrinsic disentanglement definition in [2], given by:

Definition (Disentangled Intervention). *A counterfactual mapping $M : \mathcal{X} \to \mathcal{X}$ is a disentangled intervention with respect to U_i, if there exists a transformation $M' : \mathcal{U} \to \mathcal{U}$ affecting only U_i, such that for any $u \in \mathcal{U}$,*

$$M(g(u)) = g(M'(u)). \tag{1}$$

Then we have the following theorem:

Theorem (Counterfactual Faithfulness Theorem). *The counterfactual mapping $M_i(X)$ is faithful if and only if M_i is a disentangled intervention with respect to U_i.*

Note that by definition, if $M_i(X)$ is faithful, $M_i(X) \in \mathcal{X}$. To prove the above theorem, one conditional is trivial: if M_i is a disentangled intervention, it is by definition an endomorphism of $\mathcal{X}$ so the counterfactual mapping must be faithful. For the second conditional, let us assume a faithful counterfactual mapping $M_i(X)$. Given g is embedded, the counterfactual mapping can be decomposed as:

$$M_i(X) = g \circ M_i' \circ g^{-1}(X), \tag{2}$$

where $\circ$ denotes function composition and $M_i' : \mathcal{U} \to \mathcal{U}$ affecting only U_i. Now for any $u \in \mathcal{U}$, the quantity $M_i(g(u))$ can be similarly decomposed as:

$$M_i(g(u)) = g \circ M_i' \circ g^{-1} \circ g(u) = g \circ M_i'(u). \tag{3}$$

Since M_i' is a transformation in $\mathcal{U}$ that only affects U_i, we show that faithful counterfactual transformation $M_i(X)$ is a disentangled intervention with respect to U_i, hence completing the proof.

With this theory, faithfulness=disentangled intervention. In Section 3.1, we train M_i to ensure $M_i(\mathbf{x}_s) \sim P(X_t)$ (faithfulness) for every sample $\mathbf{x}_s$ in $S = s$, hence encouraging M_i to be a disentangled intervention. Note that the above analysis can easily generalize to M_i^{-1}.

A.1.2. Sufficient Condition

We will prove the following sufficient condition: *if* (M_i, M_i^{-1}) *intervenes* U_i*, the* i*-th mapping function outputs the counterfactual faithful generation,* i.e.*, the smallest* $\mathcal{L}^i_{CycleGAN}$.

Without loss of generality, we will prove for the $S = s \rightarrow S = t$ mapping M_i, which can be extended to M_i^{-1}. For a sample $\mathbf{x}_s$ in $S = s$, let $g^{-1}(\mathbf{x}_s) = \mathbf{u} = (u_1, \ldots, u_i, \ldots, u_k)$. We modify U_i by changing u_i to a value $\hat{u}_i$ drawn from $P(U_i|S = t)$. Denote the modified attribute as $\hat{\mathbf{u}} = (u_1, \ldots, \hat{u}_i, \ldots, u_k)$. Denote the sample with attribute $\hat{\mathbf{u}}$ as $\hat{\mathbf{x}}$. Given M_i intervenes U_i, $M_i(\mathbf{x}_s)$ corresponds to a counterfactual outcome when U_i is set to $\hat{u}_i$ through intervention (or U set as $\hat{\mathbf{u}}$). Now as $g^{-1}(\hat{\mathbf{x}}) = \hat{\mathbf{u}}$, using the *counterfactual consistency rule* [9], we have $M_i(\mathbf{x}_s) = \hat{\mathbf{x}}$. As $\hat{\mathbf{x}}$ is faithful with the Counterfactual Faithfulness theorem, we prove that $M_i(\mathbf{x}_s)$, *i.e.*, the output of the i-th mapping function, is also faithful,*i.e.*, the smallest $\mathcal{L}^i_{CycleGAN}$.

A.2. Proof and Derivation for Section 4

In this section, we will first derive the Proxy Function theorem and the domain-agnostic nature of the proxy function, and then derive Eq. (7) under our chosen function forms in Section 4.

A.2.1. Proxy Function Theorem

We will derive for the general case where $\tilde{X}$ is any continuous proxy. We will assume that the confounder U follows the completeness condition in [8], which accommodates most commonly-used parametric and semi-parametric models such as exponential families.

Given $h_y(X, \hat{X})$ solves Eq. (3), we have:

$$P(Y|Z, X, S) = \int_{-\infty}^{+\infty} h_y(X, \hat{X}) P(\hat{X}|Z, X, S) d\hat{X}$$
$$= \int_{-\infty}^{+\infty} h_y(X, \hat{X}) \left\{ \int_{-\infty}^{+\infty} P(\hat{X}|U) P(U|Z, X, S) dU \right\} d\hat{X}. \tag{4}$$

From the law of total probability, we have:

$$P(Y|Z, X, S) = \int_{-\infty}^{+\infty} P(Y|U, X) P(U|Z, X, S) dU. \tag{5}$$

With Eq. (4) and Eq. (5) and the completeness condition, we have:

$$P(Y|U, X) = \int_{-\infty}^{+\infty} h_y(X, \hat{X}) P(\hat{X}|U) d\hat{X}, \tag{6}$$

which proves the Proxy Function theorem.

From Eq. (6), we have:

$$\int_{-\infty}^{+\infty} h_y(X, \hat{X}, S = s) P(\hat{X}|U) d\hat{X}$$
$$= \int_{-\infty}^{+\infty} h_y(X, \hat{X}, S = t) P(\hat{X}|U) d\hat{X}. \tag{7}$$

Hence from the completeness condition, we have $h_y(X, \hat{X}, S = s) = h_y(X, \hat{X}, S = t)$ [8]. Hence we prove that $h_y(X, \hat{X})$ is domain-agnostic.

Note that in Section 4, our proxy $\hat{X}$ is a continuous random variable takes values from $\{M_i(\mathbf{x}_s)\}_{i=1}^k$ for sample $\mathbf{x}_s$ in $S = s$ or $\{M_i^{-1}(\mathbf{x}_t)\}_{i=1}^k$ for sample $\mathbf{x}_t$ in $S = t$. This is a special case of the analysis above with the probability mass of $\hat{X}$ centers around the set of its possible values.

A.2.2. Derivation of Eq. (7)

We derive Eq. (7) as a corollary to [8]. The goal is to solve for $h_y(X, \hat{X})$ under the function form in Eq. (6) from the formula below:

$$P(Y|Z, X, S = s) = \int_{-\infty}^{\infty} P(\hat{X}|Z, X) h_y(X, \hat{X}) d\hat{X}. \tag{8}$$

For simplicity, we define a standard multivariate Gaussian function $\phi(\cdot)$. If $A \sim \mathcal{N}(0, \mathbf{I})$ and $A \in \mathbb{R}^n$, we have:

$$P(A) = \phi(A) = \frac{1}{(2\pi)^{n/2}} exp(-\frac{1}{2} A^T A). \tag{9}$$

Our function form for $P(Y|Z, X, S = s)$ is given by $\mathcal{N}(\mathbf{b}_1 + \mathbf{W}_1 Z + \mathbf{W}_2 X, \mathbf{\Sigma}_1)$ and for $P(\hat{X}|Z, X, S = s)$ is given by $\mathcal{N}(\mathbf{b}_2 + \mathbf{W}_3 Z + \mathbf{W}_4 X, \mathbf{\Sigma}_2)$, where the variance terms are omitted in the main text for brevity, as the final results only depend on the means. Specifically, $\mathbf{\Sigma}_2 \in \mathbb{R}^{n \times n}$ is a symmetrical matrix with Eigen-decomposition given by $\mathbf{\Sigma}_2 = \boldsymbol{\mathcal{U}} \boldsymbol{\Lambda} \boldsymbol{\mathcal{U}}^T$, where $\boldsymbol{\mathcal{U}} \in \mathbb{R}^{n \times n}$ is a full-rank matrix containing the eigen-vectors and $\boldsymbol{\Lambda} \in \mathbb{R}^{n \times n}$ is a diagonal matrix with eigen-values. We define $\mathbf{B} = \boldsymbol{\mathcal{U}} \boldsymbol{\Lambda}^{\frac{1}{2}}$, and hence $\mathbf{\Sigma}_2 = \mathbf{B}\mathbf{B}^T$. We can rewrite $P(\hat{X}|Z, X, S = s)$ as:

$$P(\hat{X}|Z, X, S = s) = |\mathbf{B}\mathbf{B}^T|^{-\frac{1}{2}} \phi(\mathbf{B}^{-1}(\hat{X} - \hat{\boldsymbol{\mu}})), \tag{10}$$

where $\hat{\mu} = \mathbf{b}_2 + \mathbf{W}_3 Z + \mathbf{W}_4 X$. Define $Z' = \mathbf{B}^{-1} \hat{\boldsymbol{\mu}}$ and $\hat{X}' = \mathbf{B}^{-1} \hat{X}$. We define

$$t_y(Z', X) = P(Y|Z = \mathbf{W}_3^+ (\mathbf{B}Z' - \mathbf{W}_4 X - \mathbf{b}_1), X, S = s). \tag{11}$$

Now we can solve h_y from

$$t_y(Z', X) = \int_{-\infty}^{\infty} \phi(\hat{X}' - Z') h_y(X, \mathbf{B}\hat{X}') d\hat{X}'. \quad (12)$$

Specifically, let h_1 and h_2 represent the Fourier transform of ϕ and t_y, respectively:

$$\begin{aligned} h_1(\boldsymbol{\nu}) &= \int_{-\infty}^{\infty} exp(-i\boldsymbol{\nu}Z')\phi(Z')dZ' \\ &= \int_{-\infty}^{\infty} exp(-i\boldsymbol{\nu}Z)\phi(Z)dZ \end{aligned} \quad (13)$$

$$h_2(\boldsymbol{\nu}, X, Y) = \int_{-\infty}^{\infty} exp(-i\boldsymbol{\nu}Z')t_y(Z', X)dZ', \quad (14)$$

where $i = (-1)^{\frac{1}{2}}$ is the imaginary unity. Substituting Eq. (12) (13) into Eq. (14), we have:

$$h_2(\boldsymbol{\nu}, X, Y) = h_1(\boldsymbol{\nu}) \int_{-\infty}^{\infty} exp(-i\boldsymbol{\nu}\hat{X}')h_y(X, \mathbf{B}\hat{X}')d\hat{X}'. \quad (15)$$

Hence

$$\int_{-\infty}^{\infty} exp(-i\boldsymbol{\nu}\hat{X}')h_y(X, \mathbf{B}\hat{X}')d\hat{X}' = \frac{h_2(\boldsymbol{\nu}, X, Y)}{h_1(\boldsymbol{\nu})}. \quad (16)$$

By Fourier inversion, we have:

$$h_y(X, \mathbf{B}\hat{X}') = \frac{1}{2\pi} \int_{-\infty}^{\infty} exp(-i\boldsymbol{\nu}\hat{X}') \frac{h_2(\boldsymbol{\nu}, X, Y)}{h_1(\boldsymbol{\nu})} d\boldsymbol{\nu} \quad (17)$$

By substituting $\hat{X} = \mathbf{B}\hat{X}'$, we have

$$h_y(X, \hat{X}) = \frac{1}{2\pi} \int_{-\infty}^{\infty} exp(-i\boldsymbol{\nu}\mathbf{B}^{-1}\hat{X}) \frac{h_2(\boldsymbol{\nu}, X, Y)}{h_1(\boldsymbol{\nu})} d\boldsymbol{\nu} \quad (18)$$

Solving for Eq. (18) yields:

$$\begin{aligned} h_y(X, \hat{X}) = \mathbf{b}_1 - \mathbf{W}_1\mathbf{W}_3^+\mathbf{b}_2 + \mathbf{W}_1\mathbf{W}_3^+\hat{X} \\ + (\mathbf{W}_2 - \mathbf{W}_1\mathbf{W}_3^+\mathbf{W}_4)X, \end{aligned} \quad (19)$$

where scaling terms not related with $X, \hat{X}$ are dropped as they do not impact inference. This completes the derivation.

A.3. Implementation Details

In this section, we will first provide a system overview, followed by the implementation details for DCMs in Section 3 and implementation details for backbone, VAE and discriminators in Section 4. Then we will give a more detailed discussion on the experiment design.

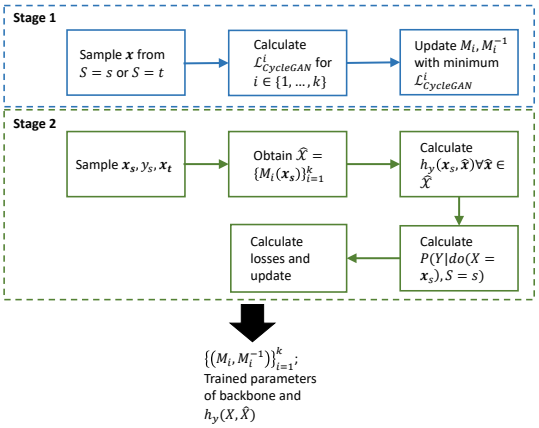

Figure A1: Overview of the two-stage training procedure.

A.3.1. Implementation Details for DCMs

System Overview. The two-stage training procedure is depicted in Figure A1. For inference, please refer to Algorithm **??**.

Network Architecture for DCMs. Let `Conv2D(n, c)` represent 2-d convolutional layer with $n \times n$ as kernel size and c output channels. For Each M_i, M_i^{-1}, the network architecture is given by `ReflectionPad(3) Conv2D(7,64) InstanceNorm ReLU Conv2D(3,128) InstanceNorm ReLU Conv2D(3,256) InstanceNorm ReLU ResNetBlock×2 ConvTranspose2D(3,128) InstanceNorm ReLU ConvTranspose2D(3,64) InstanceNorm ReLU ReflectionPad(3) Conv2D(7,3) Tanh`, where each `ResNetBlock` is implemented by `Conv2D(3,256) ReflectionPad(1) Conv2D(3,256)` with skip connection.

Network Architecture for CycleGAN Discriminators. To calculate each $\mathcal{L}^i_{CycleGAN}$, two discriminators taking image as input is required for $S = s$ and $S = t$, respectively. We implement each discriminator with `Conv2D(4,64) LeakyReLU(0.2) Conv2D(4,128) InstanceNorm LeakyReLU(0.2) Conv2D(4,256) InstanceNorm LeakyReLU(0.2) Conv2D(4,512) InstanceNorm LeakyReLU(0.2) Conv2D(4,1).`

CycleGAN Loss. Next, we will detail the implementation of $\mathcal{L}^i_{CycleGAN}$, which is given by:

$$\mathcal{L}^i_{CycleGAN} = \mathcal{L}^i_{adv} + \alpha_1\mathcal{L}^i_{cyc} + \alpha_2\mathcal{L}^i_{idt} \quad (20)$$

where α_1, α_2 are trade-off parameters. The adversarial loss $\mathcal{L}^i_{adv}$ requires the transformed images to look like real images in the opposing domain. For a sample $\mathbf{x}$, $\mathcal{L}^i_{real}$ is given

by:

$$\mathcal{L}_{adv}^{i} = \begin{cases} \log(1 - D_t'(M_i(\mathbf{x}))), & \text{if } \mathbf{x} \text{ from } S = s, \\ \log(1 - D_s'(M_i^{-1}(\mathbf{x}))), & \text{otherwise,} \end{cases} \tag{21}$$

where D_s', D_t' are discriminators that return a large value when its input looks like images in $S = s, S = t$, respectively. $\mathcal{L}_{cyc}^{i}$ is the cycle consistency loss [22] given by

$$\mathcal{L}_{cyc}^{i} = \begin{cases} \left\| M_i^{-1}(M_i(\mathbf{x})) - \mathbf{x} \right\|, & \text{if } \mathbf{x} \text{ from } S = s, \\ \left\| M_i(M_i^{-1}(\mathbf{x})) - \mathbf{x} \right\|, & \text{otherwise,} \end{cases} \tag{22}$$

where $\|\cdot\|$ is implemented using L1 norm. We follow CycleGAN [22] to use an identity loss $\mathcal{L}_{idt}^{i}$ to improve generation quality:

$$\mathcal{L}_{idt}^{i} = \begin{cases} \left\| M_i^{-1}(\mathbf{x}) - \mathbf{x} \right\|, & \text{if } \mathbf{x} \text{ from } S = s, \\ \left\| M_i(\mathbf{x}) - \mathbf{x} \right\|, & \text{otherwise.} \end{cases} \tag{23}$$

Discriminator Training. D_s', D_t' are trained to maximize the following loss:

$$\mathcal{L}_d = \begin{cases} \log D_s'(\mathbf{x}) + \frac{1}{k}\sum_{i=1}^{k} \mathcal{L}_{adv}^{i}, & \text{if } \mathbf{x} \text{ from } S = s, \\ \log D_t'(\mathbf{x}) + \frac{1}{k}\sum_{i=1}^{k} \mathcal{L}_{adv}^{i}, & \text{otherwise,} \end{cases} \tag{24}$$

which requires D_s', D_t' to recognize real samples and reject fake samples generated by *all* DCMs.

Training Details. The networks are randomly initialized by sampling from normal distribution with standard deviation of 0.02. We use Adam optimizer [6] with initial learning rate 0.0002, beta1 as 0.5 and beta2 as 0.999. We train the network with the initial learning rate for 100 epochs and decay the learning rate linearly for another 100 epochs on ImageCLEF-DA [1] and OfficeHome [18]. We train the network with the initial learning rate for 20 epochs and decay the learning rate linearly for another 20 epochs on VisDA-2017 [13]. The CycleGAN loss and discriminator loss are updated iteratively. The adopted competitive training scheme to train DCMs can sometimes be sensitive to network initialization. Hence all DCMs are first trained for 8000 iterations, before using the competitive training scheme.

A.3.2. Implementation Details for Section 4

Backbone. We adopt ResNet-50 [4] as our backbone, where the architecture is shown in Figure A2. Specifically, each convolutional layer is described as ($n \times n$ conv, p), where n is the kernel size and p is the number of output channels. Convolutional layers with $/2$ have a stride of 2 and are used to perform down-sampling. The solid curved lines represent skip connection. The batch normalization and ReLU layers are omitted in Figure A2 to highlight the key structure of the backbone. We fine-tuned the pre-trained backbone on ImageNet [16] for our experiments.

VAE. Denote `Linear(n,m)` as a linear layer with n input channels and m output channels. The encoder Q_θ is implemented with `Linear(2048,1200) ReLU Linear(1200,600) ReLU Linear(600,200)`, where the dimension of Z is 100, the first 100 dimensions of the output are the mean and the last 100 dimensions are the variance. The decoder P_θ is implemented with `Linear(100,600) ReLU Linear(600,2048) ReLU`.

Discriminators. The discriminator D_s and D_t are implemented with `Linear(2048,1024) ReLU Linear(1024,1024) ReLU Linear(1024,1).`

Training Details. The networks are randomly initialized with `kaiming` initialization with gain as 0.02. We employ mini-batch stochastic gradient descent (SGD) with momentum of 0.9 and nesterov enabled to train our model. We trained the networks for 10000 iterations on VisDA-2017 [13] and OfficeHome [18], and 3000 iterations on ImageCLEF-DA [1]. The linear functions $f_y, f_{\hat{x}}$, VAE, backbone and discriminators D_s, D_t are updated iteratively.

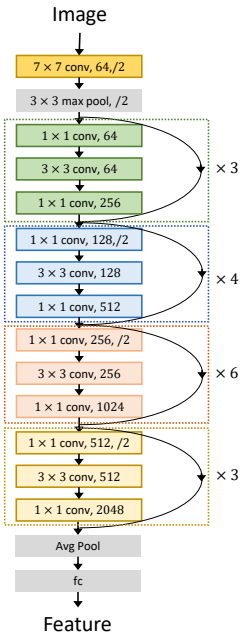

Figure A2: ResNet-50 architecture.

A.3.3. Discussion on Experiment

Choice of Dataset. While Office-31 [17] is also a popular dataset in UDA, some tasks in the dataset become too trivial, where many UDA algorithms achieve 100% (or almost) accuracy. Moreover, a recent study [15] reveals that the dataset is plagued with wrong labels, ambiguous ground-truth and data leakage, especially in the Amazon domain, which explains the low performance in task DSLR→Amazon and Webcam→Amazon. Hence we followed [20, 7, 19] and performed experiments on ImageCLEF-DA instead, which substitute Office-31 as a small-scale dataset with relatively small domain gap.

Detailed Discussion on t-SNE Plot. Note that the proxy loss in Section 4 aligns the proxy features with the sample features in the counterpart domain, which is different from

existing methods that try to align X_s and X_t directly. Our inference uses the proxy function implemented with Eq. (7), where in training, X takes values of X_s and $\hat{X}$ takes values of $M_i(X_s)$, in testing, X takes values of X_t and $\hat{X}$ takes values of $M_i^{-1}(X_t)$. While X_s and X_t (or $M_i(X_s)$ and $M_i^{-1}(X_t)$) are not aligned in TCM, we can still achieve competitive performance by finding how to transport with a causality theoretic viewpoint of the domain shift.

A.4. Additional Results

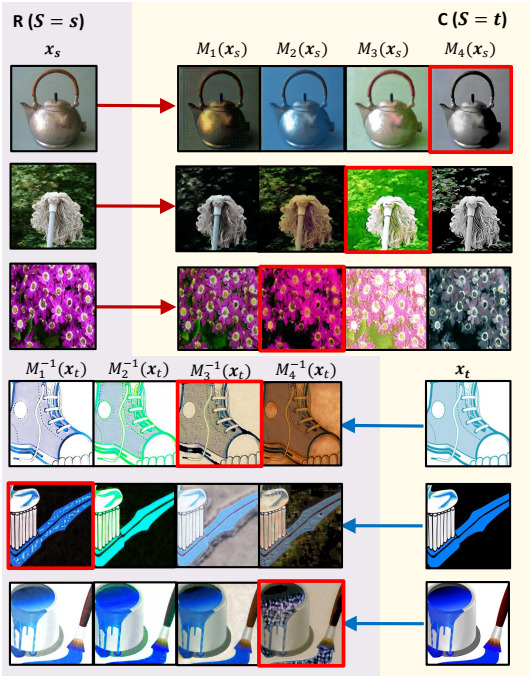

Figure A3: Supplementary to Figure 4.Transformation between "Real World" (R) and "Clipart" (C) domain with 4 trained DCMs $\{(M_i, M_i^{-1})\}_{i=1}^4$. The winning DCM is outlined in red.

Additional Visualizations of DCMs Outputs. In Figure A3, we show additional generated images from DCMs in the "Real World" (R)→"Clipart" (C) task of OfficeHome [18], which also has a large domain gap. $\{M_1, \ldots, M_4\}$ (or $\{M_1^{-1}, \ldots, M_4^{-1}\}$) roughly correspond to reducing (increasing) brightness, changing color, increasing (decreasing) saturation and removing (adding) color.

Additional CAM Responses. In Figure A4, we show the CAM responses in the "Real World"→"Clipart" task, where GVB-GD and baseline focuses on the background to distinguish "bed" and "bike" and the shape semantic is lost, leading to poor generalization in $S = t$. While TCM preserves the foreground object shape semantic. In Figure A5, all three methods fail on the two samples from $S = t$. On the "bed" sample, all methods indeed focus on the object.

Method	Backbone GFLOPS	Additional GFLOPS	#Parameters (M)
DANN	133	0.084	1.3
CDAN	133	1.158	18.1
GVB-GD	133	0.092	1.4
Baseline	133	0.072	1.1
TCM	133	0.364	10.1

Table A1: GFLOPS of feature extractor backbone and additional modules (*e.g.*, discriminator networks) as well as #parameters of those additional modules on Office-Home.

However, the object itself is not very discriminative, which may explain the failure. On the "bike" sample, the object is small and all methods fail to distinguish it from the context.

Convergence Speed, GFLOPS, #Parameters. The convergence speeds of TCM and related methods are shown in Figure A6. Our TCM (red) converges slightly slower in early iterations (*e.g.*, before 20 on ImageCLEF-DA and 250 on Office-Home) due to the training of VAE in Eq. 9 and the linear layers in Eq. 6. After that, TCM converges to the best performance. The GFLOPS and #Parameters are shown in Table A1. All methods use the same backbone and the backbone fine-tuning takes the major time cost (133), compared with the GFLOPS from additional modules. Regarding the parameters, our TCM has more (but not the most) because of the additional VAE and linear layers.

References

[1] Imageclef-da. http://imageclef.org/2014/adaptation, 2014. 4

[2] M. Besserve, A. Mehrjou, R. Sun, and B. Schölkopf. Counterfactuals uncover the modular structure of deep generative models. In *ICLR*, 2020. 1

[3] Shuhao Cui, Shuhui Wang, Junbao Zhuo, Chi Su, Qingming Huang, and Qi Tian. Gradually vanishing bridge for adversarial domain adaptation. In *CVPR*, 2020. 6

[4] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 4

[5] Irina Higgins, David Amos, David Pfau, Sebastien Racaniere, Loic Matthey, Danilo Rezende, and Alexander Lerchner. Towards a definition of disentangled representations. *arXiv preprint arXiv:1812.02230*, 2018. 1

[6] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 4

[7] Mingsheng Long, ZHANGJIE CAO, Jianmin Wang, and Michael I Jordan. Conditional adversarial domain adaptation. In *NeurIPS*, 2018. 4

[8] Wang Miao, Zhi Geng, and Eric J Tchetgen Tchetgen. Identifying causal effects with proxy variables of an unmeasured confounder. *Biometrika*, 2018. 2

[9] Judea Pearl. *Causality: Models, Reasoning and Inference*. Cambridge University Press, 2nd edition, 2009. 2

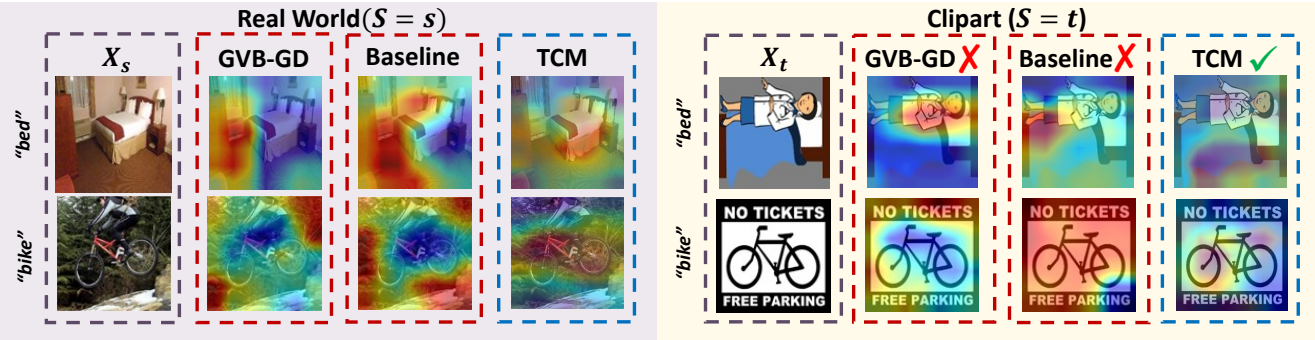

Figure A4: Class Activation Maps (CAMs) [21] of GVB-GD [3], Baseline and TCM on the Real World(S=s)→Clipart($S = t$) task in Office-Home dataset [18]. The left column shows two samples in each domain, whose class name is indicated on the top. In $S = t$, we show two samples predicted wrongly by GVB-GD and Baseline, but correctly with TCM. Supplementary to Figure 8

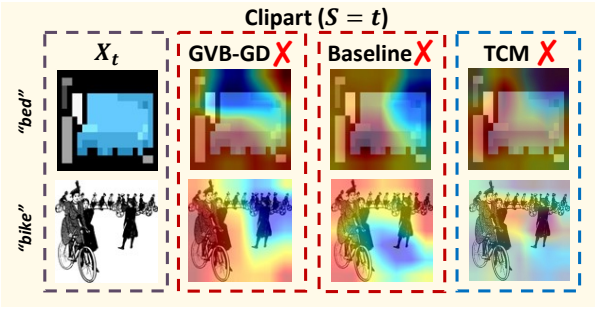

Figure A5: Class Activation Maps (CAMs) [21] of GVB-GD [3], Baseline and TCM on the Real World(S=s)→Clipart($S = t$) task in Office-Home dataset [18]. The left column shows two samples in each domain, whose class name is indicated on the left. All methods fail on the two samples. Supplementary to Figure 8.

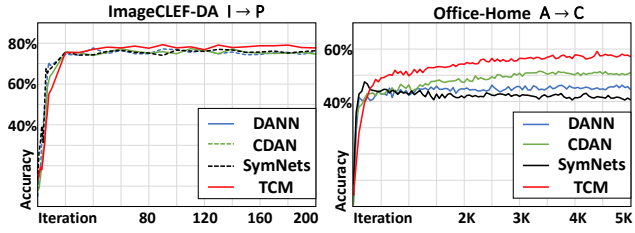

Figure A6: UDA accuracy (%) using DANN, CDAN, SymNets and our TCM, on different training iterations. Batch size is 32.

[10] Judea Pearl and Elias Bareinboim. External validity: From do-calculus to transportability across populations. *Statistical Science*, 2014. 1

[11] J. Pearl, M. Glymour, and N.P. Jewell. *Causal Inference in Statistics: A Primer*. Wiley, 2016. 1

[12] Judea Pearl and Dana Mackenzie. *The Book of Why: The New Science of Cause and Effect*. 2018. 1

[13] Xingchao Peng, Ben Usman, Neela Kaushik, Judy Hoffman, Dequan Wang, and Kate Saenko. Visda: The visual domain adaptation challenge. *arXiv preprint arXiv:1710.06924*, 2017. 4

[14] Michael Puthawala, Konik Kothari, Matti Lassas, Ivan Dokmanić, and Maarten de Hoop. Globally injective relu networks. *arXiv preprint arXiv:2006.08464*, 2020. 1

[15] Tobias Ringwald and Rainer Stiefelhagen. Adaptiope: A modern benchmark for unsupervised domain adaptation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2021. 4

[16] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 2015. 4

[17] Kate Saenko, Brian Kulis, Mario Fritz, and Trevor Darrell. Adapting visual category models to new domains. In *ECCV*, pages 213–226, 2010. 4

[18] Hemanth Venkateswara, Jose Eusebio, Shayok Chakraborty, and Sethuraman Panchanathan. Deep hashing network for unsupervised domain adaptation. In *CVPR*, 2017. 4, 5, 6

[19] Yabin Zhang, Bin Deng, Kui Jia, and Lei Zhang. Label propagation with augmented anchors: A simple semi-supervised learning baseline for unsupervised domain adaptation. In *ECCV*, 2020. 4

[20] Yabin Zhang, Hui Tang, Kui Jia, and Mingkui Tan. Domain-symmetric networks for adversarial domain adaptation. In *CVPR*, 2019. 4

[21] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *CVPR*, 2016. 6

[22] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *ICCV*, 2017. 4