Counterfactual Zero-Shot and Open-Set Visual Recognition

Zhongqi Yue[1,3]*; **Tan Wang**[1]*; **Hanwang Zhang**[1], **Qianru Sun**[2], **Xian-Sheng Hua**[3]

[1]Nanyang Technological University, [2]Singapore Management University, [3]Damo Academy, Alibaba Group

yuez0003@ntu.edu.sg, TAN317@e.ntu.edu.sg, hanwangzhang@ntu.edu.sg,
qianrusun@smu.edu.sg, xiansheng.hxs@alibaba-inc.com

Abstract

We present a novel counterfactual framework for both Zero-Shot Learning (ZSL) and Open-Set Recognition (OSR), whose common challenge is generalizing to the unseen-classes by only training on the seen-classes. Our idea stems from the observation that the generated samples for unseen-classes are often out of the true distribution, which causes severe recognition rate imbalance between the seen-class (high) and unseen-class (low). We show that the key reason is that the generation is not ***Counterfactual Faithful****, and thus we propose a faithful one, whose generation is from the sample-specific counterfactual question: What would the sample look like, if we set its class attribute to a certain class, while keeping its sample attribute unchanged? Thanks to the faithfulness, we can apply the* ***Consistency Rule*** *to perform unseen/seen binary classification, by asking: Would its counterfactual still look like itself? If "yes", the sample is from a certain class, and "no" otherwise. Through extensive experiments on ZSL and OSR, we demonstrate that our framework effectively mitigates the seen/unseen imbalance and hence significantly improves the overall performance. Note that this framework is orthogonal to existing methods, thus, it can serve as a new baseline to evaluate how ZSL/OSR models generalize. Codes are available at* https://github.com/yue-zhongqi/gcm-cf.

1. Introduction

Generalizing visual recognition to novel classes unseen in training is perhaps the Holy Grail of machine vision [39]. For example, if machines could classify new classes accurately by Zero-Shot Learning (ZSL[1]) [31, 63], we could collect labelled data as many as possible for free; if machines could reject samples of unknown classes by Open-Set Recognition (OSR) [52, 11], any recognition system would be shielded against outliers. Unfortunately, this goal is far from achieved, as it is yet a great challenge for them to "imagine" the unseen world based on the seen one [30, 53].

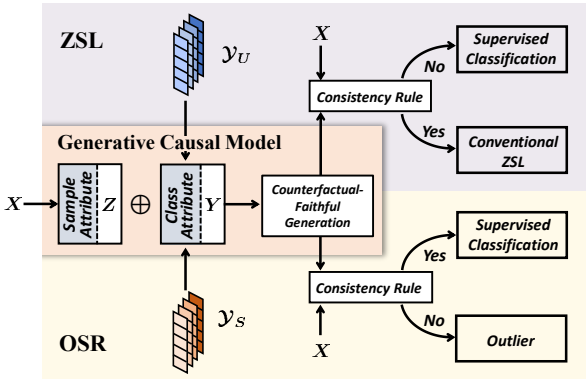

Figure 1: Our counterfactual framework for ZSL and OSR. Here $\oplus$ denotes vector concatenation, $\mathcal{Y}_U$ and $\mathcal{Y}_S$ denote the set of unseen and seen class attributes, respectively.

Over the past decade, all the unseen-class recognition methods stem from the same grand assumption: *attributes (or features) learned from the training seen-classes are transferable to the testing unseen-classes*. Therefore, if we have the ground-truth *class attributes* describing both of the seen- and unseen-classes (or only those of the seen in OSR), ZSL (or OSR) can be accomplished by comparing the predicted class attributes of the test sample and the ground-truth ones [16, 6]; or by training a classifier on the samples generated from the ground-truth attributes [66, 44].

Not surprisingly, the above assumption is hardly valid in practice. As the model only sees the seen-classes in training, it will inevitably cater to the seen idiosyncrasies, and thus result in an unrealistic imagination of the unseen world. Figure 2a illustrates that the samples, generated from the class attribute of an unseen-class, do not lie in the sample domain between the ground-truth seen and unseen, *i.e.*, they resemble neither the seen nor the unseen. As a result, the seen/unseen boundary learned from the generated unseen and the true seen samples is imbalanced. Such imbalanced

*Equal contribution

[1]Conventional ZSL [31] only evaluates the recognition on the unseen-classes. We refer to ZSL as a more challenging setting: Generalized ZSL [12], which is evaluated on the both seen- and unseen-classes.

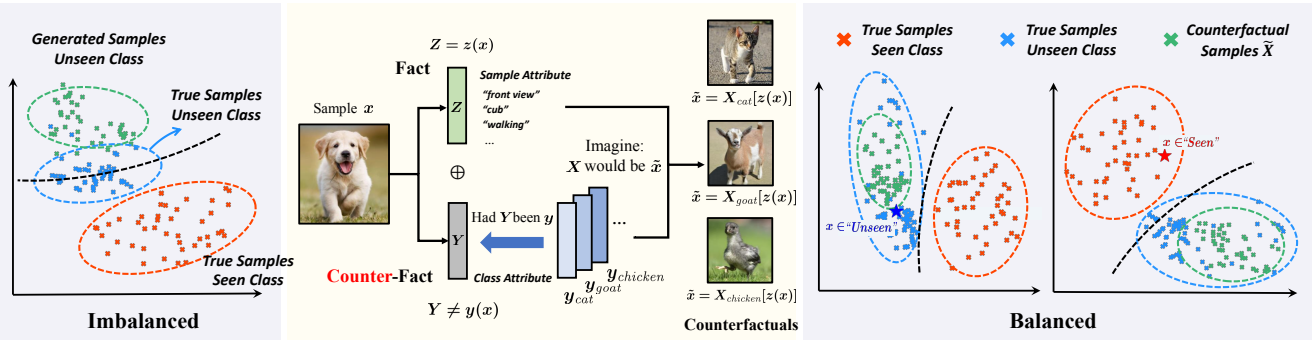

Figure 2: (a) t-SNE [37] plot of the CUB [61] samples in ZSL using a conventional unseen generation method [41], where a single pair of seen- and unseen-class is shown to avoid clutter. Due to the out-of-distribution generation (green), the decision boundary (black dashed lines) is imbalanced between the true seen (red) and unseen (blue) samples. (b) Illustration of our counterfactual generation. (c) t-SNE plot of the CUB using our counterfactual generation. The decision boundary is balanced. See Figure 4 for more examples of OSR.

classification increases the recall of the seen-class by sacrificing that of the unseen. Interestingly, we find that all the existing ZSL methods suffer from this imbalance: the seen accuracy is much higher than the unseen (*cf.* Table 3).

Astute readers may intuitively realize that it is all about *disentanglement*: if every attribute is disentangled, any unseen combination will be sensible. However, it is well-known that learning disentangled features is difficult, or even impossible without proper supervision [36]. In this paper, we propose another way around: **Counterfactual Inference** [48], which does not require the disentanglement for the class attributes. As illustrated in Figure 2b, denote X as the sample variable, which can be encoded into the *sample* attributes $Z = z(X = x)$ (*e.g.*, "front view") and *class* attributes $Y = y(X = x)$. The counterfactual [47]: $\tilde{x} = X_y[z(x)]$ of sample x, is read as:

> X would be $\tilde{x}$, had Y been y,
> given the fact that $Z = z(X = x)$.

Note that the given "fact" is $Z = z(x)$ and the "counterfact" (what if) is $Y \neq y(x)$, indicating that the encoded $y(x)$ — also an observed fact — clashes with the counterfact y. The key difference between the conventional unseen generation and the counterfactual is that: the former is purely based on the **sample-agnostic** class attributes y (or together with Guassian prior z [65]), while the latter is also grounded by the **sample-specific** attributes $z(x)$.

So, why sample-specific? The imagination of an unseen-class is indeed not necessary to start from scratch, *i.e.*, purely from the class definition Y, where the effect of some attributes may be lost due to entanglement [13]; instead, it should also start from the observed fact $Z = z(x)$, which can make up those entangled attributes. In fact, such grounded generation simulates how we humans imagine an unseen sample [49]: we imagine a "dinosaur" (class attributes) based on its fossil (sample attributes). More formally, by disentangling the two groups: class attribute Y and sample Z, we can use the theorem of **Counterfactual Faithfulness** (Section 3.3 & 3.4) to guarantee that the counterfactual distribution is coherent with the ground-truth seen/unseen distribution. As shown in Figure 2c (left), the unseen-class generation grounded by a sample is in the true unseen domain, leading to a more balanced decision boundary. It is worth noting that the group disentanglement between Z and Y is much more relaxed and thus more approachable than the full disentanglement [7, 8], *i.e.*, the attributes within each group are not required to be disentangled. In Section 3.4, we design a training procedure to achieve this.

We propose a counterfactual framework for ZSL and OSR, because they are both underpinned by the generalization to unseen-classes. As summarized in Figure 1, the counterfactual is powered by a Generative Causal Model (Section 3.2). The counterfactual faithfulness allows us to use the **Consistency Rule**: if the counter-fact y is indeed the underlying ground-truth, the counterfactual $\tilde{x}$ equals to the factual x. Therefore, we can use the rule as a seen/unseen binary classifier by varying y across the seen/unseen class attributes (Section 3.3): If a sample is seen, we can apply the conventional supervised learning classifier; otherwise, for ZSL, we apply the conventional ZSL classifier, and for OSR, we reject it.

To the best of our knowledge, the proposed counterfactual framework is the first to provide a theoretical ground for balancing and improving the seen/unseen classification. In particular, we show that the quality of disentangling Z and Y is the key bottleneck, so it is a potential future direction for ZSL/OSR [58, 19, 46, 9]. Our method can serve as an unseen/seen binary classifier, which can plug-and-play and boost existing ZSL/OSR methods to achieve new state-of-the-arts (Section 4).

2. Related Work

ZSL is usually provided with an auxiliary set of class attributes to describe each seen- and unseen-class [31, 63]. Therefore, ZSL can be approached by either inferring a sample's attribute and finding the closest match in the attribute space [16, 1, 13, 25], or generating features using the attributes and matching in the feature space [65, 34, 41, 45]. **OSR** is a more challenging open environment setting with no information on the unseen-classes [51, 52], and the goal is to build a classifier for seen-classes that additionally rejects unseen-class samples as outliers. The inference methods in OSR calibrate a discriminative model by adjusting the classification logits [6, 75], and the generation methods estimate the seen-class density and reject test samples in the low-density area as outliers [44, 56]. **Out-Of-Distribution (OOD)** detection [17, 2, 35] also focuses on unseen detection, where seen and unseen samples are usually from different domains [10]. However, in OOD, the unseen-class information is available. In particular, some works employ OOD techniques in ZSL to distinguish seen- and unseen-classes [38, 14]. Our work is based on **Causal Inference** [47], which has shown promising results in various computer vision tasks [72, 60, 69] including few-shot classification [71], long-tailed classification [59] and incremental learning [21]. In particular, a recent paper [5] uses causal intervention on ZSL. However, it builds on a restrictive setting where class attributes are fully disentangled (*e.g.*, shape and color). Our work uses *counterfactual inference* and relies on the much-relaxed group disentanglement [8], allowing us to outperform on complex benchmark datasets.

3. Approach

3.1. Problem Definitions

We train a model using a labelled dataset $\mathcal{D} = \{\mathbf{x}_i, l_i\}_{i=1}^N$ on seen-classes $\mathcal{S}$, to recognize samples from both $\mathcal{S}$ and unseen-classes $\mathcal{U}$, where $\mathbf{x}_i \in \mathcal{X} \subset \mathbb{R}^d$ is the d-dimensional feature vector of the i-th sample and $l_i \in \mathcal{S}$ is its corresponding label. We assume that the samples of $\mathcal{S}$ and $\mathcal{U}$ are embedded in the same feature space $\mathcal{X}$, whose ambient space is the RGB color space or the network feature space provided by the dataset curator [63].

Zero-Shot Learning (ZSL). It includes two settings: 1) *Conventional ZSL*, where the model is only evaluated on $\mathcal{U}$, and 2) *Generalized ZSL*, where the model is evaluated on $\mathcal{S} \cup \mathcal{U}$. A common practice [12] is to use an additional set of class attribute vectors $\mathcal{Y}_S$ and $\mathcal{Y}_U$ to describe the seen- and unseen-classes, respectively. Compared to the one-hot label embeddings, these attributes can be considered as dense label embeddings [55, 40]. When the context is clear, we refer to ZSL as Generalized ZSL.

Open-Set Recognition (OSR). It is evaluated on both $\mathcal{S}$ and $\mathcal{U}$. Compared to ZSL, $\mathcal{U}$ in OSR is marked as "unknown". Instead of using dense labels, each seen-class is described by the one-hot embedding of K dimensions, where K is the number of seen-classes. The one-hot embeddings are considered as the seen-class attributes set $\mathcal{Y}_S$. Since OSR is evaluated in an open environment, there is no unseen-class attributes set $\mathcal{Y}_U$.

3.2. Generative Causal Model

Our assumption is that both ZSL and OSR follow a Generative Causal Model (GCM) [47] shown in Figure 3, where the *class attribute* Y and the *sample attribute* Z jointly determine the observed image feature X. In general, the generative causal process $Z \to X, Y \to X$ can be confounded, represented as the dashed links in Figure 3. One can remove the confounders through more elaborate data collection [63] and experimental design [24]. In this paper, we follow the conventions in ZSL and OSR to ignore the confounders [34, 44]. Our GCM entails a generation and inference process. Specifically, given Z and Y, we can generate X by sampling from the conditional distribution $P_\theta(X|Z,Y)$. While given X, we can infer Z and Y through the posterior $Q_\phi(Z|X)$ and $Q_\psi(Y|X)$. Therefore, GCM can support both generative and inference-based methods. As the former prevails over the latter in literature, we focus on it in this paper.

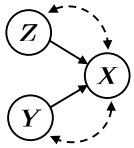

Figure 3: Our GCM for ZSL and OSR.

In ZSL, Y takes values fom the dense labels set $\mathcal{Y}_S \cup \mathcal{Y}_U$. Generative ZSL methods [65, 34] aim to learn $P_\theta(X|Z,Y)$. In testing, they generate unseen-class samples X by using a Gaussian prior as Z and the attributes Y in $\mathcal{Y}_U$. Then, the generated unseen samples and the seen $\mathcal{D}$ are used to train a classifier to recognize both $\mathcal{S}$ and $\mathcal{U}$. In OSR, the value of Y is from the one-hot embedding set $\mathcal{Y}_S$. Generative OSR methods [44, 56] first infer Y from a test sample $\mathbf{x}$ by sampling $\mathbf{y} \sim Q_\psi(Y|X=\mathbf{x})$. Then, the inferred $\mathbf{y}$ and Gaussian noise $\mathbf{z}$ are used to generate $\mathbf{x}'$ from $P_\theta(X|Z=\mathbf{z}, Y=\mathbf{y})$. Finally, the sample is marked as "unknown" if $\mathbf{x}$ is dissimilar to $\mathbf{x}'$ subject to a specified threshold.

Note that all the above generation methods adopt a prior Gaussian noise for Z, which is not sample-specific. As Y is inevitably entangled (examples in Appendix), there is no mechanism to make up for the missing sample attribute effect during generation, so, the generated unseen-class samples will be unrealistic and lie outside $\mathcal{X}$, rendering the decision rule trained on the seen samples inapplicable.

3.3. Counterfactual Generation and Inference

Given a sample $\mathbf{x}$, we use the GCM to generate counterfactual samples $\tilde{\mathbf{x}} = X_{\mathbf{y}}[z(\mathbf{x})]$ following the three steps of computing counterfactual [48]:

Abduction-"*given the fact that* $Z = z(\mathbf{x})$". We solve for

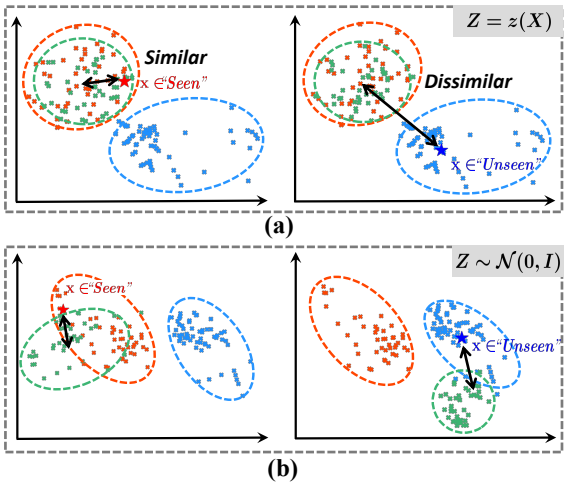

Figure 4: OSR using (a) our counterfactual framework (Z=$z(X)$); (b) existing generative approach ($Z \sim \mathcal{N}(0, I)$) when the test sample is from seen-classes (left) and unseen-classes (right). Red dashed line denotes the true samples from the seen-class and blue dash line as those from the unseen-class. Green dash line denotes generated samples using the seen-class attribute. ⋆ and ⋆ denote the samples from seen and unseen class, respectively.

the endogenous sample attribute $z(\mathbf{x})$ given the evidence $X = \mathbf{x}$. In our GCM, we can sample from the posterior $z(\mathbf{x}) \sim Q_\phi(Z|X = \mathbf{x})$.

Action-*"had Y been y"*. Here $\mathbf{y} \in \mathcal{Y}_S \cup \mathcal{Y}_U$ is the intervention target of Y. Note that the class attribute Y of this sample can be inferred from $y(\mathbf{x}) \sim Q_\psi(Y|X = \mathbf{x})$, but in this step, we intervene on Y by discarding the inferred value and setting Y as $\mathbf{y}$, which may be different from $y(\mathbf{x})$.

Prediction-*"X would be* $\tilde{\mathbf{x}}$*"*. Conditioning on the inferred $Z = z(\mathbf{x})$ (fact) and the intervention target $Y = \mathbf{y}$ (counterfact), we can generate the counterfactual sample $\tilde{\mathbf{x}}$ from $P_\theta(X|Z = z(\mathbf{x}), Y = \mathbf{y})$. Notice that the feature imagination from class attribute $Y = \mathbf{y}$ is now grounded by the sample attribute $Z = z(\mathbf{x})$.

We want that the above counterfactual sample lies in the true distribution of the seen or unseen samples. Formally, such property is defined as below:

Definition (Counterfactual Faithfulness). *Given* $\mathbf{x} \in \mathcal{X}$*, the counterfactual generation* $\tilde{\mathbf{x}}$ *using a GCM is faithful whenever* $\tilde{\mathbf{x}} \in \mathcal{X}$.

In fact, the failure of existing methods, as mentioned in Section 3.2, is because the counterfactual faithfulness does not hold; however, it holds for our generation. We delay the justification to Section 3.4. It assures that any distance metric in $\mathcal{X}$ is applicable for both $\mathbf{x}$ and its counterfactual generation $\tilde{\mathbf{x}}$. This allows us to build a binary classifier on seen/unseen by applying the **Consistency Rule**:

$$y^*(\mathbf{x}) = \mathbf{y} \Longrightarrow X_\mathbf{y}[z(\mathbf{x})] = \mathbf{x}, \tag{1}$$

where $y^*(\mathbf{x})$ is the (unobserved) ground-truth class attribute of $\mathbf{x}$ and $X_\mathbf{y}[z(\mathbf{x})]$ is the generated counterfactual with this ground-truth. Our binary classification strategy is based on the equivalent *contraposition* of the rule, which states that *if* $\mathbf{x}$ *is dissimilar to* $X_\mathbf{y}[z(\mathbf{x})]$*, the ground-truth attribute of* $\mathbf{x}$ *cannot be* $\mathbf{y}$:

$$X_\mathbf{y}[z(\mathbf{x})] \neq \mathbf{x} \Longrightarrow y^*(\mathbf{x}) \neq \mathbf{y}. \tag{2}$$

Thanks to the counterfactual faithfulness, the dissimilarity can be measured by any distance metric defined in $\mathcal{X}$ (*e.g.*, Euclidean distance). Figure 4 uses the OSR task to show the benefit of our counterfactual approach grounded by $Z = z(X)$. The class-agnostic Z in the existing methods cannot make up for the attributes that entangled in Y. This leads to non-faithful generation on unseen-class samples, and the distance can hardly tell apart the seen and unseen-class samples (Figure 4b). In contrast, our approach achieves counterfactual faithfulness and thus the distance remains discriminative (Figure 4a). Next, we detail the binary inference rule for both ZSL and OSR.

Inference in ZSL. Since ZSL is evaluated in a closed environment, *i.e.*, the set of unseen-classe attributes $\mathcal{Y}_U$ is known in testing, we use the contraposition—if feature $\mathbf{x}$ is *dissimilar* to counterfactual generations from the *unseen*-classes, $\mathbf{x}$ belongs to *seen*. For example in Figure 2c, conditioning on a seen-class sample $\mathbf{x}$, the counterfactual generations using the unseen-class attribute are indeed dissimilar to $\mathbf{x}$ as they lie in the opposite side of the classifier decision boundary. Specifically, we generate a set of counterfactual features $\tilde{X}$ of the *unseen*-classes by using the inferred Z from $Q_\phi(Z|X = \mathbf{x})$ and intervening Y with the ground-truth attributes from $\mathcal{Y}^U$. Using the counterfactual set $\tilde{X}$ of the unseen-classes and $\mathcal{D}$ of the seen-classes, we train a multi-label classifier whose vocabulary is $\mathcal{S} \cup \mathcal{U}$. Denote the mean-pooling of the top-K classifier probabilities among seen- and unseen-classes as S^K and U^K, respectively. The binary seen/unseen label $b(\mathbf{x})$ is given by:

$$b(\mathbf{x}) = \begin{cases} \text{seen}, & \text{if } U^K < S^K \\ \text{unseen}, & \text{otherwise} \end{cases} \tag{3}$$

Inference in OSR. Since OSR is in an open environment, *i.e.*, there could be infinite number of unseen-classes, it is impossible to generate unseen-class counterfactuals. Therefore, we use the contraposition the other way round—if $\mathbf{x}$ is *dissimilar* to the counterfactual generations of the *seen*-classes, $\mathbf{x}$ belongs to the *unseen*. This is shown in Figure 4a, where the unseen-class sample (right), compared to the seen-class sample (left), is much more dissimilar to the seen-class counterfactuals. Hence by thresholding the dissimilarity, we can classify both samples correctly. Specifically, we generate a set of counterfactual features $\tilde{X}$ on *seen*-classes by setting Y as the one-hot embeddings in $\mathcal{Y}_S$,

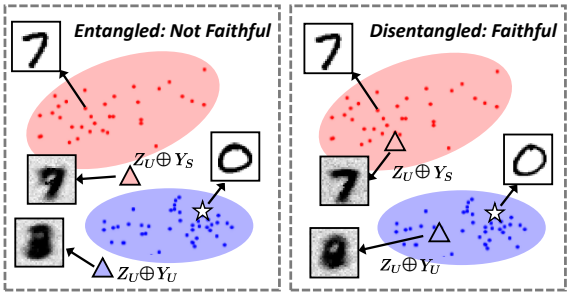

Figure 5: Counterfactual generation conditioned on an unseen sample (denoted as the star). $Z_U \oplus Y_S$ and $Z_U \oplus Y_U$ represent counterfactual images conditioned on the unseen sample, using seen and unseen attribute respectively.

and we calculate the minimum Euclidean distance between $\mathbf{x}$ and each $\tilde{\mathbf{x}} \in \tilde{X}$, denoted as d_{min}. If d_{min} is larger than a threshold τ, the sample is dissimilar to the seen-classes counterfactuals. The binary label $b(\mathbf{x})$ is given by:

$$b(\mathbf{x}) = \begin{cases} \text{unseen}, & \text{if } d_{min} > \tau \\ \text{seen}, & \text{otherwise} \end{cases} \tag{4}$$

Two-Stage Inference. As shown in Figure 1, after the stage-one binary classification, at the second stage, the predicted seen-classes samples can be classified by using the standard supervised classifier, and the predicted unseen-classes samples can be fed into any Conventional ZSL algorithms in ZSL, or rejected as outliers in OSR. The detailed implementation is in Section 4.2.

3.4. Counterfactual-Faithful Training

Our framework centers on the counterfactual-faithful generation, which can be guaranteed by the following theorem (proved as a corollary of [7], Appendix):

Theorem. *The counterfactual generation $X_{\mathbf{y}}[z(\mathbf{x})]$ is faithful if and only if the sample attribute Z and class attribute Y are group disentangled.*

Figure 5 shows the difference between the generations with and without the group disentanglement. Therefore, to achieve counterfactual faithful generation, the full disentanglement among each dimension of Y and Z, which may be impossible in general [36], is relaxed to group disentanglement between Y and Z, which can be more easily addressed by the following specially designed training objective:

$$\min_{\theta,\phi} \mathcal{L}_Z + \nu \mathcal{L}_Y + \max_{\omega} \rho \mathcal{L}_F, \tag{5}$$

where ν, ρ are a trade-off parameters. All three losses are designed to achieve the counterfactual faithfulness, which assures the distance metric of the consistency rule. Next, we detail each of them.

Disentangling Z from Y. We minimize the β-VAE [20] loss $\mathcal{L}_Z$ given by:

$$\mathcal{L}_Z = -\mathbb{E}_{Q_\phi(Z|X)}\left[P_\theta(X \mid Z, Y)\right] + \beta D_{KL}\left(Q_\phi(Z \mid X) \,\|\, P(Z)\right), \tag{6}$$

where D_{KL} denotes KL-divergence, $P_\theta(X|Z,Y)$ and $Q_\phi(Z|X)$ are implemented using the Deep Gaussian family [27], and the prior $P(Z)$ is set to the isotropic Gaussian distribution. Compared to the standard VAE objective, β-VAE re-weighs the KL divergence term by a factor of β ($\beta > 1$). This is shown to be highly effective in learning a disentangled sample attribute Z [20, 58]. Intuitively, by placing a strict constraint for Z as a whole to follow the endogenous prior $P(Z)$, the value of Z becomes unaffected by the distribution of Y, *i.e.*, Z is disentangled from Y.

Disentangling Y from Z. However, the above regularization cannot guarantee that the GCM correctly uses the class attribute Y during generation. For example, recent GAN models [74, 26] can generate a large variety of photo-realistic images by only using Z. In fact, as shown in recent literature [9], it is possible for an over-parameterized model $P_\theta(X|Z,Y)$ to ignore Y and use purely Z to generate X, leading to non-faithful generations. This is because the information in Y might be *fully contained* in Z. Therefore, it is necessary to additionally disentangle Y from Z. Specifically, given a training sample $\mathbf{x}$, its ground-truth attribute $\mathbf{y}$ and its sample attribute $z(\mathbf{x})$ sampled from $Q_\phi(Z|X=\mathbf{x})$, we require $\mathbf{x}$ to be close to $\mathbf{x}_\mathbf{y} = X_\mathbf{y}[z(\mathbf{x})]$, but far away from the counterfactual generations in the set $\tilde{X} = \{X_{\mathbf{y}'}[z(\mathbf{x})] \mid \mathbf{y}' \in \mathcal{Y}^S \wedge \mathbf{y}' \neq \mathbf{y}\}$. We use a contrastive loss as:

$$\mathcal{L}_Y = -log \frac{exp\left(-dist(\mathbf{x}, \mathbf{x}_\mathbf{y})\right)}{\sum_{\mathbf{x}' \in \tilde{X} \cup \{\mathbf{x}_\mathbf{y}\}} exp\left(-dist(\mathbf{x}, \mathbf{x}')\right)}, \tag{7}$$

where $dist$ denotes Euclidean distance. Intuitively, this loss actively intervenes the class attribute Y, given fixed sample attributes Z, and therefore maximizes the sample difference (*e.g.*, in terms of constrastiveness) before and after the intervention, which complies with the recent definition of causal disentanglement [19]. Therefore, $\mathcal{L}_Y$ can disentangles Y from Z.

Further Disentangling by Faithfulness. Note that the faithfulness $\tilde{\mathbf{x}} \in \mathcal{X}$ is a necessary condition for disentanglement, so we can also use it as an objective. Note that the VAE objective optimizes a lower bound of the likelihood $P(X)$, where the bound looseness undermines the faithfulness. To address this problem, we additionally use the Wasserstein GAN (WGAN) [3] loss. Specifically, we train a discriminator $D(X,Y)$ parameterized by ω that outputs a real value, indicating if feature X is realistic conditioned on Y (larger is more realistic). Given a feature $\mathbf{x}$ with attribute $\mathbf{y}$, we can generate the counterfactual $\mathbf{x}'$ using $z(\mathbf{x})$ and $\mathbf{y}$. The discriminator is trained to mark $\mathbf{x}$ as real (large

$D(\mathbf{x}, \mathbf{y})$) and $\mathbf{x}'$ as unreal (small $D(\mathbf{x}', \mathbf{y})$). Specifically, the WGAN loss $\mathcal{L}_F$ is given by:

$$\begin{aligned}\mathcal{L}_F =& \mathbb{E}[D(\mathbf{x}, \mathbf{y})] - \mathbb{E}[D(\mathbf{x}', \mathbf{y})] \\ & - \lambda \mathbb{E}\left[(\|\nabla_{\hat{\mathbf{x}}} D(\hat{\mathbf{x}}, \mathbf{y})\|_2 - 1)^2\right],\end{aligned} \tag{8}$$

where λ is a penalty term, $\hat{\mathbf{x}} = \alpha \mathbf{x} + (1 - \alpha)\mathbf{x}'$ with α sampled from the uniform distribution defined on $[0, 1]$. By training the generation process $P_\theta(X|Z, Y)$ and the discriminator in an adversarial fashion, the GCM is regularized to generate $\mathbf{x}'$ that is similar to $\mathbf{x}$ in the same distribution, *i.e.*, counterfactual-faithful for seen-class samples.

4. Experiments

4.1. Datasets

ZSL. We evaluated our method on standard benchmark datasets: Caltech-UCSD-Birds 200-2011 (*CUB*) [61], *SUN* [67], Animals with Attributes 2 (*AWA2*) [63] and attribute Pascal and Yahoo (*aPY*) [15]. In particular, we followed the unseen/seen split in the *Proposed Split (PS) V2.0* [63], recently released to fix a test-data-leaking bug in the original PS. The granularity, total #images, #seen-classes ($|\mathcal{S}|$) and #unseen-classes ($|\mathcal{U}|$) are given in Table 1.
OSR. We used the standard evaluation datasets: *MNIST* [33], *SVHN* [43], *CIFAR10* [29] and *CIFAR100* [29]. The image size, #classes and # images in train/test split of the datasets are given in Table 2. Following the standard benchmark [42, 70] in OSR, we split MNIST, SVHN and CIFAR10 into 6 seen-classes and 4 unseen-classes, and construct two additional datasets CIFAR+10 (*C+10*) and CIFAR+50 (*C+50*), where 4 non-animal classes in CIFAR10 are used as seen-classes, while additional 10 and 50 animal classes from CIFAR100 are used as unseen-classes.

Dataset	Granularity	Total	$\lvert\mathcal{S}\rvert$	$\lvert\mathcal{U}\rvert$
CUB [61]	Fine	11,788	150	50
SUN [67]	Fine	14,340	645	72
AWA2 [63]	Coarse	37,322	40	10
aPY [15]	Coarse	12,051	20	12

Table 1: Information on ZSL datasets.

Dataset	Image Size	Classes	Train	Test
MNIST [33]	28×28	10	60,000	10,000
SVHN [43]	32×32	10	73,257	26,032
CIFAR10 [29]	32×32	10	50,000	10,000
CIFAR100 [29]	32×32	100	50,000	10,000

Table 2: Information on OSR datasets.

4.2. Evaluation Metrics and Settings

ZSL Evaluation. It was conducted in the Generalized ZSL setting. We used two metrics: 1) **ZSL Accuracy**. It consists of 3 numbers (U, S, H), where U/S is the per-class top-1 accuracy of unseen-/seen-classes test samples, and H is the harmonic mean of U, S, given by $H = 2 \times S \times U/(S + U)$. 2) **CVb**. To measure the balance between unseen/seen classification, we propose to use the Coefficient of Variation of the seen and unseen binary classification accuracy, denoted as CVb. Let S_b and U_b be the binary accuracy on seen- and unseen-classes, respectively. CVb is given by $\sqrt{0.5(S_b - \mu)^2 + 0.5(U_b - \mu)^2}/\mu$, where $\mu = (S_b + U_b)/2$. Note that the variation between S and U of ZSL Accuracy is not a good measure of balance, as they are affected by the number of seen- and unseen-classes, which can be quite different (see SUN in Table 1). 3) **AUSUC**. We draw the Seen-Unseen accuracy Curve (SUC) by plotting a series of S against U of ZSL Accuracy, where the series is obtained by adjusting a calibration factor ω that is subtracted from the classifier logits on the seen-classes. Then we use the Area Under SUC (AUSUC) for evaluation. Compared to a single ZSL Accuracy, SUC and the area provide a more detailed view of the capability of an algorithm to balance the unseen-seen decision boundary [12, 13].
OSR Evaluation. We used the following metrics: 1) Macro-averaged **F1 scores** over seen-classes and "unknown" (for all unseen-classes), which shows how well a method can recognize seen classes while rejecting unseen-classes samples; 2) **Openness-F1 Plot**. We also studied the response of F1 scores under varying *openness* given by $1 - \sqrt{2N/(N + M)}$, where N and M are number of seen- and unseen-classes, respectively. Compared to a single F1 score where the openness is fixed, this plot shows the robustness of an OSR classifier to the open environment with an unknown number of unseen-classes.
Implementation Details. For ZSL, we implemented our GCM based on the network architecture in TF-VAEGAN [41]. Following common protocol [65, 34], we used the ResNet-101 [18] features for X and attributes provided in [63] for $\mathcal{Y}_S, \mathcal{Y}_U$. For OSR, our GCM was implemented using the networks in CGDL [56] and X represents actual images. Other details are in Appendix.

4.3. Results on ZSL

Mitigate the Imbalance. As shown in Table 3, our counterfactual approach, denoted as GCM-CF, achieves a **more balanced ZSL Accuracy** and significantly improves the existing state-of-the-art (SOTA) by 2.2% to 4.3%, with a much higher score on U. For example, compared to LisGAN on aPY, our method gains 3.9% on U while sacrificing only 0.1% on S. To further show that GCM-CF mitigates the unseen/seen imbalance, we diagnosed the binary classification accuracy using CVb in Table 4. Note that existing methods have very large CVb, which means that there is a large difference between seen and unseen classification accuracy and reveals the imbalance problem. Our method has the **lowest CVb**. This shows that our approach indeed achieves

	Method	CUB			AWA2			SUN			aPY		
		U	S	H	U	S	H	U	S	H	U	S	H
Inf.	ALE† [1]	23.7	62.8	34.4	14.0	81.8	23.9	21.8	33.1	26.3	4.6	73.7	8.7
	DEVISE† [16]	23.8	53.0	32.8	17.1	74.7	27.8	16.9	27.4	20.9	3.5	78.4	6.7
	LATEM† [62]	15.2	57.3	24.0	11.5	77.3	20.0	14.7	28.8	19.5	1.3	71.4	2.6
	RelationNet [57]	36.3	**63.8**	46.3	22.1	**91.4**	35.5	15.8	25.5	19.6	11.5	**80.7**	20.2
Gen.	GDAN [22]	35.0	28.7	31.6	26.0	78.5	39.1	38.2	19.8	26.1	29.0	63.7	39.9
	CADA-VAE [54]	50.3	56.1	53.0	55.4	76.1	64.0	43.6	36.4	39.7	34.0	54.2	41.7
	LisGAN [34]	44.9	59.3	51.1	53.1	68.8	60.0	41.9	37.8	39.8	33.2	56.9	41.9
	TF-VAEGAN [41]	50.7	62.5	56.0	52.5	82.4	64.1	41.0	**39.1**	40.0	31.7	61.5	41.8
	GCM-CF (Ours)	**61.0**	59.7	**60.3**	**60.4**	75.1	**67.0**	**47.9**	37.8	**42.2**	**37.1**	56.8	**44.9**

Table 3: ZSL Accuracy ($U\%$, $S\%$, $H\%$) on the four datasets, where Inf. means inference-based methods and Gen. means generation-based methods. Note that PS V2.0 was released recently in https://drive.google.com/file/d/1p9gtkuHCCCyjkyezSarCw-1siCSXUykH/view to fix a test-data leaking bug. This can have large impacts on the performance of existing methods, such as GDAN [22]. Therefore all our evaluations were conducted on PS V2.0. † indicates that the results are taken from the PS V2.0 report [64], and we reproduced the results on all other methods using the official code. For our method GCM-CF, we used AREN [68] for Conventional ZSL on CUB, and otherwise used TF-VAEGAN [41] for supervised classification and Conventional ZSL.

Method	CUB	AWA2	SUN	aPY
GDAN [61]	15.1	25.1	27.3	14.4
CADA-VAE [67]	6.7	6.0	7.3	11.2
LisGAN [63]	4.9	7.0	4.6	4.0
TF-VAEGAN [15]	8.0	9.4	10.2	5.8
GCM-CF (Ours)	**1.5**	**2.1**	**1.0**	**2.3**

Table 4: CVb (%) of generative models on all datasets.

Stage 1 / Stage 2	TF-VAEGAN [41]			GCM-CF (Ours)		
	U	S	H	U	S	H
RelationNet [57]	49.3	81.2	61.3	55.8	75.0	**64.0**
CADA-VAE [54]	49.5	81.1	61.5	57.6	75.0	**65.2**
LisGAN [34]	48.8	80.4	60.7	56.1	74.3	**63.9**
TF-VAEGAN [41]	52.8	83.2	64.6	60.4	75.1	**67.0**

Table 5: Comparison of the two-stage inference performance on AWA2 [63] using TF-VAEGAN [41] and our GCM-CF as the stage-one binary classifier. The results on other datasets are shown in Appendix, where using GCM-CF as stage-one binary classifier improves all the methods.

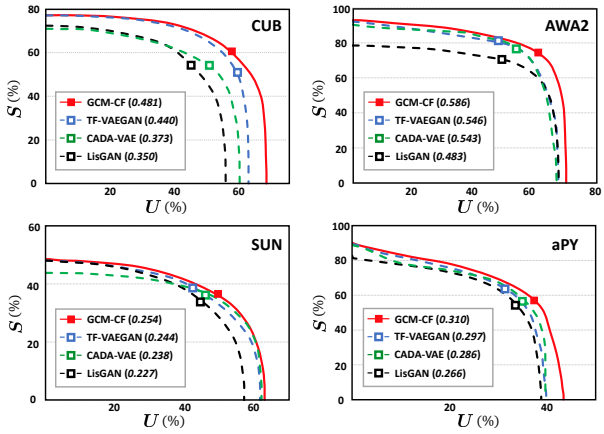

Figure 6: The Seen-Unseen accuracy Curve (SUC) together with the Area Under SUC (AUSUC) on the four datasets. The rectangle denotes the ZSL Accuracy when calibration factor ω=0.

a more balanced binary decision boundary between seen and unseen. However, one may argue that the imbalance problem can be solved by simply adjusting the calibration factor ω. Therefore, we plot the SUC using varying ω and measured the AUSUC on all datasets. The result is shown in Figure 6, where GCM-CF outperforms other methods in every inch and achieves **the best AUSUC**. This shows that GCM-CF fundamentally improves the unseen/seen classification beyond the reach of simple calibration. Overall, the balanced and much improved ZSL Accuracy, lower CVb for binary classification, and higher accuracy on varying calibrations demonstrate that our method mitigates the unseen/seen imbalance in ZSL. This indicates that our GCM generates faithful counterfactuals (see Figure 2c) and supports the effectiveness of our counterfactual-faithful training in disentangling Z and Y.

Stage-One Binary Classifier. Our GCM-CF can serve as a stage-one binary unseen/seen classifier and plug into *all* ZSL methods for subsequent supervised learning on seen and conventional ZSL on unseen. We performed experiments on 4 representative methods—inference based: RelationNet [57], generation with VAE: CADA-VAE [54], with GAN: LisGAN [34] and with VAE-GAN [32]: TF-VAEGAN [41]. The ZSL Accuracy on AWA2 is shown in Table 5. By comparing with Table 3, we observe that GCM-CF can significantly improve all of them on H. For comparison, we used the current SOTA ZSL method TF-VAEGAN as a binary unseen/seen classifier and performed the same experiments. The results on the ZSL Accuracy show that our GCM-CF significantly outperforms TF-VAEGAN. This demonstrates that compared to the class-agnostic Z in existing methods, the use of sample attribute Z that is disentangled from class attribute Y in our GCM-CF is highly effective in improving ZSL performance. Overall, our method as a robust binary classifier can serve as a new strong baseline to evaluate ZSL methods using two-stage inference.

4.4. Results on OSR

Strong Open-Set Classifier. In Table 6, our GCM-CF achieves SOTA F1 scores in all datasets. We discovered that the common evaluation setting [42, 70] of averaging F1 scores over 5 random splits can result in a large variance in

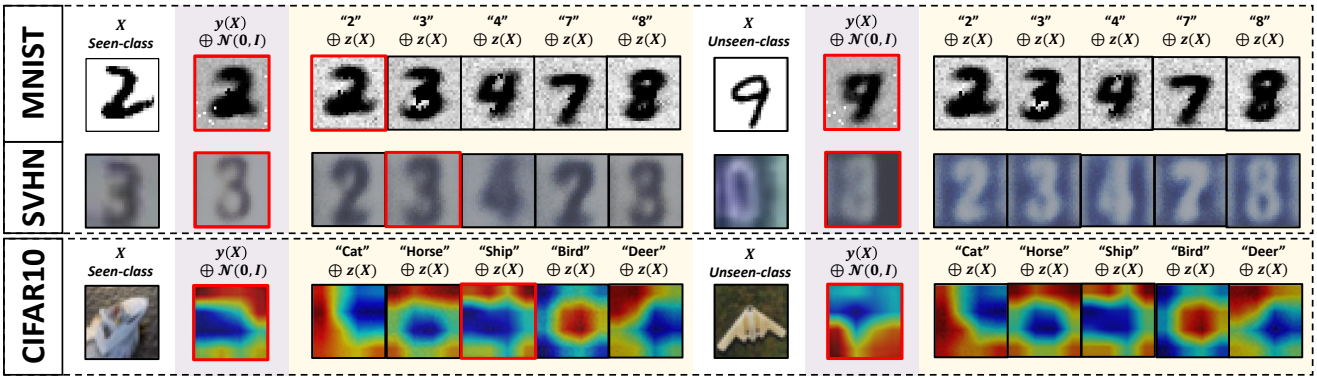

Figure 7: Comparison of the reconstructed images using CGDL [56] ($y(X) \oplus \mathcal{N}(0, I)$) and the counterfactual images generated from our GCM-CF ($y \oplus z(X)$) on the seen- and unseen-class samples. The red box on a generated image denotes that it is similar to the original. For CIFAR10, due to the conflict between visual perception and discriminative training [23], we first train a classifier on seen-classes images and then use CAM [73] to show the sensible yet discrimiantive features: although the pixel-level generation is not sensible, the pixel-level distance remains discriminative as shown in Table 6.

Method	MNIST	SVHN	CIFAR10	C+10	C+50
Softmax	76.82	76.16	70.39	77.82	65.96
OpenMax [6]	85.93	77.95	71.38	78.68	67.68
CGDL [56]	88.95	76.31	71.03	77.92	70.96
GCM-CF (Ours)	**91.37**	**79.25**	**72.63**	**79.38**	**74.60**

Table 6: Comparison of the F1 score averaged over 5 random splits in OSR. We used the official code on CGDL [56] and implemented Softmax and OpenMax [6] for evaluation. For GCM-CF, after binary classification, we used CGDL for supervised classification on the seen-classes.

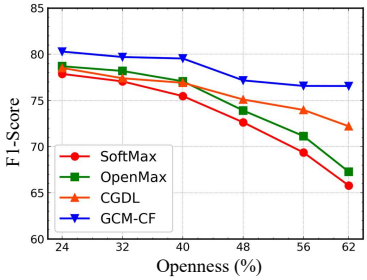

Figure 8: Openness-F1 Plot where 4 non-animal classes from CIFAR10 [29] were used as seen-classes and various classes were drawn from CIFAR100 [29] as unseen-classes.

the F1 score. Therefore, we additionally show the results on all 5 splits in the Appendix, where GCM-CF outperforms other methods in every split. This strongly supports that GCM-CF improves F1 scores in general, *i.e.*, not only on some particular splits. So, where does the improvement come from? To rule out the possibility that F1 improves due to a higher classification accuracy on the seen-classes, we measure the Closed-Set Accuracy of a standard CNN trained in a supervised fashion and that of CGDL, which was used in GCM-CF for 2nd stage supervised classification. The results are shown in the Appendix where the performances are similar. Therefore, the increased F1 score indeed comes from improved unseen/seen binary classification. Furthermore, a strong open-set classifier should stay robust regardless of the number of unseen-classes during evaluation. Therefore, we evaluate OSR classifiers with the Openness-F1 Plot. As shown in Figure 8, our GCM-CF achieves the highest F1 score on all openness settings, and our performance is especially competitive in the challenging environment with large openness. Overall, these results clearly demonstrate the effectiveness of our GCM-CF on seen-unseen classification in OSR.

Qualitative Results. Figure 7 shows OSR evaluation using existing reconstruction-based approach and our counterfactual approach. Notice the reconstructed image on the unseen-class sample (*e.g.*, "0, 9") can be similar to the original image. This makes it difficult to tell apart unseen and seen by thresholding reconstruction error. By using our GCM-CF to generate counterfactuals on each seen-class, the generated images are counterfactual-faithful and indeed look like seen-classes. Note that this holds on CIFAR10, where given seen- or unseen-class samples, the CAMs of the generated counterfactuals using the same seen-class attribute look similar (more results in the Appendix). Therefore, by thresholding the dissimilarity between the test sample and its counterfactuals, we can easily distinguish unseen from seen. These results show the effectiveness of our binary inference strategy. Interestingly, we also observe that the generated samples from GCM indeed capture sample attributes better (*e.g.*, text color, background color on SVHN), which validates that we can learn a disentangled Z using the proposed counterfactual-faithful training.

5. Conclusions

We presented a novel counterfactual framework for Zero-Shot Learning (ZSL) and Open-Set Recognition (OSR) to provide a theoretical ground for balancing and improving the seen/unseen classification imbalance. Specifically, we proposed a Generative Causal Model to generate faithful counterfactuals, which allows us to use the Consistency Rule for balanced binary seen/unseen classification. Extensive results in ZSL and OSR show that our method indeed improves the balance and hence achieves the state-of-the-art performance. As future direction, we will seek new definitions on disentanglement [58] and devise practical implementations to achieve improved disentanglement [9].

6. Acknowledgements

The authors would like to thank all the anonymous reviewers for their constructive comments and suggestions. This research is partly supported by the Alibaba-NTU Singapore Joint Research Institute, Nanyang Technological University (NTU), Singapore; the Singapore Ministry of Education (MOE) Academic Research Fund (AcRF) Tier 1 and Tier 2 grant; and Alibaba Innovative Research (AIR) programme. We also want to thank Alibaba City Brain Group for the donations of GPUs.

References

[1] Zeynep Akata, Florent Perronnin, Zaid Harchaoui, and Cordelia Schmid. Label-embedding for image classification. *TPAMI*, 2015. 3, 7

[2] Martin Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz. Invariant risk minimization. *arXiv preprint arXiv:1907.02893*, 2019. 3

[3] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. Proceedings of Machine Learning Research. PMLR, 2017. 5

[4] Mark Anthony Armstrong. *Basic topology*. Springer Science & Business Media, 2013. 12

[5] Yuval Atzmon, Felix Kreuk, Uri Shalit, and Gal Chechik. A causal view of compositional zero-shot recognition. *Advances in Neural Information Processing Systems*, 2020. 3

[6] Abhijit Bendale and Terrance E Boult. Towards open set deep networks. In *CVPR*, 2016. 1, 3, 8, 15

[7] M. Besserve, A. Mehrjou, R. Sun, and B. Schölkopf. Counterfactuals uncover the modular structure of deep generative models. In *ICLR*, 2020. 2, 5, 12

[8] Michel Besserve, Naji Shajarisales, Bernhard Schölkopf, and Dominik Janzing. Group invariance principles for causal generative models. In *International Conference on Artificial Intelligence and Statistics*, 2018. 2, 3

[9] Michel Besserve, Rémy Sun, Dominik Janzing, and Bernhard Schölkopf. A theory of independent mechanisms for extrapolation in generative models. *arXiv preprint arXiv:2004.00184*, 2020. 2, 5, 8

[10] Saikiran Bulusu, Bhavya Kailkhura, Bo Li, Pramod K Varshney, and Dawn Song. Anomalous instance detection in deep learning: A survey. *arXiv preprint arXiv:2003.06979*, 2020. 3

[11] D. O. Cardoso, F. França, and J. Gama. A bounded neural network for open set recognition. In *2015 International Joint Conference on Neural Networks (IJCNN)*, 2015. 1

[12] Wei-Lun Chao, Soravit Changpinyo, Boqing Gong, and Fei Sha. An empirical study and analysis of generalized zero-shot learning for object recognition in the wild. In *ECCV*, 2016. 1, 3, 6

[13] Long Chen, Hanwang Zhang, Jun Xiao, Wei Liu, and Shih-Fu Chang. Zero-shot visual recognition using semantics-preserving adversarial embedding networks. In *CVPR*, 2018. 2, 3, 6

[14] Xingyu Chen, Xuguang Lan, Fuchun Sun, and Nanning Zheng. A boundary based out-of-distribution classifier for generalized zero-shot learning. In *ECCV*, 2020. 3

[15] Ali Farhadi, Ian Endres, Derek Hoiem, and David Forsyth. Describing objects by their attributes. In *CVPR*, 2009. 6, 7, 13, 14

[16] Andrea Frome, Greg S Corrado, Jon Shlens, Samy Bengio, Jeff Dean, Marc'Aurelio Ranzato, and Tomas Mikolov. Devise: A deep visual-semantic embedding model. In *Advances in neural information processing systems*, pages 2121–2129, 2013. 1, 3, 7

[17] Will Grathwohl, Kuan-Chieh Wang, Jörn-Henrik Jacobsen, David Duvenaud, Mohammad Norouzi, and Kevin Swersky. Your classifier is secretly an energy based model and you should treat it like one. In *ICLR*, 2020. 3

[18] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 6

[19] 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. 2, 5

[20] I. Higgins, Loïc Matthey, A. Pal, C. Burgess, Xavier Glorot, M. Botvinick, S. Mohamed, and Alexander Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. In *ICLR*, 2017. 5

[21] Xinting Hu, Kaihua Tang, Chunyan Miao, Xian-Sheng Hua, and Hanwang Zhang. Distilling causal effect of data in class-incremental learning. In *CVPR*, 2021. 3

[22] He Huang, Changhu Wang, Philip S Yu, and Chang-Dong Wang. Generative dual adversarial network for generalized zero-shot learning. In *CVPR*, 2019. 7

[23] Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Logan Engstrom, Brandon Tran, and Aleksander Madry. Adversarial examples are not bugs, they are features. In *Advances in Neural Information Processing Systems*, 2019. 8

[24] Kosuke Imai, Dustin Tingley, and Teppei Yamamoto. Experimental designs for identifying causal mechanisms. *Journal of the Royal Statistical Society: Series A (Statistics in Society)*, 2013. 3

[25] Huajie Jiang, Ruiping Wang, Shiguang Shan, and Xilin Chen. Transferable contrastive network for generalized zero-shot learning. In *ICCV*, 2019. 3

[26] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *CVPR*, 2019. 5

[27] Diederik Kingma and Max Welling. Auto-encoding variational bayes. In *ICLR*, 2014. 5

[28] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 13

[29] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 6, 8

[30] Brenden M Lake, Tomer D Ullman, Joshua B Tenenbaum, and Samuel J Gershman. Building machines that learn and think like people. *Behavioral and brain sciences*, 2017. 1

[31] C. H. Lampert, H. Nickisch, and S. Harmeling. Learning to detect unseen object classes by between-class attribute transfer. In *CVPR*, 2009. 1, 3

[32] Anders Boesen Lindbo Larsen, Søren Kaae Sønderby, Hugo Larochelle, and Ole Winther. Autoencoding beyond pixels using a learned similarity metric. In *ICML*, 2016. 7, 12

[33] Yann LeCun and Corinna Cortes. MNIST handwritten digit database. 2010. 6

[34] Jingjing Li, Mengmeng Jing, Ke Lu, Zhengming Ding, Lei Zhu, and Zi Huang. Leveraging the invariant side of generative zero-shot learning. In *CVPR*, 2019. 3, 6, 7, 14

[35] Weitang Liu, Xiaoyun Wang, John Owens, and Sharon Yixuan Li. Energy-based out-of-distribution detection. *Advances in Neural Information Processing Systems*, 2020. 3

[36] Francesco Locatello, Stefan Bauer, Mario Lucic, Gunnar Raetsch, Sylvain Gelly, Bernhard Schölkopf, and Olivier Bachem. Challenging common assumptions in the unsupervised learning of disentangled representations. In *ICML*, 2019. 2, 5

[37] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 2008. 2

[38] Devraj Mandal, Sanath Narayan, Sai Kumar Dwivedi, Vikram Gupta, Shuaib Ahmed, Fahad Shahbaz Khan, and Ling Shao. Out-of-distribution detection for generalized zero-shot action recognition. In *CVPR*, 2019. 3

[39] D. Marr. *Vision: A Computational Investigation Into the Human Representation and Processing of Visual Information*. 2010. 1

[40] Jonathan Munro and Dima Damen. Multi-modal domain adaptation for fine-grained action recognition. In *CVPR*, 2020. 3

[41] Sanath Narayan, Akshita Gupta, Fahad Shahbaz Khan, Cees GM Snoek, and Ling Shao. Latent embedding feedback and discriminative features for zero-shot classification. In *ECCV*, 2020. 2, 3, 6, 7, 12, 13, 14

[42] Lawrence Neal, Matthew Olson, Xiaoli Fern, Weng-Keen Wong, and Fuxin Li. Open set learning with counterfactual images. In *ECCV*, 2018. 6, 7

[43] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. In *NeurIPS Workshop on Deep Learning and Unsupervised Feature Learning*, 2011. 6

[44] Poojan Oza and Vishal M Patel. C2ae: Class conditioned auto-encoder for open-set recognition. In *CVPR*, 2019. 1, 3

[45] Ayyappa Pambala, Titir Dutta, and Soma Biswas. Generative model with semantic embedding and integrated classifier for generalized zero-shot learning. In *WACV*, 2020. 3

[46] Giambattista Parascandolo, Niki Kilbertus, Mateo Rojas-Carulla, and Bernhard Schölkopf. Learning independent causal mechanisms. In *ICML*, 2018. 2

[47] Judea Pearl. *Causality: Models, Reasoning and Inference*. Cambridge University Press, 2nd edition, 2009. 2, 3

[48] J. Pearl, M. Glymour, and N.P. Jewell. *Causal Inference in Statistics: A Primer*. Wiley, 2016. 2, 3

[49] Judea Pearl and Dana Mackenzie. *The Book of Why: The New Science of Cause and Effect*. 2018. 2

[50] Michael Puthawala, Konik Kothari, Matti Lassas, Ivan Dokmanić, and Maarten de Hoop. Globally injective relu networks. *arXiv preprint arXiv:2006.08464*, 2020. 12

[51] W. J. Scheirer, A. de Rezende Rocha, A. Sapkota, and T. E. Boult. Toward open set recognition. *TPAMI*, 2013. 3

[52] Walter J Scheirer, Lalit P Jain, and Terrance E Boult. Probability models for open set recognition. *TPAMI*, 2014. 1, 3

[53] Bernhard Schölkopf. Causality for machine learning. *arXiv preprint arXiv:1911.10500*, 2019. 1

[54] Edgar Schonfeld, Sayna Ebrahimi, Samarth Sinha, Trevor Darrell, and Zeynep Akata. Generalized zero-and few-shot learning via aligned variational autoencoders. In *CVPR*, 2019. 7, 13, 14

[55] Eli Schwartz, Leonid Karlinsky, Rogerio Feris, Raja Giryes, and Alex M Bronstein. Baby steps towards few-shot learning with multiple semantics. In *CVPR*, 2019. 3

[56] Xin Sun, Zhenning Yang, Chi Zhang, Keck-Voon Ling, and Guohao Peng. Conditional gaussian distribution learning for open set recognition. In *CVPR*, 2020. 3, 6, 8, 13, 14, 15, 16, 17

[57] Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip HS Torr, and Timothy M Hospedales. Learning to compare: Relation network for few-shot learning. In *CVPR*, 2018. 7, 14

[58] Raphael Suter, Djordje Miladinovic, Bernhard Schölkopf, and Stefan Bauer. Robustly disentangled causal mechanisms: Validating deep representations for interventional robustness. In *ICML*, 2019. 2, 5, 8

[59] Kaihua Tang, Jianqiang Huang, and Hanwang Zhang. Long-tailed classification by keeping the good and removing the bad momentum causal effect. In *Advances in Neural Information Processing Systems*, 2020. 3

[60] Tan Wang, Jianqiang Huang, Hanwang Zhang, and Qianru Sun. Visual commonsense r-cnn. In *CVPR*, 2020. 3

[61] P. Welinder, S. Branson, T. Mita, C. Wah, F. Schroff, S. Belongie, and P. Perona. Caltech-UCSD Birds 200. Technical report, California Institute of Technology, 2010. 2, 6, 7, 13, 14

[62] Yongqin Xian, Zeynep Akata, Gaurav Sharma, Quynh Nguyen, Matthias Hein, and Bernt Schiele. Latent embeddings for zero-shot classification. In *CVPR*, 2016. 7

[63] Yongqin Xian, H. Christoph Lampert, Bernt Schiele, and Zeynep Akata. Zero-shot learning - a comprehensive evaluation of the good, the bad and the ugly. *TPAMI*, 2018. 1, 3, 6, 7, 13, 14

[64] Yongqin Xian, H. Christoph Lampert, Bernt Schiele, and Zeynep Akata. Zero-shot learning - a comprehensive evaluation of the good, the bad and the ugly. 2020. 7

[65] Yongqin Xian, Tobias Lorenz, Bernt Schiele, and Zeynep Akata. Feature generating networks for zero-shot learning. In *CVPR*, 2018. 2, 3, 6

[66] Yongqin Xian, Saurabh Sharma, Bernt Schiele, and Zeynep Akata. f-vaegan-d2: A feature generating framework for any-shot learning. In *CVPR*, 2019. 1

[67] J. Xiao, J. Hays, K. A. Ehinger, A. Oliva, and A. Torralba. Sun database: Large-scale scene recognition from abbey to zoo. In *CVPR*, 2010. 6, 7, 13, 14

[68] Guo-Sen Xie, Li Liu, Xiaobo Jin, Fan Zhu, Zheng Zhang, Jie Qin, Yazhou Yao, and Ling Shao. Attentive region embedding network for zero-shot learning. In *CVPR*, 2019. 7

[69] Xu Yang, Hanwang Zhang, and Jianfei Cai. Deconfounded image captioning: A causal retrospect. *arXiv preprint arXiv:2003.03923*, 2020. 3

[70] Ryota Yoshihashi, Wen Shao, Rei Kawakami, Shaodi You, Makoto Iida, and Takeshi Naemura. Classification-reconstruction learning for open-set recognition. In *CVPR*, 2019. 6, 7

[71] Zhongqi Yue, Hanwang Zhang, Qianru Sun, and Xian-Sheng Hua. Interventional few-shot learning. In *NeurIPS*, 2020. 3

[72] Dong Zhang, Hanwang Zhang, Jinhui Tang, Xiansheng Hua, and Qianru Sun. Causal intervention for weakly-supervised semantic segmentation. *NeurIPS*, 2020. 3

[73] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *CVPR*, 2016. 8

[74] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *CVPR*, 2017. 5

[75] Sergey Demyanov Zongyuan Ge and Rahil Garnavi. Generative openmax for multi-class open set classification. In *BMVC*, 2017. 3

Appendix

This appendix is organized as follows:

- Section A.1 proves the theorem in Section 3.4 as a corollary of [7];
- Section A.2 gives the implementation details of our model in ZSL (Section A.2.1) and OSR (Section A.2.2);
- Section A.3 includes additional experimental results; Specifically, Section A.3.1 shows additional results on two-stage inference and a comparison between entangled and disentangled model in ZSL Accuracy. Section A.3.2 shows additional results for OSR, with the details of the 5 splits that we used and the F1 score for each split, closed-set accuracy, a comparison between entangled and disentangled model and more qualitative results.

A.1. Proof of the Theorem in 3.4

Let $V = (Z, Y)$ and V takes values from the space $\mathcal{V}$. For class attribute Y that is a K-dimensional real vector, denote $\mathcal{E} = \{e_1, \ldots, e_K\}$ as the subset of dimensions spanned by the class attribute Y, and $\bar{\mathcal{E}}$ as those by the sample attribute Z. Therefore, $\mathcal{V}^{\mathcal{E}}$ and $\mathcal{V}^{\bar{\mathcal{E}}}$ represent the space of class attribute Y and sample attribute Z, respectively. We use $g : \mathcal{V} \to \mathcal{X}$ to denote the endogenous mapping to the feature space $\mathcal{X}$. Note that this g corresponds to sampling from $P_\theta(X|Z, Y)$ in our GCM. We will introduce the concept of embedded GCM to facilitate the proof.

Definition (Embedded GCM). *We say that a GCM is embedded if $g : \mathcal{V} \to \mathcal{X}$ is a continuous injective function with continuous inversion g^{-1}.*

Using the results in [4], if V is compact (*i.e.*, bounded), the GCM is embedded if and only if g is injective. Though we implement our GCM using VAE, whose latent space is not compact, it is shown that restricting a VAE to a product of compact intervals that covers most of the probability mass (using KL-divergence in the objective) will result in an embedded GCM that approximates the original one for most samples [7]. Moreover, $P_\theta(X|Z, Y)$ are implemented using deterministic mappings in our model (see Section A.2.1, A.2.2), which are indeed injective as shown in [50]. Therefore, without loss of generality, our GCM can be considered as embedded. We will give the formal definition of intrinsic disentanglement, which can be used to show that group disentanglement of Z and Y leads to faithfulness.

Definition (Intrinsic Disentanglement). *In a GCM, the endomorphism $T : \mathcal{X} \to \mathcal{X}$ is intrinsically disentangled with respect to the subset $\mathcal{E}$ of endogenous variables, if there exists a transformation T' affecting only variables indexed by $\mathcal{E}$, such that for any $\mathbf{v} \in \mathcal{V}$,*

$$T(g(\mathbf{v})) = g(T'(\mathbf{v})). \tag{9}$$

Now we will first establish the equivalence between intrinsic disentanglement and faithfulness using the theorem below.

Theorem (Intrinsic Disentanglement and Faithfulness). *The counterfactual mapping $X_{\mathbf{y}}[z(X)]$ is faithful if and only if it is intrinsically disentangled with respect to the subset $\mathcal{E}$.*

To prove the above theorem, one conditional is trivial: if a transformation is intrinsically disentangled, 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 $X_{\mathbf{y}}[z(X)]$. Based on the three steps of computing counterfactuals and the embedding property discussed earlier, the counterfactual mapping can be decomposed as:

$$X_{\mathbf{y}}[z(X)] = g \circ T' \circ g^{-1}(X), \tag{10}$$

where $\circ$ denotes function composition, T' is the counterfactual transformation of V as defined in Section 3.2, where Z is kept as $Z = z(X)$ and Y is set as $\mathbf{y}$. Now for any $\mathbf{v} \in \mathcal{V}$, the quantity $X_{\mathbf{y}}[z(g(\mathbf{v}))]$ can be similarly decomposed as:

$$X_{\mathbf{y}}[z(g(\mathbf{v}))] = g \circ T' \circ g^{-1} \circ g(\mathbf{v}) = g \circ T'(\mathbf{v}). \tag{11}$$

Since T' is a transformation that only affects variables in $\mathcal{E}$ (*i.e.*, Y), we show that faithful counterfactual transformation $X_{\mathbf{y}}[z(X)]$ is intrinsically disentangled with respect to $\mathcal{E}$.

In our work, we define group disentanglement of Z and Y as intrinsic disentanglement with respect to the set of variables in Y. We used the sufficient condition, *i.e.*, learning a GCM where Y and Z are group disentangled, such that when we fix Z and only change Y, the resulting generation lies in $\mathcal{X}$ according to the theorem that we just proved.

A.2. Implementation Details

A.2.1. ZSL

Our GCM implementation is based on the generative models in TF-VAEGAN [41]. Besides $P_\theta(X|Z, Y)$ and $Q_\phi(Z|X)$ that is common in a VAE-GAN [32], it additionally implements $Q_\psi(Y|X)$ and a feedback module, which takes the intermediate layer output from the network in $Q_\psi(Y|X)$ as input and generate a feedback to assist the generation process $P_\theta(X|Z, Y)$. The rest of the section will detail the network architecture for each component, followed by additional training and inference details supplementary to Section 3.4 and 3.3, respectively.

Sample Attribute Z. The dimension of sample attribute Z is set to be the same as that of the class attribute for each

dataset. For example, in CUB [61], the dimension of Z is set as 312. $P(Z)$ is defined as $\mathcal{N}(\mathbf{0}, \mathbf{I})$ for all datasets.

Decoder $P_\theta(X|Z,Y)$. The module that implements this conditional distribution is commonly known as the decoder in literature [54, 41]. We implemented $P_\theta(X|Z,Y)$ with Deep Gaussian Family $\mathcal{N}(\mu_D(Z,Y), \mathbf{I})$ with its variance fixed and mean generated from a network μ_D. μ_D was implemented with a Multi-Layer Perceptron (MLP) with two layers and LeakyReLU activation (alpha=0.2) in between. The input to the MLP is the concatenated Z and Y. The hidden layer size is set as 4,096. The MLP outputs a real vector of size 2,048 (same as that of X) and the output goes through a Sigmoid activation to produce the mean of $P_\theta(X|Z,Y)$.

Encoder $Q_\phi(Z|X)$. For convenience, we refer to $Q_\phi(Z|X)$ as the encoder. We implemented $Q_\phi(Z|X)$ with $\mathcal{N}(\mu_E(X), \sigma_E^2(X))$, where $\mu_E(X), \sigma_E^2(X)$ are neural networks that share identical architecture. Specifically, they are 3-layer MLP with LeakyReLU activation (alpha=0.2) that takes X as input and outputs a vector with the same dimension as Z. The first hidden layer size is set as 4,096 and the second hidden layer size is set as two times of the dimension of Z. Note that in the original TF-VAEGAN implementation, the encoder additionally conditions on Y. We argue that this may cause the encoded Z to contain information about Y, undermining the disentanglement. Hence we make the encoder conditioned only on X.

Regressor $Q_\psi(Y|X)$. It is a 2-layer MLP with LeakyReLU activation (alpha=0.2). The hidden layer size is set as 4,096.

Discriminator $D(X,Y)$. It is a 2-layer MLP with LeakyReLU activation (alpha=0.2). It takes the concatenated X and Y as input and outputs a single real value. The hidden layer size is set as 4,096.

Feedback Module. It is a 2-layer MLP with LeakyReLU activation (alpha=0.2). The hidden layer output of the regressor is sent to the feedback module as input. This module generates a real vector with 4,096 dimensions, which is added to the hidden layer output of μ_D as feedback signal.

Training. The networks are trained in an iterative fashion. First, all networks except the decoder are optimized. Then the discriminator is frozen and all other networks are optimized. We followed the optimization settings in TF-VAEGAN. Specifically, the Adam [28] optimizer is used with learning rate set as $1e^{-4}$ in CUB [61], $1e^{-5}$ in AWA2 [63], $1e^{-3}$ in SUN [67] and $1e^{-5}$ in aPY [15]. For the hyperparameters in our counterfactual-faithful training, we used $\beta = 6.0, \nu = 1.0$ for CUB, $\beta = 6.0, \nu = 1.0$ for AWA2, $\beta = 4.0, \nu = 1.0$ for SUN and $\beta = 6.0, \nu = 0$ for aPY. On CUB, AWA2, and SUN, we additionally used annealing on the KL divergence loss, where β is initially set as 0 and linearly increased to the set value over 40 epochs. The parameter ρ is set to 1.0 in ZSL task.

Inference. In ZSL, we train a linear classifier with one fully-connected layer using the Adam optimizer. On CUB, the classifier was trained for 15 epochs with learning rate as $1e^{-3}$. On AWA2, it was trained for 3 epochs with learning rate as $1e^{-3}$. On SUN, it was trained for 6 epochs with learning rate as $5e^{-4}$. On aPY, it was trained for 3 epochs with learning rate as $1e^{-3}$. After training, the classifier was used for inference following the decision rule introduced in Section 3.3.

A.2.2. OSR

Our proposed GCM-CF is implemented based on the architecture of CGDL [56]. Notice that the original CGDL doesn't distinguish sample attribute Z and class attribute Y explicitly. To keep consistent with the ZSL model and follow the common VAE-GAN architecture, here we revise the encoder to model Z and Y respectively.

Encoder $Q_\phi(Z|X)$. Given an actual image $X = \boldsymbol{x}$, we follow [56] to implement $Q_\phi(Z|X)$ with the probabilistic ladder architecture to extract the high-level abstract latent features $\boldsymbol{z}$. In detail, the l-th layer in the ladder encoder is expressed as:

$$\begin{aligned}
\boldsymbol{x}_l &= \text{Conv}(\boldsymbol{x}_{l-1}) \\
\boldsymbol{h}_l &= \text{Flatten}(\boldsymbol{x}_l) \\
\boldsymbol{\mu}_l &= \text{Linear}(\boldsymbol{h}_l) \\
\boldsymbol{\sigma}_l^2 &= \text{Softplus}(\text{Linear}(\boldsymbol{h}_l))
\end{aligned}$$

where `Conv` is a convolutional layer followed by a batch-norm layer and a PReLU layer, `Flatten` is a linear layer to flatten 2-dimensional data into 1-dimension, `Linear` is a single linear layer and `Softplus` applies $\log(1+\exp(\cdot))$ non-linearity to each component of its argument vector. The latent representation $Z = \boldsymbol{z}$ can be obtained as:

$$\begin{aligned}
\boldsymbol{\mu_z}, \boldsymbol{\sigma_z^2} &= \text{LadderEncoder}(\boldsymbol{x}), \\
\boldsymbol{z} &= \mu_z + \sigma_z \odot \mathcal{N}(0, \mathbf{I}),
\end{aligned} \tag{12}$$

where $\odot$ is the element-wise product.

Decoder $P_\theta(X|Z,Y)$. Given the latent sample attribute $Z = \boldsymbol{z}$ and the class attribute $Y = \boldsymbol{y}$, the l-th layer of the ladder decoder is expressed as follows:

$$\begin{aligned}
\tilde{\boldsymbol{c}}_{l+1} &= \text{Unflatten}(\boldsymbol{t}_{l+1}) \\
\tilde{\boldsymbol{x}}_{l+1} &= \text{ConvT}(\tilde{\boldsymbol{c}}_{l+1}) \\
\tilde{\boldsymbol{h}}_{l+1} &= \text{Flatten}(\tilde{\boldsymbol{x}}_{l+1}) \\
\tilde{\boldsymbol{\mu}}_l &= \text{Linear}(\tilde{\boldsymbol{h}}_{l+1}) \\
\tilde{\boldsymbol{\sigma}}_l^2 &= \text{Softplus}(\text{Linear}(\tilde{\boldsymbol{h}}_{l+1})) \\
\boldsymbol{t}_l &= \tilde{\boldsymbol{\mu}}_l + \tilde{\boldsymbol{\sigma}}_l^2 \odot \boldsymbol{\epsilon}
\end{aligned}$$

where `ConvT` is a transposed convolutional layer and `Unflatten` is a linear layer to convert 1-dimensional data into 2-dimension. Note that the input $\boldsymbol{t}$ of the top decoder

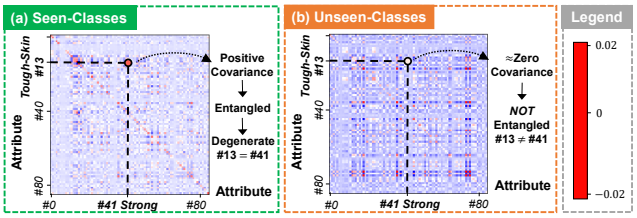

Figure A1: Visualization of the covariance matrix in AWA2 dataset of the (a) seen-classes and (b) unseen-classes attributes.

layer is the concatenation of $\boldsymbol{z}$ and $\boldsymbol{y}$. Overall, the reconstructed image $\tilde{\boldsymbol{x}}$ can be represented as:

$$\tilde{\boldsymbol{x}} = \text{LadderDecoder}(\boldsymbol{z}, \boldsymbol{y}). \tag{13}$$

For more details please refer to [56].

Known Classifier. The known classifier is a Softmax Layer taking the one-hot embedding $\boldsymbol{y}$ as the input and produces the probability distribution over the known classes.

Training. The network is trained in an end-to-end fashion. We directly follow the optimization settings and hyperparameters of ladder architecture in CGDL [56]. For the hyperparameters in our counterfactual-faithful training, we used $\beta = 1.0, \nu = 10.0$ for MNIST, $\beta = 1.0, \nu = 2.0$ for SVHN, $\beta = 6.0, \nu = 1.0$ for CIFAR10, $\beta = 1.0, \nu = 20.0$ for CIFARAdd10, $\beta = 1.0, \nu = 1.0$ for CIFARAdd50. Note that the ρ is set to 0 in OSR task.

Inference. When training is completed, we follow [56] to use the reconstruction errors and a multivariate Gaussian model to judge the unseen samples. The threshold τ_l is set to 0.9 for MNIST, 0.6 for SVHN, 0.9 for CIFAR10, 0.8 for CIFARAdd10 and 0.5 for CIFARAdd50. More details about the multivariate Gaussian model please refer to [56].

A.3. Additional Results

A.3.1. ZSL

Entanglement of Y. Two class attributes are entangled when they always or never appear together, and they effectively degenerate to one feature. If two attributes are entangled among seen-classes but not unseen, the degenerated feature learnt from seen cannot tell unseen-classes apart. Entanglement can be identified by finding pairs of attributes with large absolute covariance. For example, Figure A1 shows a "strong" animal usually has "tough skin" among seen-classes. Yet the co-appearance no longer holds for unseen-classes as shown in Figure A1.

Stage-One Binary Classifier. We extend the results in Table 5 by showing comparison of the two-stage inference performance on CUB [61], SUN [67] and aPY [15] dataset. Our GCM-CF improves all of them and outperforms the current SOTA ZSL method TF-VAEGAN as a binary unseen/seen classifier.

Dataset	CUB [61]					
Stage 1 / Stage 2	**TF-VAEGAN [41]**			**GCM-CF (Ours)**		
	U	S	H	U	S	H
RelationNet [57]	40.5	65.3	50.0	47.7	57.6	**52.2**
CADA-VAE [54]	43.2	63.4	51.4	51.4	57.6	**54.3**
LisGAN [34]	41.1	66.0	50.7	47.9	58.1	**52.5**
TF-VAEGAN [41]	50.8	64.0	56.6	55.4	60.0	**57.6**
Dataset	SUN [67]					
Stage 1 / Stage 2	**TF-VAEGAN [41]**			**GCM-CF (Ours)**		
	U	S	H	U	S	H
RelationNet [57]	30.8	23.0	26.3	37.2	21.9	**27.6**
CADA-VAE [54]	37.6	39.3	38.4	44.6	37.6	**40.8**
LisGAN [34]	36.3	41.7	38.8	43.0	38.9	**40.8**
TF-VAEGAN [41]	41.7	39.9	40.8	47.9	37.8	**42.2**
Dataset	aPY [15]					
Stage 1 / Stage 2	**TF-VAEGAN [41]**			**GCM-CF (Ours)**		
	U	S	H	U	S	H
RelationNet [57]	31.5	63.3	42.1	34.6	56.6	**43.0**
CADA-VAE [54]	31.1	64.8	41.9	35.0	57.2	**43.5**
LisGAN [34]	31.2	64.6	42.0	34.7	57.3	**43.2**
TF-VAEGAN [41]	33.1	64.2	43.7	37.1	56.8	**44.9**

Table A1: Supplementary to Table 5. Comparison of the two-stage inference performance on CUB [61], SUN [67] and aPY [15] using TF-VAEGAN [41] and our GCM-CF as the stage-one binary classifier.

Dataset	**Entangled**			**Disentangled**		
	U	S	H	U	S	H
CUB [61]	71.6	15.8	25.9	61.0	59.7	**60.3**
AWA2 [63]	70.5	27.9	40.0	60.4	75.1	**67.0**
SUN [67]	62.9	17.5	27.4	47.9	37.8	**42.2**
aPY [15]	42.4	14.4	21.5	37.1	56.8	**44.9**

Table A2: Comparison of ZSL Accuracy using an entangled model without using the proposed counterfactual-faithful training and the disentangled model with the proposed training.

Effect of Disentanglement. To show that the quality of disentangling Z and Y is the key bottleneck, we compared an entangled model (without counterfactual-faithful training) and the disentangled model on ZSL Accuracy and the results are shown in Table A2. Notice that the entangled model has a much lower S. This is because the training is conducted on the seen-classes and the encoded Z is entangled with seen-classes attributes. Therefore the generated counterfactuals are biased towards the seen-classes, *i.e.*, the green counterfactual samples in Figure 2c are closer to true samples from seen-classes, pushing the classifier boundary towards seen-classes and increasing the recall of the unseen-class by sacrificing that of the seen.

A.3.2. OSR

5 Splits Results. In our main paper we have argued that the common evaluation setting of averaging F1 scores over 5 random splits can result in a large variance in the F1 score. Here we additionally report the split details in Table A3 and the results on all splits in Tabel A4. Note that since the official code of CGDL [56] is not complete, we implemented

the code of dataloader, computing F1 score and set part of parameters. The 5 seeds (*i.e.*, 5 splits) are randomly chosen without any selection.

Split	1 (seed: 777)	
Dataset	Seen	Unseen
MNIST	3,7,8,2,4,6	0,1,5,9
SVHN	3,7,8,2,4,6	0,1,5,9
CIFAR10	3,7,8,2,4,6	0,1,5,9
CIFARAdd10	0,1,8,9	27, 46, 98, 38, 72, 31, 36, 66, 3, 97
CIFARAdd50	0,1,8,9	27, 46, 98, 38, 72, 31, 36, 66, 3, 97, 75, 67, 42, 32, 14, 93, 6, 88, 11, 1, 44, 35, 73, 19, 18, 78, 15, 4, 50, 65, 64, 55, 30, 80, 26, 2, 7, 34, 79, 43, 74, 29, 45, 91, 37, 99, 95, 63, 24, 21
Split	2 (seed: 1234)	
Dataset	Seen	Unseen
MNIST	7,1,0,9,4,6	2,3,5,8
SVHN	7,1,0,9,4,6	2,3,5,8
CIFAR10	7,1,0,9,4,6	2,3,5,8
CIFARAdd10	0,1,8,9	98, 46, 14, 1, 7, 73, 3, 79, 93, 11
CIFARAdd50	0,1,8,9	98, 46, 14, 1, 7, 73, 3, 79, 93, 11, 37, 29, 2, 74, 91, 77, 55, 50, 18, 80, 63, 67, 4, 45, 95, 30, 75, 97, 88, 36, 31, 27, 65, 32, 43, 72, 6, 26, 15, 42, 19, 34, 38, 66, 35, 21, 24, 99, 78, 44
Split	3 (seed: 2731)	
Dataset	Seen	Unseen
MNIST	8,1,6,7,2,4	0,3,5,9
SVHN	8,1,6,7,2,4	0,3,5,9
CIFAR10	8,1,6,7,2,4	0,3,5,9
CIFARAdd10	0,1,8,9	79, 98, 67, 7, 77, 42, 36, 65, 26, 64
CIFARAdd50	0,1,8,9	79, 98, 67, 7, 77, 42, 36, 65, 26, 64, 66, 73, 75, 3, 32, 14, 35, 6, 24, 21, 55, 34, 30, 43, 93, 38, 19, 99, 72, 97, 78, 18, 31, 63, 29, 74, 91, 4, 27, 46, 2, 88, 45, 15, 11, 1, 95, 50, 80, 44
Split	4 (seed: 3925)	
Dataset	Seen	Unseen
MNIST	7,3,8,4,6,1	0,2,5,9
SVHN	7,3,8,4,6,1	0,2,5,9
CIFAR10	7,3,8,4,6,1	0,2,5,9
CIFARAdd10	0,1,8,9	46, 77, 29, 24, 65, 66, 79, 21, 1, 95
CIFARAdd50	0,1,8,9	46, 77, 29, 24, 65, 66, 79, 21, 1, 95, 36, 88, 27, 99, 67, 19, 75, 42, 2, 73, 32, 98, 72, 97, 78, 11, 14, 74, 50, 37, 26, 64, 44, 30, 31, 18, 38, 4, 35, 80, 45, 63, 93, 34, 3, 43, 6, 55, 91, 15
Split	5 (seed: 5432)	
Dataset	Seen	Unseen
MNIST	2,8,7,3,5,1	0,4,6,9
SVHN	2,8,7,3,5,1	0,4,6,9
CIFAR10	2,8,7,3,5,1	0,4,6,9
CIFARAdd10	0,1,8,9	21, 95, 64, 55, 50, 24, 93, 75, 27, 36
CIFARAdd50	0,1,8,9	21, 95, 64, 55, 50, 24, 93, 75, 27, 36, 73, 63, 19, 98, 46, 1, 15, 72, 42, 78, 77, 29, 74, 30, 14, 38, 80, 45, 4, 26, 31, 11, 97, 7, 66, 65, 99, 34, 6, 18, 44, 3, 35, 88, 43, 91, 32, 67, 37, 79

Table A3: The detailed label splits of 5 random seeds

Closed Set Results. Closed-Set Accuracy is the standard supervised classification accuracy on the seen-classes with open set detection disabled. As shown in Table A5, the network were trained without any large degradation in closed set accuracy from the plain CNN.

Split			1		
Method	MNIST	SVHN	CIFAR10	C+10	C+50
Softmax	76.26	73.06	69.81	77.87	65.78
OpenMax [6]	83.34	75.34	71.49	78.70	67.27
CGDL [56]	91.79	77.42	70.02	78.52	72.7
GCM-CF (Ours)	**94.21**	**79.23**	**73.03**	**80.29**	**74.70**
Split			2		
Method	MNIST	SVHN	CIFAR10	C+10	C+50
Softmax	77.06	75.03	73.02	77.82	65.87
OpenMax [6]	86.97	77.27	73.74	79.02	67.56
CGDL [56]	86.76	73.63	73.15	76.46	70.79
GCM-CF (Ours)	**91.82**	**80.28**	**75.71**	**79.67**	**74.79**
Split			3		
Method	MNIST	SVHN	CIFAR10	C+10	C+50
Softmax	77.44	78.67	70.79	77.61	66.21
OpenMax [6]	83.39	80.00	72.01	78.38	67.83
CGDL [56]	92.36	77.59	74.77	77.92	71.93
GCM-CF (Ours)	**93.86**	**80.51**	**75.38**	**79.40**	**76.56**
Split			4		
Method	MNIST	SVHN	CIFAR10	C+10	C+50
Softmax	76.03	75.47	70.25	77.67	66.01
OpenMax [6]	87.06	76.80	70.76	78.64	68.21
CGDL [56]	90.34	73.53	70.30	78.27	71.69
GCM-CF (Ours)	**91.34**	**80.76**	**71.12**	**78.74**	**74.53**
Split			5		
Method	MNIST	SVHN	CIFAR10	C+10	C+50
Softmax	77.33	78.30	68.12	78.13	65.97
OpenMax [6]	89.88	80.33	68.90	78.70	67.55
CGDL [56]	83.51	79.41	66.89	78.00	68.18
GCM-CF (Ours)	**92.45**	**80.33**	**69.52**	**78.79**	**72.40**

Table A4: Comparison of the F1 score averaged over 5 random splits in OSR. Note that since the official code of CGDL [56] is not complete, we implemented the code of dataloader, F1 score and set part of parameters. Moreover, we also implemented Softmax and OpenMax [6] for evaluation. For GCM-CF, after binary classification, we used CGDL for supervised classification on the seen-classes.

Method	MNIST	SVHN	CIFAR10	C+10	C+50
Plain CNN	0.995	0.965	0.917	0.941	0.940
CGDL [56]	0.996	0.962	0.913	0.934	0.935

Table A5: Comparison of the Closed-Set accuracy in OSR.

Effect of disentanglement. To further demonstrate the effectiveness of disentangling Z and Y in OSR, we also compared an entangled model (without the counterfactual-faithful training) and the disentangled model on F1 scores. The results are shown in Table A6. Similar to the ZSL, we can also see the F1 scores of entangled model are much lower than those of disentangled model. The constructed green counterfactual samples are still closer to the unseen sample though given the seen attributes without disentanglement, which demonstrate the necessity of the proposed disentangle loss.

More Qualitative Results. Figure A2 and A3 show the additional qualitative results comparing existing reconstruction-based approach with our proposed counterfactual approach on MNIST and SVHN dataset. For the

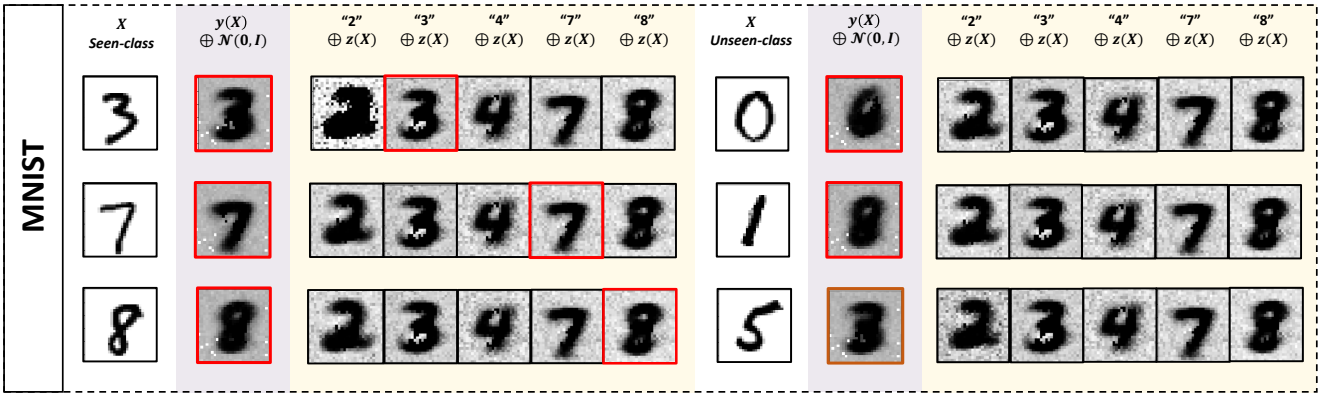

Figure A2: The additional qualitative results of the reconstructed images using CGDL [56] ($y(X) \oplus \mathcal{N}(0, I)$) and the counterfactual images generated from our GCM-CF ($y \oplus z(X)$) on MNIST dataset. The red box on a generated image denotes that it is similar to the original, while the brown box represents the failure generation.

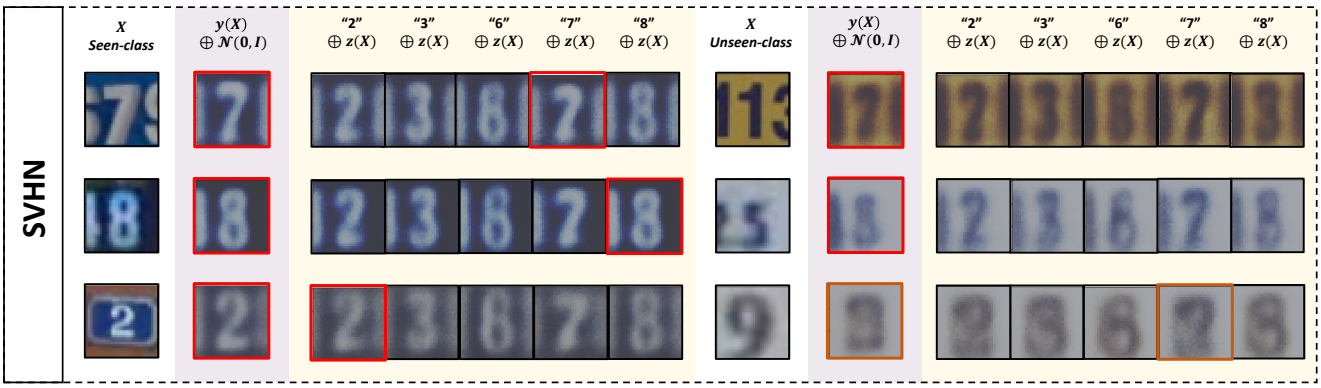

Figure A3: The additional qualitative results of the reconstructed images using CGDL [56] ($y(X) \oplus \mathcal{N}(0, I)$) and the counterfactual images generated from our GCM-CF ($y \oplus z(X)$) on SVHN dataset. The red box on a generated image denotes that it is similar to the original, while the brown box represents the failure generation.

Model	MNIST	SVHN	CIFAR10	C+10	C+50
Entangled	91.37	62.57	67.03	73.81	69.18
Disentangled	**94.21**	**79.23**	**73.03**	**80.29**	**74.70**

Table A6: Comparison of the F1 scores using entangeled and disentangled model in OSR.

Seen-class (*i.e.*, the left part), both the baseline model and our GCM-CF can reconstruct samples with low reconstruction error, which means both of them can handle well given the seen-class images. When coming to the unseen-class (*i.e.*, the right part), the baseline method would still generate similar samples (red box), with a much lower reconstruction error comparing to the counterfactual samples produced by GCM-CF, resulting in a failure rejection to the unknown outlier. The brown box represents the failure reconstructions for the baseline model (*i.e.*, generated sample is also dissimilar with the original input image) given some unseen-class samples. Note that this is reasonable since the model haven't seen the unseen-class samples during training, which also corresponds to Figure 4b in the main paper. In this case, our counterfactual model can still make better generation (*e.g.*, "3" in the last row of Figure A2).

For the CIFAR10 dataset, as dicussed in the main paper, we cannot generate realistic images due to the conflict between visual perception and discriminative training. Therefore, we apply a pretrained image classifier to generate CAM to reveal the sensible yet discriminating features. Here we show the additional examples in Figure A4. The first row is the direct image reconstruction results generated by baseline and proposed model. The disordered appearance explains that the pixel-level generation is not sensible. However, when utilizing the tool of CAM, something magical happened. The insensible pixel-level generation becomes discriminative in the view of the pretrained classifier. The generation samples of our proposed GCM-CF reveal different heat maps given different counterfactual class conditions. Among them the heat map of "ship" condition is quite similar to that of the reconstruction of the baseline model (red box), showing the consistency of the CAM heat map. Moreover, for different samples in the same class (*i.e.*, the different rows), the class-specific CAM heat maps keep

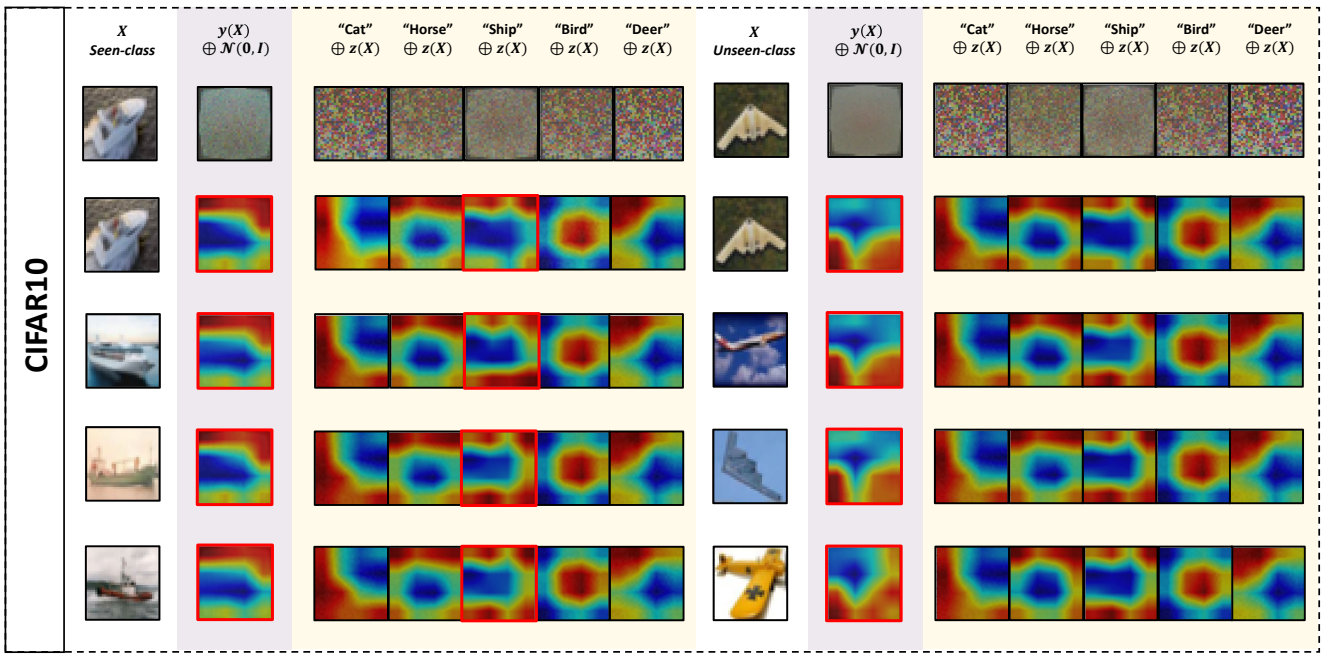

Figure A4: The additional qualitative results of the reconstructed images using CGDL [56] ($y(X) \oplus \mathcal{N}(0, I)$) and the counterfactual images generated from our GCM-CF ($y \oplus z(X)$) on CIFAR10 dataset. The red box on a generated image denotes that it is similar to the original.

stable with only minor changes. It further demonstrates that the CAM heat map can be considered a kind of substitution of the original pixel images to reveal the discrimitative feature.