Mixstyle-Entropy: Whole Process Domain Generalization with Causal Intervention and Perturbation (Supplementary Material)

Luyao Tang*[1]
http://lytang63.github.io
Yuxuan Yuan*[1]
yuanyuxuan0908@stu.xmu.edu.cn
Chaoqi Chen[2]
cqchen1994@gmail.com
Xinghao Ding[1]
dxh@xmu.edu.cn
Yue Huang†[1]
yhuang2010@xmu.edu.cn

[1] School of Informatics
Xiamen University
Xiamen, China

[2] Department of Computer Science
The University of Hong Kong
Hong Kong, China

A More Dataset and Implementation Details

A.1 Multi-domain Classification

Datasets. We fully verify the generalization performance of `Mixstyle-Entropy` on three standard DG benchmarks: PACS [11], VLCS [7], and Office-Home [25] (1) PACS is the most-widely used DG benchmark, exhibits significant distribution discrepancies across different domains, which contains 9,991 images of 7 classes from four kinds of domain: Photo, Art Painting, Cartoon, and Sketch. (2) VLCS contains 10,729 images of 5 classes from four photographic domains: VOC 2007 [6], LabelMe [24], Caltech [8], and Sun [26]. (3) Office-Home is composed of 15,500 images of 65 classes from four domains: Artistic, Clipart, Product, and Real World. The images are all collected from office and home environments. Each domain represents a different visual environment and presents distinct challenges, such as variations in lighting conditions, backgrounds, and object appearances. Moreover, the Office-Home dataset is known for its large number of categories, which further increases the complexity of the domain generalization task.

Implementation. For multi-domain classification, We implement our experiments mainly based on the opensource toolboxes, i.e., Dassl.pytorch [29], including data preparation, model training, and model selection. Specifically, we select ResNet18 and ResNet50 [9] pre-trained on the ImageNet[5] as our backbones. All images are resized to 224 × 224. The basic data augmentation consists of random horizontal flip and translation. During training, the batch size is fixed at 64. The networks are trained using SGD with a momentum of 0.9 and weight decay of 5e-4 for 100 epochs. The initial learning rate is set to 0.002 and decayed by the cosine annealing rule. For EnIn, the cropped area ratio of $\mathcal{M}$ is empirically set to $\frac{1}{4}$. We insert the causal intervention module after the shallower layers, specifically block-1 and block-2, as these layers contain a more abundant wealth of stylistic information. For HoPer, we ensure consistent batch sizes during testing and training. Further, the prototype classifier's resource allocation is controlled using the entropy filter threshold β. We set β to four times the testing batch size to form a stable generalized classifier early in the testing phase.

A.2 Semantic Segmentation

Datasets. GTA5 [21] is a synthetic dataset generated from Grand Theft Auto 5 game engine, which includes 24,966 high-resolution synthetic game screenshots. Cityscapes [4] is a real-world dataset collected from different cities in primarily Germany, which includes street images from 50 German cities at different times, weather, and seasons.

Implementation. Consistent with prior cross-domain semantic segmentation approaches [12], we employ the DeepLab-v2 [3] segmentation network with a ResNet-101 backbone. We use SGD optimizer with an initial learning rate of 5×10^{-4}, momentum of 0.9, and weight decay of 10^{-4}. Mean Intersection over Union (mIoU) and mean Accuracy (mAcc) of all object categories are used for evaluation. We integrate the EnIn component after the backbone's block-1, block-2 and conducted ablation experiments to provide insights into the insertion position.

A.3 Instance Retrieval

Datasets. Our experiment uses two commonly used Re-ID datasets: Market1501 [27] and DukeMTMC [22]. The pedestrian images in Market1501 dataset The Market1501 dataset contains pedestrian images from six campus cameras, annotated for a total of 1,501 pedestrians, among which 751 pedestrians are part of the training set while 750 are part of the testing set. There are no overlapping pedestrian IDs between the training and testing sets, which means that the 751 pedestrians in the training set do not appear in the testing set. The DukeMTMC dataset contains a total of 36,411 images of 1,812 pedestrians. Among them, 1,404 pedestrians are captured by more than two cameras, while 408 pedestrians are only captured by a single camera. Since person re-identification is essentially a cross-camera search task, these 408 pedestrians cannot be used for person re-identification and are included in the dataset as distractors.

Implementation. To evaluate the model's generalizability, we take one dataset as training and test the performance on the other domain. Meanwhile, We evaluate the performance using mean average precision (mAP) and ranking accuracy metrics. Following the previous cross-domain Re-ID works [13, 29], we employ the Adam optimizer with an initial learning rate of 3.5×10^{-4} and train for 50 epochs on ResNet-50. During training, the batch size is fixed at 64 and the EnIn module is inserted after block-1, block-2, and block-3.

B More experiments

B.1 Multi-domain Classification

Results on VLCS. We conduct a consistent reproduction of the current mainstream augmentation-based methods on ResNet-18 and ResNet-50, and compare them with `Mixstyle-Entropy`.

The results indicate that the previous methods, in the case of challenging datasets like VLCS, fail to achieve improvements in average accuracy and may exhibit performance degradation in certain domains. In contrast, by successfully decoupling domain-related variables, we achieve overall performance improvement. Detailed experimental results are reported in Table 1 and Table 2.

Method	VOC	LabelMe	Caltech	Sun	Avg (%)
ERM	73.7 ± 0.4	66.4 ± 0.5	91.2 ± 0.3	70.3 ± 0.5	75.4 ± 0.3
pAdaIN [17]	73.3 ± 0.3	66.2 ± 0.5	91.7 ± 0.6	69.3 ± 0.2	75.1 ± 0.4
Mixstyle [29]	73.1 ± 0.3	66.3 ± 0.3	91.7 ± 0.5	70.0 ± 0.4	75.3 ± 0.3
DSU [13]	74.1 ± 0.5	66.4 ± 0.6	90.8 ± 0.4	70.9 ± 0.4	75.5 ± 0.5
Ours	**77.0** ± 0.6	**68.8** ± 0.4	**91.9** ± 0.4	**73.7** ± 0.5	**77.8** ± 0.4

Table 1: Generalization results (%) of VLCS benchmark on ResNet-18.

Method	VOC	LabelMe	Caltech	Sun	Avg (%)
ERM	77.1 ± 0.4	67.2 ± 0.4	92.2 ± 0.3	73.9 ± 0.4	77.6 ± 0.2
pAdaIN [17]	78.0 ± 0.5	67.6 ± 0.4	91.7 ± 0.6	72.5 ± 0.3	77.5 ± 0.4
Mixstyle [29]	78.3 ± 0.5	68.3 ± 0.3	91.5 ± 0.5	72.0 ± 0.4	77.5 ± 0.4
DSU [13]	77.3 ± 0.6	66.8 ± 0.5	92.4 ± 0.4	74.6 ± 0.3	77.8 ± 0.5
Ours	**79.9** ± 0.3	**70.0** ± 0.4	**93.0** ± 0.5	**75.4** ± 0.4	**79.6** ± 0.4

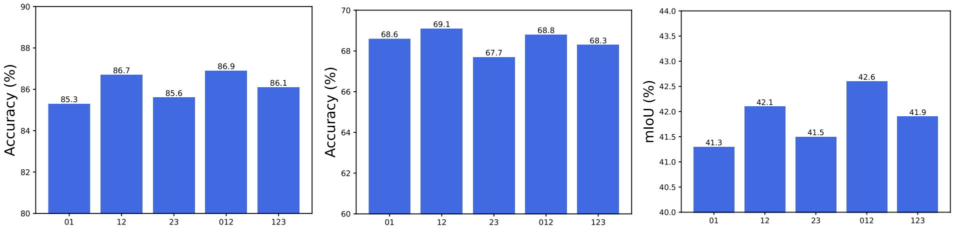
Table 2: Generalization results (%) of VLCS benchmark on ResNet-50.

Results on Camelyon17. Considering multiple factors such as imaging devices, medical image analysis is highly susceptible to domain shift, with protocols causing significant domain transfer. However, due to the complex and challenging data distribution, there is a lack of reported experiments on DG for medical images. We validate the performance of our model on the challenging Camelyon17 [1], which comprises images from five medical centers. This dataset consists of pathological images as input and labels indicating whether the central region contains any tumor tissue. Given the lack of reported performance in the current literature, we conduct experiments from scratch using the WILDS [10] and directly use the official implementation of each method without any modifications. Table 3 provides evidence of the effectiveness of our model. Compared to the baseline or other methods, `Mixstyle-Entropy` achieves impressive improvements. This suggests that by causality modeling, even with highly challenging medical data, `Mixstyle-Entropy` can facilitate the induction of more generalizable models.

Method	H1	H2	H3	H4	H5	Avg (%)
ERM	95.3 ± 0.4	91.4 ± 0.3	89.5 ± 0.3	96.2 ± 0.2	94.6 ± 0.4	93.4 ± 0.4
Mixstyle [29]	96.0 ± 0.2	91.2 ± 0.3	93.1 ± 0.4	94.9 ± 0.3	92.9 ± 0.4	93.6 ± 0.4
pAdaIN [17]	96.3 ± 0.3	93.1 ± 0.4	94 .7 ± 0.5	95.1 ± 0.5	94.1 ± 0.4	94.7 ± 0.5
DSU [13]	96.6 ± 0.7	93.1 ± 0.6	91.7 ± 0.7	96.2 ± 0.4	94.1 ± 0.7	94.3 ± 0.7
Ours	**96.8** ± 0.4	**94.2** ± 0.5	**94.9** ± 0.7	**96.9** ± 0.4	**95.7** ± 0.5	**95.7** ± 0.6

Table 3: Rresults on Camelyon17. H1-H5 represents different hospitals.

Effect of insert position. We conduct further exploration regarding the insertion position of the EnIn module. Due to the dependence on slicing the feature map, the selection of $\mathcal{M}$ requires its insertion into a shallower layer of the network. We arrange different combinations of the number and position of insertion layers, and the results for the classification task are depicted in Figure 1(a) and Figure 1(b). Taking ResNet as an example, 0 represents insertion after the original image, while 1, 2, and 3 correspond to insertion after block-1, block-2, and block-3, respectively. It can be observed that blocks-1 and blocks-2 in the network stack exhibit better overall performance in the presence of causal interventions. This is attributed to the fact that deeper-level features contain less domain-related information, which hinders feature decoupling.



((a)) PACS ((b)) Office-Home ((c)) GTA5

Figure 1: The effect of EnIn insertion location on model generalization performance.

Effect of blending intensity. In Mixstyle-Entropy, only a few hyperparameters require adjustment. For the mixture ratio of instance-wise mean and variance, we sample from the beta distribution. As illustrated in Table 4, we conduct experiments on ResNet-18 and ResNet-50 to explore the blending intensity of instance characteristic statistics. Overall, an appropriate value for this parameter is 0.1, and it remains robust across networks of different depths and datasets.

(a) Performance of ResNet-18

α	PACS	Office-Home	Avg
0.05	86.5 ± 0.3	69.1 ± 0.2	77.8 ± 0.2
0.1	86.7 ± 0.4	69.1 ± 0.3	77.9 ± 0.3
0.2	86.4 ± 0.3	68.9 ± 0.4	77.7 ± 0.3
0.3	86.2 ± 0.2	68.4 ± 0.3	77.3 ± 0.4
0.4	85.7 ± 0.4	68.6 ± 0.2	77.2 ± 0.3
0.5	85.6 ± 0.3	68.2 ± 0.4	76.9 ± 0.2

(b) Performance of ResNet-50

α	PACS	Office-Home	Avg
0.05	89.7 ± 0.3	74.0 ± 0.4	81.9 ± 0.4
0.1	89.9 ± 0.2	74.4 ± 0.4	82.2 ± 0.3
0.2	90.1 ± 0.3	74.3 ± 0.3	82.2 ± 0.2
0.3	89.5 ± 0.3	73.8 ± 0.2	81.7 ± 0.3
0.4	89.4 ± 0.4	73.5 ± 0.3	81.5 ± 0.3
0.5	89.1 ± 0.3	73.5 ± 0.4	81.3 ± 0.4

Table 4: The effect of blending intensity in EnIn on classification.

Effect of prototype classifier size. During the testing phase, after applying the HomeoScore filter, we are able to obtain feature representations that are closer to the centroids of the classes. These representations are used to construct the prototype classifier, where the samples are selected based on lower entropy values. We conduct experiments on the robustness of the prototype classifier's size using multiples β of the testing batch size, which is a more intuitive scale choice. The results, as shown in Table 5, indicate that smaller classifier sizes are not conducive to finding the optimal prototypes. When $\beta \geq 2$, the model's generalization performance improves significantly, and this holds true across different network depths and datasets. Therefore, we can make a sensible choice of classifier size based on the specific deployment environment.

(a) Performance of ResNet-18

β	PACS	Office-Home	Avg
1	87.7 ± 0.4	69.9 ± 0.3	78.8 ± 0.3
2	88.5 ± 0.5	70.5 ± 0.3	79.5 ± 0.2
3	89.0 ± 0.3	70.5 ± 0.2	79.8 ± 0.2
4	88.6 ± 0.4	70.4 ± 0.4	79.5 ± 0.3
5	88.5 ± 0.2	70.0 ± 0.3	79.3 ± 0.4

(b) Performance of ResNet-50

β	PACS	Office-Home	Avg
1	89.9 ± 0.5	75.5 ± 0.3	82.7 ± 0.4
2	90.4 ± 0.3	75.6 ± 0.5	83.0 ± 0.4
3	90.9 ± 0.4	75.7 ± 0.4	83.3 ± 0.2
4	91.0 ± 0.4	75.7 ± 0.6	83.4 ± 0.5
5	90.7 ± 0.3	75.9 ± 0.4	83.3 ± 0.3

Table 5: The effect of prototype classifier size in HoPer on classification.

Visualization. In the EnIn module, we propose the computation of feature entropy, which is different from similar approaches like CAM [28] that directly utilize the label to visualize feature activations. The key distinction lies in our avoidance of direct label utilization, as it often introduces correlations between objects and domains [15]. As shown in Figure 2, we visualize the early-stage feature maps during the network training. The first row displays the original images, the second row depicts CAM (with bright areas indicating higher values), and the third row showcases the feature entropy within the EnIn process (with bright areas representing lower values). Through visual observation, it becomes apparent that direct modeling of the label fails to harness the generalization capabilities of pretrained weights on ImageNet [6], leading to overconfidence or erroneous focus. Conversely, by considering entropy, we gain a clear understanding of the overall attention of the network towards features, rather than specific classes. Interestingly, regions with high entropy tend to concentrate in the background, which aligns with the notion of domain-related information as suggested by previous works [2, 16].

B.2 Semantic Segmentation

Effect of insert position. In the context of semantic segmentation, we also explore the placement of the EnIn module, and the findings are similar to those in the classification task. Specifically, when inserted after block-1 and block-2, the model's generalization ability significantly improves. Additionally, since segmentation involves dense pixel-level predictions, applying the EnIn operation after the original image can be seen as providing additional stylized augmentation, further enhancing the performance. The specific mIoU is presented in Figure 1(c).

Effect of blending intensity. As shown in Table 6, similar to the multi-domain classification task, the model exhibits better generalization performance when α takes on a smaller value. We fix it at 0.1 across all tasks and datasets.

α	0.05	0.1	0.2	0.3	0.4	0.5
mIoU (%)	42.5 ± 0.4	42.6 ± 0.5	42.4 ± 0.3	41.9 ± 0.4	41.7 ± 0.3	41.6 ± 0.5
mAcc (%)	53.3 ± 0.2	53.7 ± 0.3	53.8 ± 0.3	53.0 ± 0.2	52.7 ± 0.4	52.3 ± 0.3

Table 6: The effect of blending intensity in EnIn on segmentation.

Visualization. The visualization of semantic segmentation is presented in Figure 3, where we compare it with MxiStyle and DSU. It is evident that previous approaches struggle to accurately segment regions with similar features, such as shrubs and lawns, sidewalks and roads. However, EnIn successfully decouples domain-related and class-related information

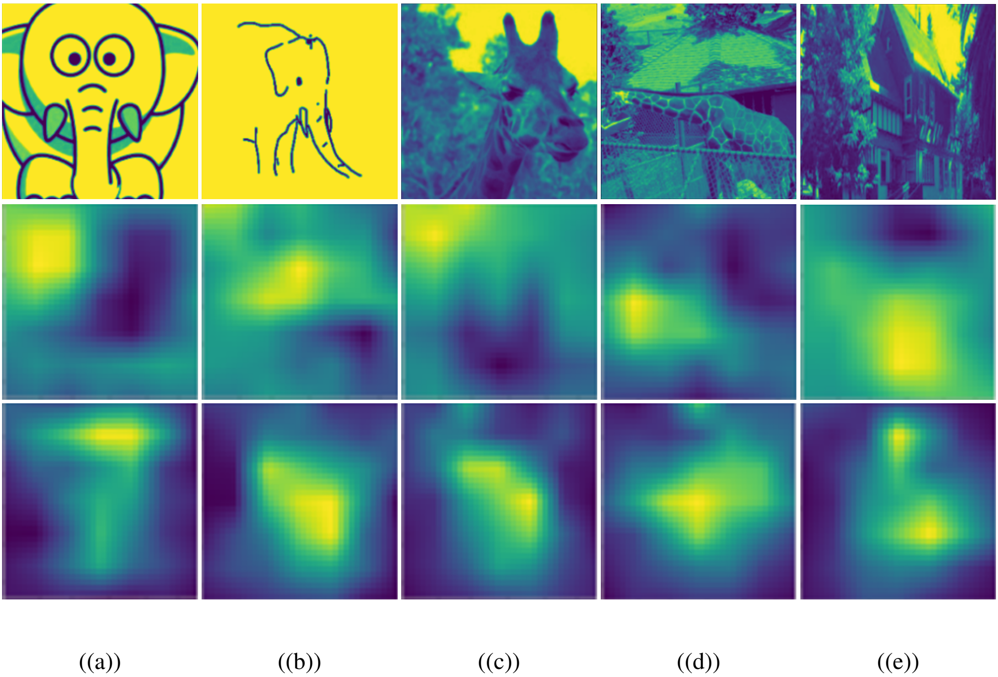


Figure 2: The visualization between Visible Entropy and CAM.

at the embedding level using feature entropy. This enables the extraction of causal variables for predicting the correct category, resulting in more precise segmentation.

C Theoretical Insights

C.1 Additional Definitions

Normalizing features with instance-specific mean and standard deviation has been found effective for removing image style. Given batch level feature maps $x \in \mathbb{R}^{B\times C\times H\times W}$ of the network, with B, C, H and W denoting the dimension of batch, channel, height and width, respectively. We can formulate the instance-specific feature statistics mean $\mu \in \mathbb{R}^{B\times C}$ as:

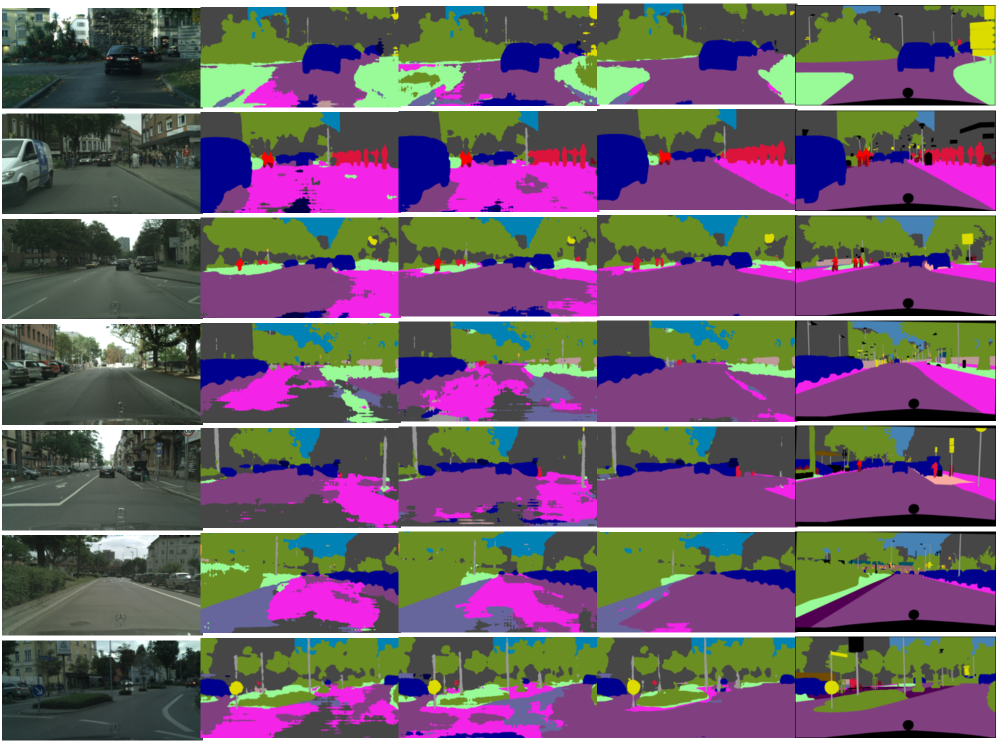
$$\mu(x) = \frac{1}{HW}\sum_{h=1}^{H}\sum_{w=1}^{W} x_{b,c,h,w}, \tag{1}$$

and standard deviation $\sigma \in \mathbb{R}^{B\times C}$ of each instance in a mini-batch can be formulated as:

$$\sigma(x) = \sqrt{\frac{1}{HW}\sum^{H}\sum^{W}\left(x_{b,c,h,w} - \mu(x)\right)^2}. \tag{2}$$

The network receives randomly sampled input x and generates the corresponding feature $g(x)$. When the domain label is unavailable, $\tilde{g}(x)$ is created by shuffling the order of samples in the current batch, forming a new reference batch. This operation, referred to as shuffle, is represented as follows:

$$\tilde{g}(x) = \text{shuffle}(g(x)). \tag{3}$$



(a) Unseen domain (b) Mixstyle (c) DSU (d) EnIn (Ours) (e) Ground truth

Figure 3: Visualization of segmentation results for the task GTA5 → Cityscapes.

Algorithm 1 The algorithm of the training phase: EnIn

```
Input: Intermediate feature in a mini-batch $g(\mathrm{x}) \in \mathbb{R}^{B\times C\times H\times W}$, shuffled feature $\tilde{g}(\mathrm{x})$, local feature vector $\mathbf{v_i}$, classifier weight $\mathbf{W}$.
Output: Intermediate feature $\hat{g}(\mathrm{x}) \in \mathbb{R}^{B\times C\times H\times W}$ after EnIn.
Step1. Feature entropy extraction:
  Compute feature entropy mask $\mathcal{M}$.
  Compute final prediction scores $\mathbf{F}$ for $K$ classes:
  $\mathbf{F} = \frac{1}{hw}\sum_i \mathbf{W}^\top \mathbf{v}_i = \frac{1}{hw}\sum_i \hat{\mathbf{F}}_i$.
  Local class probability at location $i$:
  $\hat{\mathbf{p}}_i = \mathrm{softmax}(\hat{\mathbf{F}}_i)$.
  Compute the Shannon entropy:
  $H(\hat{\mathbf{p}}_i) = -\sum_{k=1}^{K} \hat{\mathbf{p}}_i(k) \log \hat{\mathbf{p}}_i(k)$.
  Generate the feature entropy mask, choose the region with maximum and minimum entropy:
  $\mathcal{M} = \mathrm{Normalize}(H(\hat{\mathbf{p}}_i))$,
  $\mathcal{M}_{crop}^{max} = \max \mathcal{M}_{crop}$ for all $\mathcal{M}_{crop} \in \mathcal{M}$ ,
  $\mathcal{M}_{crop}^{min} = \min \mathcal{M}_{crop}$ for all $\mathcal{M}_{crop} \in \mathcal{M}$.
Step2. Feature causal intervention:
  Sample $p \sim U(0,1)$.
  if $p < 0.5$ and Training then
    Compute the channel-wise mean and standard deviation of each instance in a mini-batch:
    $\mu(g(\mathrm{x})) = \frac{1}{HW}\sum_{h=1}^{H}\sum_{w=1}^{W} g(\mathrm{x})$,
    $\sigma(g(\mathrm{x})) = \sqrt{\frac{1}{HW}\sum_{h=1}^{H}\sum_{w=1}^{W}(g(\mathrm{x}) - \mu(g(\mathrm{x})))^2}$.
    Calculate mixed feature statistics for the region with maximum and minimum entropy:
    $\tilde{\gamma}_{mix} = \lambda\sigma(g(\mathrm{x})) + (1-\lambda)\sigma(\tilde{\mathcal{M}}_{crop}^{max} \odot \tilde{g}(\mathrm{x}))$  $\tilde{\beta}_{mix} = \lambda\mu(g(\mathrm{x})) + (1-\lambda)\mu(\tilde{\mathcal{M}}_{crop}^{max} \odot \tilde{g}(\mathrm{x}))$,
    $\gamma_{mix} = \lambda\sigma(g(\mathrm{x})) + (1-\lambda)\sigma(\mathcal{M}_{crop}^{min} \odot g(\mathrm{x}))$  $\beta_{mix} = \lambda\mu(g(\mathrm{x})) + (1-\lambda)\mu(\mathcal{M}_{crop}^{min} \odot g(\mathrm{x}))$,
    $\lambda \sim \mathrm{Beta}(\alpha, \alpha)$.
    Obtain the feature after EnIn:
    Case 1: $\tilde{g}(\mathrm{x}) = \mathcal{M} \odot g(\mathrm{x}) + \tilde{\gamma}_{\mathrm{mix}} \frac{g(\mathrm{x}) - \mu(g(\mathrm{x}))}{\sigma(g(\mathrm{x}))} + \tilde{\beta}_{\mathrm{mix}}$,
    Case 2: $\tilde{g}(\mathrm{x}) = \gamma_{\mathrm{mix}} \frac{g(\mathrm{x}) - \mu(g(\mathrm{x}))}{\sigma(g(\mathrm{x}))} + \beta_{\mathrm{mix}}$.
    return $\tilde{g}(\mathrm{x})$.
  else
    adopt the original feature $g(\mathrm{x})$ and skip this module.
```

Algorithm 2 The algorithm of the testing phase: HoPer

```
Input: Feature generator $g_\theta$, the batch of data $\mathrm{x_t}$, and memory bank $\mathbb{B}$ at test time $t$.
Output: Prediction for $\mathrm{x_t}$.
Obtain feature presentation $g_\theta(\mathrm{x_t})$, prediction score $p_t$ and corresponding pseudo-label $y_t$.
Perform the similar feature transformation as EnIn:
  $g'(\mathrm{x_t}) = \gamma_{\mathrm{mix}} \frac{g(\mathrm{x_t}) - \mu(g(\mathrm{x_t}))}{\sigma(g(\mathrm{x_t}))} + \beta_{\mathrm{mix}}$.
Obtain prediction score $p_t'$ and corresponding pseudo-label $y_t'$.
Calculate HomeoScore:
  $\mathrm{HomeoScore} = \left( \sum_{j=1}^{k} \left| p_t^j - p_t^{j\prime} \right|^2 \right)^{\frac{1}{2}}$.
Adjust memory bank:
  $\mathbb{B}_t^k = \mathbb{B}_{t-1}^k \cup \left\{ \frac{g'(\mathrm{x_{t-1}})}{\|g'(\mathrm{x_{t-1}})\|} \right\}$ for $y'_{(t-1)} = y^k$ and HomeoScore $< \alpha$,
  $\mathbb{B}_t^k = \left\{ g'(\mathrm{x}) \mid g'(\mathrm{x}) \in \mathbb{B}_t^k, H(p') \le \beta \right\}$.
Predict based on feature similarities to prototypes for class k:
  $y_j^k = \frac{\exp(sim(g_j(\mathrm{x_t}), c_k))}{\sum_{k'=1}^{|Y|} \exp(sim(g_j(\mathrm{x_t}), c_{k'}))}$ for all $x_t \in \mathrm{x_t}$.
return $y_j^k$.
```

C.2 Structural Causal Graph

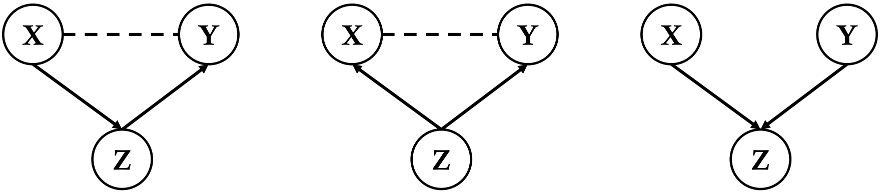
Three perspectives exist regarding the relationship between internal feature elements and class labels in the context of classification tasks. As discussed by previous works [14, 23], the first perspective suggests that the true class labels determine the features observed in the data. On the other hand, according to some analysis [19, 23], it is the features that lead to the labels. MatchDG [15] acknowledges that both mechanisms are possible, where the true class label Y_{true} determines these features, but it remains unobserved. The observed features are then utilized to assign a class label Y to each input.

In the context of supervised learning, we acknowledge that labels are manually annotated, and this process inevitably introduces biased information. Within a single image, there may exist multiple objects of different classes, yet only one class can be annotated. Furthermore, objects belonging to the same class may be assigned different class labels due to their diverse distributions. Since our network optimization relies on the class label Y, we construct a causal graph that explicitly includes Y and implicitly represents the correlation between O and D. We choose not to explicitly model the parent nodes of O and D, as the generative mechanisms of these two variables are not fixed and cannot be regarded as unrelated [2].

C.3 D-separation

Let X, Y, Z be the three non-intersecting subsets of nodes in a causal graph. Interpreting these three types of junctions in Figure 4 holds significant importance, as all Bayesian networks (or causal graphs) can be decomposed into combinations of different junction modules.

The first type of structural linkage, known as head-to-tail, as the name suggests, allows information to flow from one end to the other, resembling a chain. In the second type, called tail-to-tail or forked linkage, we can see that the two arrows resemble a fork. Most importantly, the third type, called reverse fork or head-to-head linkage, resembles two colliding asteroids, with information flowing into the middle node.



(a) chain (b) fork (c) inverted fork (collider)

Figure 4: D-separation

If two nodes are located at opposite ends of a pathway and there is information flow towards both nodes or from one node to the other, then these two nodes are considered to be interconnected [18]. In the context of chain, if information flows from node X through node Z to node Y, then nodes X and Y are considered to be interrelated (i.e., not independent). In the case of fork, if information flows from node Z to both node X and node Y, then nodes X and Y are not independent. In the case of colliding, if information flows from nodes X and Y to node Z, and a collision occurs at node Z, then node Z is also referred to as a collider. In this scenario, both X and Y influence Z, but there is no information flowing from Z to either X or Y, thus X and Y are mutually independent. If all paths from X to Y are blocked, then X is $d-separated$ from Y by Z: $\text{dsep}(X,Y,Z) \Rightarrow X \perp\!\!\!\perp Y|Z$.

C.4 Invariance Conditions

Like the SCM (a) depicted in Figure 1 in the main text, let X_O denote an unobserved higher-level feature that is exclusively relevant to object generation and forms an optimal classifier. We acknowledge that $X_O \perp\!\!\!\perp D|O$. For the necessary condition of domain-invariant representation, $g(\mathrm{x}) \perp\!\!\!\perp D$ is required. However, by employing the d-separation criterion on the SCM, we can infer $X_O \not\perp\!\!\!\perp D$. Similarly, for domain-invariant representation conditioned on the class, the required condition is $g(\mathrm{x}) \perp\!\!\!\perp D|Y$. Nonetheless, through d-separation, we discover that $X_O \not\perp\!\!\!\perp D|Y$, regardless of whether the relationship between O and D is explicitly modeled. Therefore, naturally, neither X_O nor any function of X_O is the optimal solution. To attain the optimal solution, additional assumptions such as an infinite number of samples are required, which are infeasible. Hence, we can solely rely on the causal relationships during the training process, leveraging feature representation entropy to extract and utilize inter-variable independence, thus approaching the performance upper bound. The algorithm of the proposed method is illustrated in Algorithm 1 and 2.

D Discussion

The potential of HomeoScore. In the main text, we mentioned the incorporation of samples with lower HomeoScore values into the construction of the prototype classifier. This approach, compared to the traditional method of using classification entropy for filtering, leverages the greater distribution variance of HomeoScore values to exclude erroneous pseudo-labels and prevent performance degradation. In fact, HomeoScore holds immense potential for broader applications, such as anomaly detection and open-set recognition. By enhancing the class-related variables in sample features, samples that deviate from the distribution typically lack stable causal features, resulting in greater prediction fluctuations. By comparing their HomeoScore values with those of normal samples within the distribution, it is possible to partially identify anomalous samples or filter out other classes not known during the testing process.

Future research directions. `Mixstyle-Entropy` has achieved impressive advancements across multiple tasks, datasets, and different models. Building upon this foundation, there are several promising directions for future integration and improvement. For instance, the implementation framework of decoupling feature entropy can be explored, extending its application to various types of networks such as ViT and MLP. Based on these theoretical underpinnings, such extensions are feasible. Furthermore, our proposed approach does not require extra training parameters. With the continuous development of TTA techniques and improving hardware performance, combining test-time training with the prototype classifier presents an intriguing direction. Leveraging more accurate pseudo-labels during testing to guide the model towards improved performance on the target domain warrants further discussion. Additionally, introducing multimodal information [20] into the process of causal intervention, blocking domain-related variables in another modality, is also worth exploring.

References

[1] Peter Bandi, Oscar Geessink, Quirine Manson, Marcory Van Dijk, Maschenka Balkenhol, Meyke Hermsen, Babak Ehteshami Bejnordi, Byungjae Lee, Kyunghyun Paeng, Aoxiao Zhong, et al. From detection of individual metastases to classification of lymph node status at the patient level: the camelyon17 challenge. *IEEE transactions on medical imaging*, 38(2):550–560, 2018.

[2] Chaoqi Chen, Luyao Tang, Feng Liu, Gangming Zhao, Yue Huang, and Yizhou Yu. Mix and reason: Reasoning over semantic topology with data mixing for domain generalization. In *Advances in Neural Information Processing Systems*.

[3] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence*, 40(4): 834–848, 2017.

[4] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3213–3223, 2016.

[5] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[6] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88: 303–338, 2010.

[7] Chen Fang, Ye Xu, and Daniel N Rockmore. Unbiased metric learning: On the utilization of multiple datasets and web images for softening bias. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1657–1664, 2013.

[8] Li Fei-Fei, Rob Fergus, and Pietro Perona. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. In *2004 conference on computer vision and pattern recognition workshop*, pages 178–178. IEEE, 2004.

[9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[10] Pang Wei Koh, Shiori Sagawa, Henrik Marklund, Sang Michael Xie, Marvin Zhang, Akshay Balsubramani, Weihua Hu, Michihiro Yasunaga, Richard Lanas Phillips, Irena Gao, et al. Wilds: A benchmark of in-the-wild distribution shifts. In *International Conference on Machine Learning*, pages 5637–5664. PMLR, 2021.

[11] Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy M Hospedales. Deeper, broader and artier domain generalization. In *Proceedings of the IEEE international conference on computer vision*, pages 5542–5550, 2017.

[12] Guangrui Li, Guoliang Kang, Wu Liu, Yunchao Wei, and Yi Yang. Content-consistent matching for domain adaptive semantic segmentation. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIV*, pages 440–456. Springer, 2020.

[13] Xiaotong Li, Yongxing Dai, Yixiao Ge, Jun Liu, Ying Shan, and Lingyu Duan. Uncertainty modeling for out-of-distribution generalization. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. ICLR, 2022.

[14] Sara Magliacane, Thijs Van Ommen, Tom Claassen, Stephan Bongers, Philip Versteeg, and Joris M Mooij. Domain adaptation by using causal inference to predict invariant conditional distributions. *Advances in neural information processing systems*, 31, 2018.

[15] Divyat Mahajan, Shruti Tople, and Amit Sharma. Domain generalization using causal matching. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pages 7313–7324. PMLR, 2021. URL http://proceedings.mlr.press/v139/mahajan21b.html.

[16] Sangwoo Mo, Hyunwoo Kang, Kihyuk Sohn, Chun-Liang Li, and Jinwoo Shin. Object-aware contrastive learning for debiased scene representation. *Advances in Neural Information Processing Systems*, 34:12251–12264, 2021.

[17] Oren Nuriel, Sagie Benaim, and Lior Wolf. Permuted adain: Reducing the bias towards global statistics in image classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9482–9491, 2021.

[18] Judea Pearl. *Causality*. Cambridge university press, 2009.

[19] 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)*, pages 947–1012, 2016.

[20] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021.

[21] Stephan R Richter, Vibhav Vineet, Stefan Roth, and Vladlen Koltun. Playing for data: Ground truth from computer games. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part II 14*, pages 102–118. Springer, 2016.

[22] Ergys Ristani, Francesco Solera, Roger Zou, Rita Cucchiara, and Carlo Tomasi. Performance measures and a data set for multi-target, multi-camera tracking. In *Computer Vision–ECCV 2016 Workshops: Amsterdam, The Netherlands, October 8-10 and 15-16, 2016, Proceedings, Part II*, pages 17–35. Springer, 2016.

[23] Mateo Rojas-Carulla, Bernhard Schölkopf, Richard Turner, and Jonas Peters. Invariant models for causal transfer learning. *The Journal of Machine Learning Research*, 19(1):1309–1342, 2018.

[24] Bryan C Russell, Antonio Torralba, Kevin P Murphy, and William T Freeman. Labelme: a database and web-based tool for image annotation. *IJCV*, 77(1-3):157–173, 2008.

[25] Hemanth Venkateswara, Jose Eusebio, Shayok Chakraborty, and Sethuraman Panchanathan. Deep hashing network for unsupervised domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5018–5027, 2017.

[26] Jianxiong Xiao, James Hays, Krista A Ehinger, Aude Oliva, and Antonio Torralba. Sun database: Large-scale scene recognition from abbey to zoo. In *2010 IEEE computer society conference on computer vision and pattern recognition*, pages 3485–3492. IEEE, 2010.

[27] Liang Zheng, Liyue Shen, Lu Tian, Shengjin Wang, Jingdong Wang, and Qi Tian. Scalable person re-identification: A benchmark. In *Proceedings of the IEEE international conference on computer vision*, pages 1116–1124, 2015.

[28] Bolei Zhou, Aditya Khosla, Àgata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 2921–2929. IEEE Computer Society, 2016. doi: 10.1109/CVPR.2016.319. URL https://doi.org/10.1109/CVPR.2016.319.

[29] Kaiyang Zhou, Yongxin Yang, Yu Qiao, and Tao Xiang. Domain generalization with mixstyle. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. ICLR, 2021.