Instance-Dependent Partial Label Learning with Identifiable Causal Representations

Yizhi Wang[1,2], **Weijia Zhang**[3], **Min-Ling Zhang**[1,2]
[1] School of Computer Science and Engineering, Southeast University, Nanjing 210096, China
[2] Key Laboratory of Computer Network and Information Integration (Southeast University), Ministry of Education, China
[3] School of Information and Physical Sciences, The University of Newcastle, Callaghan, NSW 2308, Australia
wang_yz@seu.edu.cn, weijia.zhang@newcastle.edu.au, zhangml@seu.edu.cn

Abstract

Partial label learning (PLL) deals with the problem where each training example is annotated with a set of candidate labels, among which only one is true. In real-world scenarios, the candidate labels are generally dependent to the instance features. However, existing PLL methods focus solely on classification accuracy, whereas the possibility of exploiting the dependency for causal representation learning remains unexplored. In this paper, we investigate learning causal representations under the PLL paradigm and propose a novel framework which learns identifiable latent factors up to permutation, scaling and translation. Qualitative and quantitative experiments confirmed the effectiveness of this approach.

1 Introduction

Causal representation learning (CRL) [14] aims to identify latent variables from high-dimensional observations. A core task in CRL is learning identifiable latent representation, i.e., developing representation learning algorithms that can provably identify high-level latent factors such as shape, location, and colour of an object. As previous work has shown that causal representation identification is impossible for arbitrary data-generating process in an unsupervised fashion [9], much of the recent efforts have been diverted to learning causal representation from data with additional structures and supervisions [6, 7]. For example, recent studies have delved into understanding causal representations with interventions [1, 8] or under specific weak supervision signals [24, 20, 2].

In this paper, we investigate the possibility for identifying causal representation within the context of Partial Label Learning (PLL) paradigm, a form of weakly-supervised learning that has garnered significant attention over the past decade. Unlike traditional supervised learning where each instance is associated with a single class label, each training instance in PLL is annotated with a set of candidate labels, among which only one is the ground-truth. This problem naturally emerges in various real-world scenarios, such as web mining [10, 13], multimedia content analysis [21, 5], and automatic image annotations [3, 15, 16].

In PLL, the candidate labels of a sample typically correlate to the contents and styles of an instance, e.g., crowd-sourced annotators often output several possible labels for an image based on the context. Researchers have coined this concept as Instance-Dependent Partial Label Learning, i.e., the candidate labels are dependent to the instance features [19]. Characterizing this relationship is beneficial for training effective PLL models and makes partial labeling learning particular interesting for real-world scenarios.

Preprint. Under review.

However, most of the existing PLL methods are solely designed for classification, with a focus on how to recover true labels from ambiguous supervisory information. The causal relationship between candidate set, instance and the ground-truth label still remain mostly unexplored. In this paper, we will explore the possibility for learning causal representations by exploiting the intrinsic relationships among candidate labels and the instance within the PLL paradigm.

Our contributions can be summarized as follows:

- We propose a novel VAE framework with mixture priors for learning latent causal models from partial labels, and show that latent factors learned can be identifiable up to permutation, scaling and translation.
- We propose CAUSALPLL, which instantiate the above framework by utilizing a latent space consistency regularization loss to effectively learn causal representations from partial labels.
- We demonstrate the effectiveness of CAUSALPLL for causal representation learning and classification, with an improved validation protocol that better simulates candidate label generation in real-world scenarios.

2 Partial Label Learning and Identifiable Causal Representations

2.1 Partial Label Learning

Partial Label Learning (PLL) is a subfield of weakly supervised learning that learns from candidate label sets which contains one ground truth label and several false positive labels. PLL can be dated back to [4] and has recently drawn a lot of attention [18, 23, 22].

Early PLL algorithms often assume that the false positive labels of an instance are randomly chosen from the label space. However, in real-world applications, candidate labels and the instance are often dependent [19]. Taking the MNIST dataset as an example, crowd labellers are more likely to include the label "7" for digit "1" in the candidate label set. Consequently, building on this insight, the concept of Instance-Dependent Partial Label Learning [19] was introduced.

Instance-dependent candidate labels carry information that benefit learning latent representations. Contrasting to traditional supervised learning where the ground truth label only provide the content information, candidate labels in PLL provides additional information on the styles.

2.2 VAE and Identifiable Causal Representations

Variational Autoencoder (VAE) is a kind of deep generative model that combines neural networks with variational inference. It uses an encoder and an decoder, respectively, to fit the posterior and likelihood of the data. VAEs can be trained by maximizing the Evidence Lower **B**Ound (ELBO):

$$b_{\mathrm{ELBO}} = \mathbb{E}_{\boldsymbol{z}\sim q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})}\left[\ln p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})\right] - \mathrm{KL}(q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})||p_{\boldsymbol{\theta}}(\boldsymbol{z})) \tag{1}$$

VAEs are not only powerful generative models but have also been the subject of extensive research in terms of their identifiability. Identifiability in VAEs refers to the uniqueness of the learned latent representations for different data points. In other words, it addresses the question of whether two different data samples will have distinct latent representations.

Previous studies have shown that VAEs with unconditional prior distributions $p_{\boldsymbol{\theta}}(\boldsymbol{z})$ are generally not identifiable [9]. However, with a conditionally factorized prior distribution $p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{u})$, the latent factors $\boldsymbol{z}$ can be identified [6]. The iVAE [6] takes explicit observed variables to satisfy the identifiability condition. Recently, [7] have shown with a mixture prior, latent factors $\boldsymbol{z}$ can be identified without explicit observed variables, which is easier to apply in practical situations.

3 Methodology

Let $\mathcal{X} \subset \mathbb{R}^D$ denote the D-dimensional instance space and $\mathcal{Y} = \{1, 2, \cdots, C\}$ denote the label space with C distinct labels. $\mathcal{Z} \subset \mathbb{R}^M$ is the M-dimensional latent space where $M \ll D$. PLL assumes that the ground-truth label $y \in \mathcal{Y}$ of an instance $\boldsymbol{x} \in \mathcal{X}$ is contained within a candidate label set $\mathcal{S} \subset \mathcal{Y}$. For simplicity, we use the Boolean vector $\boldsymbol{s} \in \{0, 1\}^C$ to represent the partial label

corresponding to the candidate label set $\mathcal{S}$. The goal of PLL is to learn a classifier $h : \mathcal{Z} \to \mathcal{Y}$ on a partial label dataset $\mathcal{D} = \{(\boldsymbol{x}_i, \boldsymbol{s}_i)|1 \leq i \leq N\}$. For the classifier h, we use $h_c(\boldsymbol{z})$ to denote the output of classifier h on label c given input $\boldsymbol{z}$.

3.1 Identifiable Variational Auto-encoder

Although strict identifiability is difficult to achieve, identification of latent factors is often easier to be accomplished up to certain transformations. In our work, we focus on identifiability up to permutation, scaling and translation transformations. Recent results have shown that under the mixture prior assumption, latent factors $\boldsymbol{z}$ can be identified with a conditionally factorized prior distribution $p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{u})$, where $\boldsymbol{u}$ represents a specific component of the mixture [7]. Inspired by these results, we design a novel VAE-based PLL framework with mixture prior that is identifiable up to permutation, scaling and translation.

Generally speaking, a VAE-based PLL framework that learns identifiable representations should satisfy the following desiderata [7]:

1. The latent prior $p(\boldsymbol{z})$ is a (possibly degenerate) Gaussian mixture model with an unknown number of components $K \geq 1$, i.e.

$$p(\boldsymbol{z}) = \sum_{k=1}^{K} \lambda_k \varphi(\boldsymbol{z}; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k), \quad \sum_{k=1}^{K} \lambda_k = 1, \quad \lambda_k \geq 0,$$

   where $p(\boldsymbol{z})$ is the density of the prior with respect to some base measure, and $\varphi(\boldsymbol{z}; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$ is the gaussian density with mean $\boldsymbol{\mu}_k$ and covariance $\boldsymbol{\Sigma}_k$).
2. The decoder f is piecewise affine, e.g., a multilayer perceptron with ReLU activations.
3. $Z_i \perp\!\!\!\perp Z_j \mid U$ for all $i \neq j$ and there exist a pair of states $U = u_1$ and $U = u_2$ such that all $\frac{(\Sigma_{u_1})_{tt}}{(\Sigma_{u_2})_{tt}}$, $t \in \{1, 2, \cdots, M\}$ are distinct. (Z_i denotes the i-th entry of latent factors $\boldsymbol{Z}$)
4. f is injective, that is, for every y in the range of f, $|f^{-1}(y)| = 1$.

These conditions outline the design considerations for our VAE-based framework in causal representation learning from candidate label sets. For condition 1, we adopt VampPrior [17] as the latent prior which not only satisfies the requirement of mixed Gaussian priors, but also facilitates model learning; condition 2 can be satisfied by restricting the activation function of the neural network to ReLU or LeakyReLU; for condition 4, [12] have shown that a VAE with "gradually shrinking and gradually expanding" structure satisfying $D/M \geq 10.5$, f would be injective with high probabilities. (M is the input dimensionality of decoder f, and D is the output dimensionality.)

In order to satisfy condition 3, we can regularize the variances of the prior distribution as:

$$\Omega_{1,2;i,j} = -\left\| \frac{(\boldsymbol{\Sigma}_{u_1})_{ii}}{(\boldsymbol{\Sigma}_{u_2})_{ii}} - \frac{(\boldsymbol{\Sigma}_{u_1})_{jj}}{(\boldsymbol{\Sigma}_{u_2})_{jj}} \right\|_2^2 \tag{2}$$

The term above should be summed pairwisely over K and M, and comprehensive implementation details will be given in the supplementary materials (section 6.3). By maximizing the distance between different variance ratios, we can make them distinct from each other, and condition 3 would be satisfied.

3.2 CausalPLL

The overall framework for CAUSALPLL is shown in Figure 1 (left). Following the variational inference paradigm, the generation and inference process of CAUSALPLL can be written as follows:

$$p(\boldsymbol{x}, y, \boldsymbol{z}_1, \boldsymbol{z}_2, \boldsymbol{s}) = p(\boldsymbol{s}|\boldsymbol{x}, y) \cdot p(\boldsymbol{x}|\boldsymbol{z}_1, y) \cdot p(\boldsymbol{z}_1|\boldsymbol{z}_2) \cdot p(\boldsymbol{z}_2) \tag{3}$$

$$q(\boldsymbol{z}_2, \boldsymbol{z}_1, y|\boldsymbol{x}, \boldsymbol{s}) = q(\boldsymbol{z}_1|\boldsymbol{z}_2, y) \cdot q(y|\boldsymbol{z}_2, \boldsymbol{s}) \cdot q(\boldsymbol{z}_2|\boldsymbol{x}) \tag{4}$$

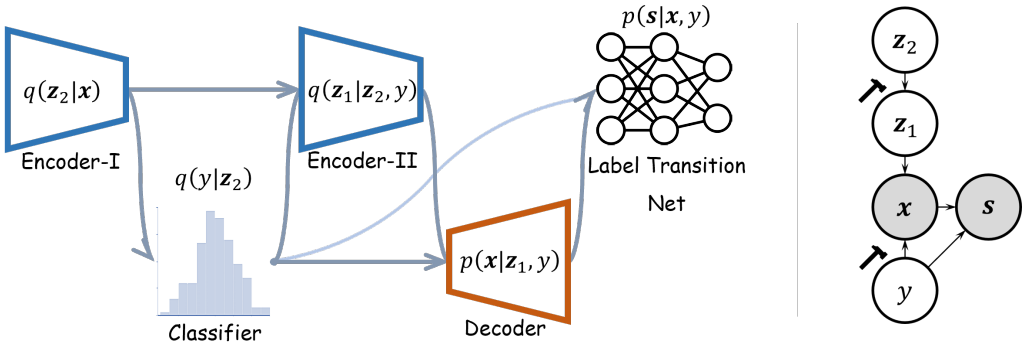

Figure 1: (Left) The framework of CAUSALPLL. (Right) The causal graph of the generation process. The "hammer" icons represent interventions which will be applied in the qualitative experiments.

Figure 1 (right) illustrate the data generation process under the PLL settings. Latent variables $\boldsymbol{z}_1$ and $\boldsymbol{z}_2$ could be considered as low-dimensional causal representations of the original instance. Meanwhile, since the ground-truth must be included in the candidate label set in PLL, y also makes a difference in the generation process of $\boldsymbol{s}$. Consequently, we have $p(\boldsymbol{s}|\boldsymbol{x}, y)$.

Specifically, our model contains two encoder networks which model distributions $q(\boldsymbol{z}_1|\boldsymbol{z}_2, y)$, $q(\boldsymbol{z}_2|\boldsymbol{x})$ in series and a decoder network which models the likelihood $p(\boldsymbol{x}|\boldsymbol{z}_1, y)$. Besides, the classifier and label transition network can also be considered as an encoder and a decoder, respectively. The former is used to model $q(y|\boldsymbol{z}_2, \boldsymbol{s})$, while the latter is used to fit $p(\boldsymbol{s}|\boldsymbol{x}, y)$. Additionally, $\boldsymbol{s}$ is implicitly provided in the loss function, consequently resulting in the classifier $q(y|\boldsymbol{z}_2, \boldsymbol{s})$ simplifying to $q(y|\boldsymbol{z}_2)$. According to the conditional independence relationship implied in the factorization above, we can simplify the variational posterior and derive the ELBO as follows. For detailed assumptions and derivations, please refer to supplementary materials (section 6.1).

$$\begin{aligned} b_{\text{ELBO}} = &\mathbb{E}_{\boldsymbol{z}_1, y\sim q(\boldsymbol{z}_1, y|\boldsymbol{z}_2)}[\ln p(\boldsymbol{x}|\boldsymbol{z}_1, y)] + \mathbb{E}_{y\sim q(y|\boldsymbol{z}_2)}[p(\boldsymbol{s}|\boldsymbol{x}, y)] - \text{KL}(q(\boldsymbol{z}_2|\boldsymbol{x})||p(\boldsymbol{z}_2)) \\ &- \mathbb{E}_{y\sim q(y|\boldsymbol{z}_2)}[\text{KL}(q(y|\boldsymbol{z}_2)||p(y))] - \mathbb{E}_{\boldsymbol{z}_2, y\sim q(\boldsymbol{z}_2, y|\boldsymbol{x})}[\text{KL}(q(\boldsymbol{z}_1|\boldsymbol{z}_2, y)||p(\boldsymbol{z}_1|\boldsymbol{z}_2))] \end{aligned} \tag{5}$$

In Formula 5, the first two terms of ELBO are reconstruction errors and the last three are KL divergence or its expectation. The fourth term $\mathbb{E}_{y\sim q(y|\boldsymbol{z}_2)}[\text{KL}(q(y|\boldsymbol{z}_2)||p(y))]$ can be seen as the supervision on the classifier which could be converted into other forms:

$$L(\boldsymbol{h}(\boldsymbol{z}_2), \boldsymbol{s}) + \lambda\Psi(\boldsymbol{h}(\boldsymbol{z}_2), \boldsymbol{s}) = \sum_{c=1}^{C}(1 - s_c) \cdot \ln(1 - h_c(\boldsymbol{z}_2)) + \lambda \sum_{\tilde{\boldsymbol{z}}_2 \in \mathcal{R}_{\boldsymbol{z}_2}} \text{KL}(\boldsymbol{p}||\boldsymbol{h}(\tilde{\boldsymbol{z}}_2)) \tag{6}$$

$\mathcal{R}_{\boldsymbol{z}_2} = \{\tilde{\boldsymbol{z}}_{2,1}, \tilde{\boldsymbol{z}}_{2,2}, \cdots, \tilde{\boldsymbol{z}}_{2,R}\}$ is the set of perturbed latent codes. For the supervision, we were inspired by [18] and adopted a modified version of consistency regularization loss. Original consistency regularization loss relies on semantic-preserving data augmentation, which is operated on the instance space. Meanwhile, semantic-preserving data augmentation is generally very limited in number and varies across different domains (CV, NLP, etc.) which limits its large-scale application. But if the perturbation could be applied to the data on high-level feature space, limitations mentioned above would not exist any longer. And this aligns precisely with the characteristics of VAE-based causal representation learning. To make predictions of $\mathcal{R}_{\boldsymbol{z}_2}$ similar to a conformal distribution $\boldsymbol{p}$, we could utilize the manifold structure in feature space and carried out PLL more effectively.

$$\begin{aligned} \mathcal{L} = &\text{BCE}(\boldsymbol{x}, \hat{\boldsymbol{x}}) + \text{BCE}(\boldsymbol{s}, \hat{\boldsymbol{s}}) + \text{KL}\Big(q(\boldsymbol{z}_1|\boldsymbol{z}_2, y)\Big|\Big|\mathcal{N}(\boldsymbol{0}, \boldsymbol{E})\Big) \\ &+ L(\hat{y}, \boldsymbol{s}) + \Psi(\hat{y}, \boldsymbol{s}) + \text{KL}\Big(q(\boldsymbol{z}_2|\boldsymbol{x})\Big|\Big|p(\boldsymbol{z}_2)\Big) + \Omega\Big(p(\boldsymbol{z}_2)\Big) \end{aligned} \tag{7}$$

From the derivations above, we could obtain the loss function (Formula 7). By minimizing reconstruction errors on $\boldsymbol{s}$, the label transition network captures the intricate relationships among $\boldsymbol{x}$, y and $\boldsymbol{s}$, thereby directing the model towards more effective classification.

Table 1: Accuracy (mean±std) comparisons on Fashion-MNIST, Kuzushiji-MNIST and MNIST with instance-dependent partial labels on different ambiguity levels.

Dataset	Method	$\tau = 16$	$\tau = 32$	$\tau = 64$
Fashion-MNIST	CausalPLL	$87.55 \pm 0.13\%$	$86.91 \pm 0.13\%$	$86.14 \pm 0.11\%$
	PRODEN	$87.32 \pm 0.19\%$	$86.34 \pm 0.08\%$	$85.15 \pm 0.24\%$
	VALEN	$88.36 \pm 0.20\%$	$87.25 \pm 0.19\%$	$85.67 \pm 0.24\%$
	Fully Supervised		$93.92 \pm 0.07\%$	
KMNIST	CausalPLL	$88.13 \pm 0.26\%$	$87.19 \pm 0.16\%$	$86.47 \pm 0.26\%$
	PRODEN	$88.50 \pm 0.24\%$	$86.27 \pm 0.33\%$	$82.92 \pm 0.45\%$
	VALEN	$86.08 \pm 0.37\%$	$82.23 \pm 0.36\%$	$77.18 \pm 0.56\%$
	Fully Supervised		98.31 ± 0.05	
MNIST	CausalPLL	$97.37 \pm 0.08\%$	$97.07 \pm 0.08\%$	$96.86 \pm 0.09\%$
	PRODEN	$98.22 \pm 0.06\%$	$98.09 \pm 0.04\%$	$97.86 \pm 0.04\%$
	VALEN	$97.96 \pm 0.05\%$	$96.98 \pm 0.11\%$	$95.57 \pm 0.09\%$
	Fully Supervised		98.32 ± 0.02	

4 Experiments

In this section, we first propose a novel method of generating instance-dependent PLL data from fully-supervised datasets, and then we evaluate the prediction performances of CausalPLL against PLL baselines, including PRODEN [11] and VALEN [19].

4.1 The Method of Generating Instance-Dependent PLL Data from Fully-Supervised Datasets

In previous partial label learning researches, a common practice is to manually corrupting the existing fully-supervised datasets into partially labeled versions by using a flipping probability [19]. However, the current candidate label generation strategies are not suitable for evaluation under the setting of causal representation learning and suffer from issues such as small flipping probabilities, confidence contradictions, and the lack of direct control over the difficulty of the task. To address these issues, we propose a new approach in this paper

Our approach first train a neural network with clean labels and generate candidate labels using temperature-adjusted flipping probabilities. In the generalized form of softmax function $\boldsymbol{y} = \mathrm{softmax}(\frac{\boldsymbol{x}}{\tau})$, τ is the temperature parameter which could adjust the smoothness of the output. The larger the τ, the smoother the output; The smaller the τ, the steeper the output.

$$\xi_c = \hat{y}_c / \max_{c' \in \mathcal{Y}} \hat{y}_{c'} \tag{8}$$

If τ is sufficiently large, the small flipping probabilities problem would not occur even if we directly use the maximum in all prediction outputs. Moreover, because all information from the prediction outputs is preserved, the confidence conflict problem also does not arise. Meanwhile, since parameter τ could change the smoothness of the output, thus affecting the magnitude of the flipping probability, it can controls the size of candidate label set. As a result, the difficulty of the problem is altered. And a method which is able to explicitly adjust the hardness of the problem as with uniform assumptions is obtained.

4.2 Quantitative Results

We first evaluate the quantitative performances of CausalPLL in Table 1. It is worth noting that as CausalPLL is a generative approach that emphasize on the identifiability of causal representation, we do not expect CausalPLL to exhibit superior accuracy than methods that focus solely on classification because causal representations/relationships do not necessarily performs the best for classification.

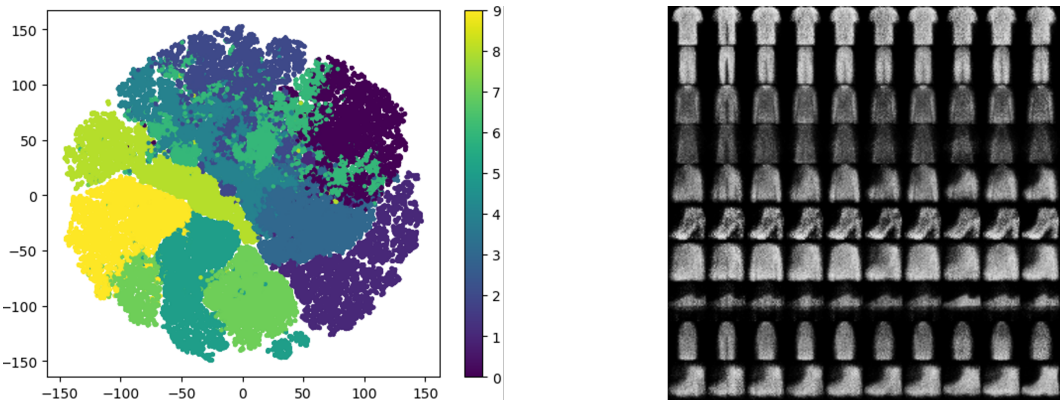

Figure 2: (Left) The t-SNE visualization of the latent space $\mathcal{Z}_2$ (i.e. Encoder-I). Each dot represents a sample, and different colors indicate different categories. (Right) Reconstructed images from Fashion-MNIST with different $\boldsymbol{z}_1$ and y. Images from the same row are generated from the same $\boldsymbol{z}_1$, but have labels which vary from 0 to C (from left to right). Images from the same column are generated from the same label y, but have different latent embeddings $\boldsymbol{z}_1$.

However, the performance of CausalPLL is still comparable to existing PLL baselines, and outperforms baselines when τ is high, i.e., CausalPLL performs better when the candidate label set is more difficult. Meanwhile, it is worth noting that CausalPLL gave high accuracy on Fashion-MNIST and KMNIST, but performed slightly worse on MNIST. This may be explained by the fact that MNIST is an easier task for classification-oriented baselines as the ambiguity in the candidate label sets is lower than other tasks, while CausalPLL may loses some accuracy due to ignoring spurious features.

4.3 Qualitative Results

4.3.1 The Effect of Latent Factors on Image Reconstruction

In this section, we visualize the reconstructed images from Fashion-MNIST with different $\boldsymbol{z}_1$ and y, i.e., the latent embedding given by encoder-II and the prediction of classifier. These images were generated by the decoder $p(\boldsymbol{x}|\boldsymbol{z}_1, y)$. We selected 10 images from different categories, inferenced the latent factors, and obtained 10 latent embedding $\boldsymbol{z}_1$. For each $\boldsymbol{z}_1$, we then fed them into the decoder with 10 different labels $y \in 1, 2, \cdots, C$. Finally, we got $10 \times 10 = 100$ images. In Figure 2, images in the same row have the same $\boldsymbol{z}_1$ and those in the same column share the same y.

From Figure 2, we can see that the second column (from the left to right, the same below) which has the label "trousers" all appear to have a slit in the middle and look likes real trousers. Moreover, in the fifth and seventh row of the last column, where there should have been coats and shirts appears what looked like boots. Similarly, this situation also appeared in sneakers and sandals. Beyond these, images on the diagonal are usually clearer than the other images in the same role, because these images have been given the correct labels. These results demonstrate that $\boldsymbol{z}_1$ and y actually play causal factor roles. Both of them have an important impact on the generation of images. By adjusting the value of $\boldsymbol{z}_1$ and y, we can achieve different reconstruction effects.

4.3.2 Visualization of Latent Space

In this section, we visualize the latent space of $\boldsymbol{z}_2$ (i.e. Encoder-I). We calculated the posterior $q(\boldsymbol{z}_{2,n}|\boldsymbol{x}_n) \sim \mathcal{N}(\mu_n, \sigma_n^2)$ for every instance in Fashion-MNIST dataset, and carried out t-SNE visualization on their mean vectors ($\{\boldsymbol{\mu}_n|1 \leq n \leq N\}$). From Figure 2 we can see that the points representing ankle boots, bags, sneakers and sandals (at the bottom left of the figure) formed clusters that have clear boundaries. But the points corresponding to t-shirts, pullovers, dresses and coats showed significant aliasing at the boundaries but still are separable.

The above phenomena, to some extent, confirmed that CAUSALPLL could capture the causal factors of instances rather than simply classifying the samples. T-shirts, pullovers, dresses and coats are all tops and shares similar features. As a consequence, the model tend to cluster them together and this

aliasing reflects the natural structure of the data. Meanwhile, ankle boots, bags, sneakers and sandals are all categories with distinct characteristics and are very different from each other. Therefore, between clusters of these classes, there are clear divides.

5 Conclusion

In this paper, we proposed CAUSALPLL, a VAE-based framework that learns semantically meaningful causal representations from partially labeled data, and investigated the causal factors in the context of instance-dependent PLL tasks. This model not only achieved comparable classification performance with state-of-the-art PLL baselines, but also learned causal factors that can significantly affect the reconstructed images. Meanwhile, there are also some limitations in our works and many future directions worth investigating. CAUSALPLL relies heavily on high-quality pre-train models and hyperparameter tuning, which implicitly increases the difficulty of its use. Quantitative evaluating the disentanglement of content and style in the latent space is also worth exploring.

References

[1] Kartik Ahuja, Divyat Mahajan, Yixin Wang, and Yoshua Bengio. Interventional causal representation learning. In *International Conference on Machine Learning*, pages 372–407. PMLR, 2023.

[2] Johann Brehmer, Pim De Haan, Phillip Lippe, and Taco S. Cohen. Weakly supervised causal representation learning. *Advances in Neural Information Processing Systems*, 35:38319–38331, 2022.

[3] Ching-Hui Chen, Vishal M Patel, and Rama Chellappa. Learning from ambiguously labeled face images. *IEEE transactions on pattern analysis and machine intelligence*, 40(7):1653–1667, 2017.

[4] Timothee Cour, Ben Sapp, and Ben Taskar. Learning from partial labels. *The Journal of Machine Learning Research*, 12:1501–1536, 2011.

[5] Timothee Cour, Benjamin Sapp, Chris Jordan, and Ben Taskar. Learning from ambiguously labeled images. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 919–926. IEEE, 2009.

[6] Ilyes Khemakhem, Diederik Kingma, Ricardo Monti, and Aapo Hyvarinen. Variational autoencoders and nonlinear ica: A unifying framework. In *International Conference on Artificial Intelligence and Statistics*, pages 2207–2217. PMLR, 2020.

[7] Bohdan Kivva, Goutham Rajendran, Pradeep Ravikumar, and Bryon Aragam. Identifiability of deep generative models without auxiliary information. *Advances in Neural Information Processing Systems*, 35:15687–15701, 2022.

[8] Phillip Lippe, Sara Magliacane, Sindy Löwe, Yuki M Asano, Taco Cohen, and Efstratios Gavves. Intervention design for causal representation learning. In *UAI 2022 Workshop on Causal Representation Learning*, 2022.

[9] 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 *international conference on machine learning*, pages 4114–4124. PMLR, 2019.

[10] Jie Luo and Francesco Orabona. Learning from candidate labeling sets. *Advances in neural information processing systems*, 23, 2010.

[11] Jiaqi Lv, Miao Xu, Lei Feng, Gang Niu, Xin Geng, and Masashi Sugiyama. Progressive identification of true labels for partial-label learning. In *International Conference on Machine Learning*, pages 6500–6510. PMLR, 2020.

[12] Michael Puthawala, Konik Kothari, Matti Lassas, Ivan Dokmanić, and Maarten De Hoop. Globally injective relu networks. *The Journal of Machine Learning Research*, 23(1):4544–4598, 2022.

[13] Tobias Scheffer, Christian Decomain, and Stefan Wrobel. Mining the web with active hidden markov models. In *Proceedings 2001 IEEE International Conference on Data Mining*, pages 645–646. IEEE, 2001.

[14] Bernhard Schölkopf, Francesco Locatello, Stefan Bauer, Nan Rosemary Ke, Nal Kalchbrenner, Anirudh Goyal, and Yoshua Bengio. Toward causal representation learning. *Proceedings of the IEEE*, 109(5):612–634, 2021.

[15] Wei Tang, Weijia Zhang, and Min-Ling Zhang. Multi-instance partial-label learning: Towards exploiting dual inexact supervision. *arXiv preprint arXiv:2212.08997*, 2022.

[16] Wei Tang, Weijia Zhang, and Min-Ling Zhang. Disambiguated attention embedding for multi-instance partial-label learning. *arXiv preprint arXiv:2305.16912*, 2023.

[17] Jakub Tomczak and Max Welling. VAE with a VampPrior. In *International Conference on Artificial Intelligence and Statistics*, pages 1214–1223. PMLR, 2018.

[18] Dong-Dong Wu, Deng-Bao Wang, and Min-Ling Zhang. Revisiting consistency regularization for deep partial label learning. In *International Conference on Machine Learning*, pages 24212–24225. PMLR, 2022.

[19] Ning Xu, Congyu Qiao, Xin Geng, and Min-Ling Zhang. Instance-dependent partial label learning. *Advances in Neural Information Processing Systems*, 34:27119–27130, 2021.

[20] Yu Yao, Tongliang Liu, Mingming Gong, Bo Han, Gang Niu, and Kun Zhang. Instance-dependent label-noise learning under a structural causal model. *Advances in Neural Information Processing Systems*, 34:4409–4420, 2021.

[21] Zinan Zeng, Shijie Xiao, Kui Jia, Tsung-Han Chan, Shenghua Gao, Dong Xu, and Yi Ma. Learning by associating ambiguously labeled images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 708–715, 2013.

[22] Min-Ling Zhang, Fei Yu, and Cai-Zhi Tang. Disambiguation-free partial label learning. *IEEE Transactions on Knowledge and Data Engineering*, 29(10):2155–2167, 2017.

[23] Min-Ling Zhang, Bin-Bin Zhou, and Xu-Ying Liu. Partial label learning via feature-aware disambiguation. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1335–1344, 2016.

[24] Weijia Zhang, Xuanhui Zhang, and Min-Ling Zhang. Multi-instance causal representation learning for instance label prediction and out-of-distribution generalization. *Advances in Neural Information Processing Systems*, 35:34940–34953, 2022.

6 Supplementary Materials

6.1 The Derivation of the ELBO

The generation and inference process of CAUSALPLL can be written as follows:

$$p(\boldsymbol{x}, y, \boldsymbol{z}_1, \boldsymbol{z}_2, \boldsymbol{s}) = p(\boldsymbol{s}|\boldsymbol{x}, y) \cdot p(\boldsymbol{x}|\boldsymbol{z}_1, y) \cdot p(\boldsymbol{z}_1|\boldsymbol{z}_2) \cdot p(\boldsymbol{z}_2) \cdot p(y)$$

$$q(\boldsymbol{z}_2, \boldsymbol{z}_1, y|\boldsymbol{x}, \boldsymbol{s}) = q(\boldsymbol{z}_1|\boldsymbol{z}_2, y, \boldsymbol{x}, \boldsymbol{s}) \cdot q(y|\boldsymbol{z}_2, \boldsymbol{x}, \boldsymbol{s}) \cdot q(\boldsymbol{z}_2|\boldsymbol{x}, \boldsymbol{s})$$

According to the conditional independence relationship implied in the causal factorization, we have: $q(\boldsymbol{z}_1|\boldsymbol{z}_2, y, \boldsymbol{x}, \boldsymbol{s}) = q(\boldsymbol{z}_1|\boldsymbol{z}_2, y),\ q(y|\boldsymbol{z}_2, \boldsymbol{x}, \boldsymbol{s}) = q(y|\boldsymbol{z}_2, \boldsymbol{s}),\ q(\boldsymbol{z}_2|\boldsymbol{x}, \boldsymbol{s}) = q(\boldsymbol{z}_2|\boldsymbol{x})$.

Then we can simplify the variational posterior as:

$$q(\boldsymbol{z}_2, \boldsymbol{z}_1, y|\boldsymbol{x}, \boldsymbol{s}) = q(\boldsymbol{z}_1|\boldsymbol{z}_2, y) \cdot q(y|\boldsymbol{z}_2, \boldsymbol{s}) \cdot q(\boldsymbol{z}_2|\boldsymbol{x})$$

Therefore, we have:

$$\begin{aligned}
&\ln \frac{p(\boldsymbol{x}, y, \boldsymbol{z}_1, \boldsymbol{z}_2, \boldsymbol{s})}{q(\boldsymbol{z}_2, \boldsymbol{z}_1, y \mid \boldsymbol{x}, \boldsymbol{s})} \\
=&\ln \frac{p(\boldsymbol{s}|\boldsymbol{x}, y) \cdot p(\boldsymbol{x}|\boldsymbol{z}_1, y) \cdot p(\boldsymbol{z}_1|\boldsymbol{z}_2) \cdot p(\boldsymbol{z}_2) \cdot p(y)}{q(\boldsymbol{z}_1|\boldsymbol{z}_2, y) \cdot q(y|\boldsymbol{z}_2, \boldsymbol{s}) \cdot q(\boldsymbol{z}_2|\boldsymbol{x})} \\
=&\ln p(\boldsymbol{x}|\boldsymbol{z}_1, y) + \ln p(\boldsymbol{s}|\boldsymbol{x}, y) - \ln \frac{q(\boldsymbol{z}_1|\boldsymbol{z}_2, y)}{p(\boldsymbol{z}_1|\boldsymbol{z}_2)} - \ln \frac{q(y|\boldsymbol{z}_2, \boldsymbol{s})}{p(y)} - \ln \frac{q(\boldsymbol{z}_2|\boldsymbol{x})}{p(\boldsymbol{z}_2)}
\end{aligned}$$

The evidence lower bound can be written as:

$$
\begin{aligned}
b_{\mathrm{ELBO}} = & \int_{\boldsymbol{z}_1, y} q(\boldsymbol{z}_1|\boldsymbol{z}_2, y) \cdot q(y|\boldsymbol{z}_2) \cdot \ln p(\boldsymbol{x}|\boldsymbol{z}_1, y) \ d\boldsymbol{z}_1 dy \\
& + \int_{y} q(y|\boldsymbol{z}_2) \cdot \ln p(\boldsymbol{s}|\boldsymbol{x}, y) \ dy \\
& - \int_{\boldsymbol{z}_2, \boldsymbol{z}_1, y} q(\boldsymbol{z}_1|\boldsymbol{z}_2, y) \cdot q(y|\boldsymbol{z}_2, \boldsymbol{s}) \cdot q(\boldsymbol{z}_2|\boldsymbol{x}) \cdot \ln \frac{q(\boldsymbol{z}_1|\boldsymbol{z}_2, y)}{p(\boldsymbol{z}_1|\boldsymbol{z}_2)} \ d\boldsymbol{z}_2 d\boldsymbol{z}_1 dy \\
& - \int_{\boldsymbol{z}_2, y} q(y|\boldsymbol{z}_2) \cdot q(\boldsymbol{z}_2|\boldsymbol{x}) \cdot \ln \frac{q(y|\boldsymbol{z}_2)}{p(y)} \ d\boldsymbol{z}_2 dy \\
& - \int_{\boldsymbol{z}_2} q(\boldsymbol{z}_2|\boldsymbol{x}) \cdot \ln \frac{q(\boldsymbol{z}_2|\boldsymbol{x})}{p(\boldsymbol{z}_2)} \ d\boldsymbol{z}_2
\end{aligned}
$$

$$
\begin{aligned}
& \int_{\boldsymbol{z}_1, y} q(\boldsymbol{z}_1|\boldsymbol{z}_2, y) \cdot q(y|\boldsymbol{z}_2) \cdot \ln p(\boldsymbol{x}|\boldsymbol{z}_1, y) \ d\boldsymbol{z}_1 dy = \mathbb{E}_{\boldsymbol{z}_1, y \sim q(\boldsymbol{z}_1, y|\boldsymbol{z}_2)}[\ln p(\boldsymbol{x}|\boldsymbol{z}_1, y)] \\
& \int_{y} q(y|\boldsymbol{z}_2) \cdot \ln p(\boldsymbol{s}|\boldsymbol{x}, y) \ dy = \mathbb{E}_{y \sim q(y|\boldsymbol{z}_2)}[p(\boldsymbol{s}|\boldsymbol{x}, y)] \\
& \int_{\boldsymbol{z}_2, \boldsymbol{z}_1, y} q(\boldsymbol{z}_1|\boldsymbol{z}_2, y) \cdot q(y|\boldsymbol{z}_2, \boldsymbol{s}) \cdot q(\boldsymbol{z}_2|\boldsymbol{x}) \cdot \ln \frac{q(\boldsymbol{z}_1|\boldsymbol{z}_2, y)}{p(\boldsymbol{z}_1|\boldsymbol{z}_2)} \ d\boldsymbol{z}_2 d\boldsymbol{z}_1 dy \\
& = \mathbb{E}_{\boldsymbol{z}_2, y \sim q(\boldsymbol{z}_2, y|\boldsymbol{x}, \boldsymbol{s})} \Big[\mathrm{KL}\Big(q(\boldsymbol{z}_1|\boldsymbol{z}_2, y) || p(\boldsymbol{z}_1|\boldsymbol{z}_2) \Big) \Big] \\
& \int_{\boldsymbol{z}_2, y} q(y|\boldsymbol{z}_2) \cdot q(\boldsymbol{z}_2|\boldsymbol{x}) \cdot \ln \frac{q(y|\boldsymbol{z}_2)}{p(y)} \ d\boldsymbol{z}_2 dy = \mathbb{E}_{y \sim q(y|\boldsymbol{z}_2)}[\mathrm{KL}(q(y|\boldsymbol{z}_2)||p(y))] \\
& \int_{\boldsymbol{z}_2} q(\boldsymbol{z}_2|\boldsymbol{x}) \cdot \ln \frac{q(\boldsymbol{z}_2|\boldsymbol{x})}{p(\boldsymbol{z}_2)} \ d\boldsymbol{z}_2 = \mathrm{KL}(q(\boldsymbol{z}_2|\boldsymbol{x})||p(\boldsymbol{z}_2))
\end{aligned}
$$

Therefore:

$$
\begin{aligned}
\mathcal{L} = & \ \mathrm{BCE}(\boldsymbol{x}, \hat{\boldsymbol{x}}) + \mathrm{BCE}(\boldsymbol{s}, \hat{\boldsymbol{s}}) + \mathrm{KL}\Big(q(\boldsymbol{z}_1|\boldsymbol{z}_2, y) \Big|\Big| \mathcal{N}(\boldsymbol{0}, \boldsymbol{E}) \Big) \\
& + L(\hat{y}, \boldsymbol{s}) + \Psi(\hat{y}, \boldsymbol{s}) + \mathrm{KL}\Big(q(\boldsymbol{z}_2|\boldsymbol{x}) \Big|\Big| p(\boldsymbol{z}_2) \Big) + \Omega\Big(p(\boldsymbol{z}_2) \Big)
\end{aligned}
$$

6.2 Consistency Regularization loss

Manifold consistency regularization, which assumes that the manifold structure in the feature space should also be preserved in the label space, has been shown very effective in traditional PLL tasks. Inspires by this, [18] revisited the utilization of consistency regularization to guide the design of deep PLL method, which is proven to be effective. However, original consistency regularization loss relies on semantic-preserving data augmentation, which is operated on the original sample space. Meanwhile, semantic-preserving data augmentation is generally very limited in number, and varies across different domains (CV, NLP, etc.). And this limits its large-scale application. But if the perturbation could be applied to the data on high-level feature space, limitations mentioned above would not exist any longer. And this aligns precisely with the characteristics of Variational Autoencoders (VAE).

$$
L(\boldsymbol{h}(\boldsymbol{z}_2), \boldsymbol{s}) + \lambda \cdot \Psi(\boldsymbol{h}(\boldsymbol{z}_2), \boldsymbol{s})
$$

The consistency regularization loss in our work consists of two parts. The first part, supervised loss $L(\boldsymbol{h}(\boldsymbol{z}_2), \boldsymbol{s})$, is the same as in [18]. The main idea of this term can be summarized as "only the negative samples counts". Because the candidate label might be a false positive label, which will introduce noise into the training process. But the negative label must be accurate.

$$
L(\boldsymbol{h}(\boldsymbol{z}_2), \boldsymbol{s}) = \sum_{c=1}^{C} (1 - s_c) \cdot \ln(1 - h_c(\boldsymbol{z}_2))
$$

The second part is the Consistency Regularization term. In original CR loss, a latent embedding has different "incarnations" which are obtained from different data augmentation approaches. And the model would align the output distribution of these different augmentations of each instance to their conformal label distribution $\boldsymbol{p}$. The reason for this is that the network's output is expected to be invariant to small changes applied to the feature space.

$$\Psi(\boldsymbol{h}(\boldsymbol{z}_2), \boldsymbol{s}) = \sum_{\tilde{\boldsymbol{z}}_2 \in \mathcal{R}_{\boldsymbol{z}_2}} \mathrm{KL}(\boldsymbol{p} || h(\tilde{\boldsymbol{z}}_2))$$

However, in our work, we adopted a novel approach which does not rely on semantic-preserving data augmentations. We know that, latent code $\boldsymbol{z}_2$ is generated using reparameterization tricks.

$$\boldsymbol{z}_2 = \boldsymbol{\mu}_2 + \boldsymbol{\sigma}_2 \odot \boldsymbol{\varepsilon}, \quad \boldsymbol{\varepsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{E})$$

In our framework, we do not use data augmentations, but directly apply perturbation to the data on latent space. Because epsilon was randomly sampled from the standard normal distribution, in this way, we can get any number of "incarnations" of a latent code. Let $\mathcal{R} = \{\tilde{\boldsymbol{z}}_2^{[r]} \mid 1 \leq r \leq R\}$.

$$\tilde{\boldsymbol{z}}_2^{[r]} = \boldsymbol{\mu}_2 + \boldsymbol{\sigma}_2 \odot \tilde{\boldsymbol{\varepsilon}}^{[r]}, \quad \tilde{\boldsymbol{\varepsilon}}^{[r]} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{E})$$

And each element of $\boldsymbol{p}^*$ on the candidate labels can be easily calculated as:

$$p_c^* = \frac{\sqrt[|\mathcal{R}_{\boldsymbol{z}}|]{\prod_{\tilde{\boldsymbol{z}} \in \mathcal{R}_{\boldsymbol{z}}} h_c(\tilde{\boldsymbol{z}})}}{\sum_{c'=1}^{C} \sqrt[|\mathcal{R}_{\boldsymbol{z}}|]{\prod_{\tilde{\boldsymbol{z}} \in \mathcal{R}_{\boldsymbol{z}}} h_{c'}(\tilde{\boldsymbol{z}})}}$$

6.3 Variance Regularization

$$\Omega_{1,2} = \sum_{i=1}^{M} \sum_{j=1}^{i} \left\| \frac{(\boldsymbol{\Sigma}_{u_1})_{ii}}{(\boldsymbol{\Sigma}_{u_2})_{ii}} - \frac{(\boldsymbol{\Sigma}_{u_1})_{jj}}{(\boldsymbol{\Sigma}_{u_2})_{jj}} \right\|_2^2$$

$$\Omega_{\mathrm{all}} = \sum_{i=1}^{K} \sum_{j=1}^{i} \Omega_{i,j}$$

In section 3.1, we briefly introduced the implementation of variance regularization. However, if we calculate Ω_{all} directly (i.e. sum $\Omega_{i,j}$ pairwisely over K and M), we have to calculate $C_K^2 \cdot C_M^2$ terms, which is very expensive, or almost computational-prohibited. One possible solution is to randomly select a fixed number (in our settings, 10000) of $\Omega_{i,j}$ at a time from C_K^2 items and optimize them. Practice has proves that this solution is effective and reduces the computation cost of the model.

6.4 The Method of Generating Instance-Dependent PLL Data from Fully-Supervised Datasets

In previous partial label learning researches, it is a popular way to convert the existing fully-supervised dataset into the partial label dataset. For instance-dependent PLL, a common method comes from [19]. In this approach, these datasets are manually corrupted into partially labeled versions by using a flipping probability $\boldsymbol{\xi} \in [0, 1]^C$. To synthesize the flipping probability, a clean neural network is trained with the original clean labels and gives the confidence prediction $\hat{\boldsymbol{y}}$. The flipping probability to ground-truth label c^* is necessarily 1. For other labels, flipping probability can be obtained by dividing their own predicted outputs by the largest output in the candidate label set respectively, i. e., $\xi_c = \hat{y}_c / \max_{c' \neq c^*} \hat{y}_{c'}$. Then, the candidate labels can be randomly sampled from the Bernoulli distribution with ξ as the parameter. The authors may have taken this approach because sometimes the output of the neural network may be concentrated on a particular label (for example, $(1e-6, 1e-6, 1.0)$). At this time, if the maximum value is directly selected in all prediction outputs, the flipping probability may be too small, so that the partial label data cannot be effectively obtained. Therefore, the largest output in the candidate label set is adopted as denominator, which could alleviate this issue.

However, this may lead to some problems. If the outputs are about the same size except for the ground-truth label, doing so results in a flipping probability very close to 1 for all label. For

example, for a instance whose ground-truth label $\boldsymbol{y}$ is $(0,0,0,0,1)^T$, if we have a very confident prediction $\hat{\boldsymbol{y}} = (.01,.01,.01,.01,.96)^T$, the corresponding $\boldsymbol{s}$ would be $(1,1,1,1,1)^T$, which is very unconfident. And the contradiction appeared. Since neural networks have a tendency to be overconfident, this is very likely to happen. Moreover, in the traditional uniform partial label generation paradigm, there is a parameter q that explicitly controls the amount of candidate labels, which represents the difficulty of the PLL problem, to some extent. But in IDPLL, lack of such a method can adjust the difficulty of the task. To solve these problems, we proposed a new approach. We know that in softmax function, there can be a temperature parameter τ.

$$\boldsymbol{y} = \mathrm{softmax}(\frac{\boldsymbol{x}}{\tau})$$

This parameter could adjust the smoothness of the output. The larger the τ, the smoother the output; The smaller the τ, the steeper the output.

$$\xi_c = \hat{y}_c / \max_{c' \in \mathcal{Y}} \hat{y}_{c'}$$

If we make τ sufficiently large, even if we directly use the maximum in **all prediction outputs**, we would not have the problem that the flipping probability is too small. Moreover, because all information from the prediction outputs is preserved, the confidence conflict problem also does not arise. Meanwhile, since parameter τ could change the smoothness of the output, thus affecting the magnitude of the flipping probability, it can controls the size of candidate label set. As a result, the difficulty of the problem is altered. And a method which is able to explicitly adjust the hardness of the problem as with uniform assumptions is obtained.