CGU-BAYES: Causal Graph Uncertainty-Guided Bayesian Inference for Domain Generalization

Supplementary Material

A. Detailed Derivations

A.1. Derivations for Eq. (1)

To perform Bayesian inference we start from the target predictive distribution $p(y \mid \boldsymbol{x}^t, \mathcal{D})$ and introduce an integral over causal graphs $\mathcal{G}$. This yields two factors in the derivation: $p(y \mid \boldsymbol{x}^t, \mathcal{G}, \mathcal{D})$ and $p(\mathcal{G} \mid \boldsymbol{x}^t, \mathcal{D})$. Under i.i.d. assumptions, the first term reduces to $p(y \mid \boldsymbol{x}^t, \mathcal{G})$. The main challenge is therefore to approximate $p(\mathcal{G} \mid \boldsymbol{x}^t, \mathcal{D})$.

In practice constructing $p(\mathcal{G} \mid \mathbf{x}^t, \mathcal{D})$ is generally infeasible. The target-domain conditional requires information that is typically unavailable at test time, for example the values of latent variables U and the label Y associated with the input $\mathbf{x}^t$. Estimating a full posterior for every test input would also be computationally prohibitive. For these reasons we instead sample candidate causal graphs $\mathcal{G}$ from the dataset-level posterior $p(\mathcal{G} \mid \mathcal{D})$ and use those samples to approximate the original predictive distribution. The detailed derivation and the assumptions used to justify this approximation are given in Eq. (9) below.

$$
\begin{aligned}
p(y|\boldsymbol{x}^t, \mathcal{D}) &= \int_{\mathcal{G}} p(y|\boldsymbol{x}^t, \mathcal{G}, \mathcal{D}) p(\mathcal{G}|\boldsymbol{x}^t, \mathcal{D}) \, \mathrm{d}\mathcal{G} \\
&\quad \text{We assume } y \perp\!\!\!\perp \mathcal{D} | \boldsymbol{x}^t, \mathcal{G} \\
&= \int_{\mathcal{G}} p(y|\boldsymbol{x}^t, \mathcal{G}) p(\mathcal{G}|\boldsymbol{x}^t, \mathcal{D}) \, \mathrm{d}\mathcal{G} \\
&\quad \text{Bayes theorem} \\
&= \int_{\mathcal{G}} p(y|\boldsymbol{x}^t, \mathcal{G}) \frac{p(\boldsymbol{x}^t|\mathcal{G}, \mathcal{D}) p(\mathcal{G}|\mathcal{D})}{p(\boldsymbol{x}^t|\mathcal{D})} \, \mathrm{d}\mathcal{G} \\
&\quad \text{We assume } \boldsymbol{x}^t \perp\!\!\!\perp \mathcal{D} | \mathcal{G} \\
&= \int_{\mathcal{G}} p(y|\boldsymbol{x}^t, \mathcal{G}) \frac{p(\boldsymbol{x}^t|\mathcal{G}) p(\mathcal{G}|\mathcal{D})}{p(\boldsymbol{x}^t|\mathcal{D})} \, \mathrm{d}\mathcal{G} \\
&= \frac{1}{p(\boldsymbol{x}^t|\mathcal{D})} \int_{\mathcal{G}} p(y|\boldsymbol{x}^t, \mathcal{G}) p(\boldsymbol{x}^t|\mathcal{G}) p(\mathcal{G}|\mathcal{D}) \, \mathrm{d}\mathcal{G} \\
&\propto \int_{\mathcal{G}} p(y|\boldsymbol{x}^t, \mathcal{G}) p(\boldsymbol{x}^t|\mathcal{G}) p(\mathcal{G}|\mathcal{D}) \, \mathrm{d}\mathcal{G} \\
&= \mathbb{E}_{\mathcal{G} \sim p(\mathcal{G}|\mathcal{D})} \Big[p(y|\boldsymbol{x}^t, \mathcal{G}) p(\boldsymbol{x}^t|\mathcal{G}) \Big]
\end{aligned}
\tag{9}
$$

A.2. Derivation for Eq. (2)

$$
\begin{aligned}
&p(y|\boldsymbol{x}^t, \mathcal{G}) \\
&\quad \text{introduce other variables in SCM, } (U, \boldsymbol{Z}). \\
&= \int_{\boldsymbol{z}} \sum_{u} p(y|\boldsymbol{x}^t, \boldsymbol{z}, u, \mathcal{G}) p(\boldsymbol{z}, u|\boldsymbol{x}^t, \mathcal{G}) \, \mathrm{d}\boldsymbol{z} \\
&= \int_{\boldsymbol{z}} \sum_{u} p(y|\boldsymbol{x}^t, \boldsymbol{z}_o^{\mathcal{G}}, \boldsymbol{z}_{cmb}^{\mathcal{G}}, u, \mathcal{G}) p(\boldsymbol{z}, u|\boldsymbol{x}^t, \mathcal{G}) \, \mathrm{d}\boldsymbol{z} \\
&\quad (Y \perp\!\!\!\perp \boldsymbol{Z}_o^{\mathcal{G}}, U | \boldsymbol{Z}_{cmb}^{\mathcal{G}})_{\mathcal{G}} \\
&= \int_{\boldsymbol{z}} \sum_{u} p(y|\boldsymbol{z}_{cmb}^{\mathcal{G}}) p(\boldsymbol{z}_{cmb}^{\mathcal{G}}, \boldsymbol{z}_o^{\mathcal{G}}, u|\boldsymbol{x}^t, \mathcal{G}) \, \mathrm{d}\boldsymbol{z} \\
&= \int_{\boldsymbol{z}_{cmb}^{\mathcal{G}}} p(y|\boldsymbol{z}_{cmb}^{\mathcal{G}}) \int_{\boldsymbol{z}_o^{\mathcal{G}}} \sum_{u} p(\boldsymbol{z}_{cmb}^{\mathcal{G}}, \boldsymbol{z}_o^{\mathcal{G}}, u|\boldsymbol{x}^t, \mathcal{G}) \, \mathrm{d}\boldsymbol{z}_o^{\mathcal{G}} \, \mathrm{d}\boldsymbol{z}_{cmb}^{\mathcal{G}} \\
&= \int_{\boldsymbol{z}_{cmb}^{\mathcal{G}}} p(y|\boldsymbol{z}_{cmb}^{\mathcal{G}}) p(\boldsymbol{z}_{cmb}^{\mathcal{G}}|\boldsymbol{x}^t, \mathcal{G}) \, \mathrm{d}\boldsymbol{z}_{cmb}^{\mathcal{G}} \\
&\quad \text{Maximum-a-posterior estimation} \\
&\approx p\big(y|(\boldsymbol{z}_{cmb}^{\mathcal{G}})^*\big), (\boldsymbol{z}_{cmb}^{\mathcal{G}})^* = \arg\max_{\boldsymbol{z}_{cmb}^{\mathcal{G}}} p(\boldsymbol{z}_{cmb}^{\mathcal{G}}|\boldsymbol{x}^t, \mathcal{G})
\end{aligned}
\tag{10}
$$

A.3. Derivation of Eq. (5)

According to the entropy-based uncertainty quantification approach in Osband et al. [47], epistemic uncertainty $\mathcal{U}_e(\boldsymbol{x}^t|\mathcal{G})$ is intrinsically the mutual information between Y and $\boldsymbol{Z}_{cmb}$ given input $\boldsymbol{X}$, i.e., $\mathcal{I}\left[y, \boldsymbol{z}_{cmb}^{\mathcal{G}} | \boldsymbol{x}^t, \mathcal{G}\right]$.

$$
\begin{aligned}
&\underbrace{\mathcal{I}\left[y, \boldsymbol{z}_{cmb}^{\mathcal{G}} | \boldsymbol{x}^t, \mathcal{G}\right]}_{\text{Epistemic Uncertainty } \mathcal{U}_e(\boldsymbol{x}^t|\mathcal{G})} \\
&= \underbrace{\mathcal{H}\left[p(y|\boldsymbol{x}^t, \mathcal{G})\right]}_{\text{Total Uncertainty } \mathcal{U}_t(\boldsymbol{x}^t|\mathcal{G})} - \underbrace{\mathbb{E}_{p(\boldsymbol{z}_{cmb}^{\mathcal{G}}|\boldsymbol{x}^t, \mathcal{G})}\left[\mathcal{H}[p(y|\boldsymbol{z}_{cmb}^{\mathcal{G}})]\right]}_{\text{Aleatoric Uncertainty } \mathcal{U}_a(\boldsymbol{x}^t|\mathcal{G})}
\end{aligned}
\tag{11}
$$

We first compute the total uncertainty term $\mathcal{U}_t(\boldsymbol{x}^t|\mathcal{G})$, as shown in Eq. (12)

$$
\begin{aligned}
&\mathcal{U}_t(\boldsymbol{x}^t|\mathcal{G}) \\
&= \mathcal{H}\left[p(y|\boldsymbol{x}^t, \mathcal{G})\right] \\
&= \mathcal{H}\left[\int_{\boldsymbol{z}_{cmb}^{\mathcal{G}}} p(y|\boldsymbol{z}_{cmb}^{\mathcal{G}}) p(\boldsymbol{z}_{cmb}^{\mathcal{G}}|\boldsymbol{x}^t, \mathcal{G}) \, \mathrm{d}\boldsymbol{z}_{cmb}^{\mathcal{G}}\right] \\
&= \mathcal{H}\left[\mathbb{E}_{p(\boldsymbol{z}_{cmb}^{\mathcal{G}}|\boldsymbol{x}^t, \mathcal{G})}[p(y|\boldsymbol{z}_{cmb}^{\mathcal{G}})]\right] \\
&= \mathcal{H}\left[\frac{1}{S}\sum_{s=1}^{S} p(y|\boldsymbol{z}_{cmb}^{s})\right], (\boldsymbol{z}_{cmb}^{\mathcal{G}})^s \sim p(\boldsymbol{z}_{cmb}^{\mathcal{G}}|\boldsymbol{x}^t, \mathcal{G})
\end{aligned}
\tag{12}
$$

Then we compute the aleatoric uncertainty $\mathcal{U}_a(\boldsymbol{x}^t|\mathcal{G})$, as outlined in Eq. (13)

$$\begin{aligned}&\mathcal{U}_a(\boldsymbol{x}^t|\mathcal{G})\\=&\mathbb{E}_{p(\boldsymbol{z}^{\mathcal{G}}_{cmb}|\boldsymbol{x}^t,\mathcal{G})}\left[\mathcal{H}[p(y|\boldsymbol{z}^{\mathcal{G}}_{cmb})]\right]\\=&\frac{1}{S}\sum_{s=1}^{S}\mathcal{H}[p\left(y|(\boldsymbol{z}^{\mathcal{G}}_{cmb})^s\right)],(\boldsymbol{z}^{\mathcal{G}}_{cmb})^s\sim p(\boldsymbol{z}^{\mathcal{G}}_{cmb}|\boldsymbol{x}^t,\mathcal{G})\end{aligned} \tag{13}$$

Substituting Eq. (12) and Eq. (13) into Eq. (5), we have

$$\begin{aligned}&\mathcal{U}_e(\boldsymbol{x}^t|\mathcal{G})\\=&\mathcal{U}_t(\boldsymbol{x}^t|\mathcal{G})-\mathcal{U}_a(\boldsymbol{x}^t|\mathcal{G})\\=&\mathcal{H}\left[\frac{1}{S}\sum_{s=1}^{S}p(y|\left(\boldsymbol{z}^{\mathcal{G}}_{cmb}\right)^s)\right]-\frac{1}{S}\sum_{s=1}^{S}\mathcal{H}[p(y|\left(\boldsymbol{z}^{\mathcal{G}}_{cmb}\right)^s)]\\&(\boldsymbol{z}^{\mathcal{G}}_{cmb})^s\sim p(\boldsymbol{z}^{\mathcal{G}}_{cmb}|\boldsymbol{x}^t,\mathcal{G})\end{aligned} \tag{14}$$

For regression task, we train a neural network that takes $\boldsymbol{Z}^{\mathcal{G}}_{cmb}$ and outputs the mean and variance of distribution $p\left(y|(\boldsymbol{z}^{\mathcal{G}}_{cmb})^s\right)$, denoted as μ_s and σ_s^2. We assume each dimension of y is independent, resulting in a diagonal covariance matrix. Therefore, we only demonstrate the uncertainty calculation for one-dimensional Gaussians to ensure alignment with the experiments in Sec. 5.2. The extension to a multi-variable Gaussian distribution is straightforward. Hence we have,

$$p\left(y|(\boldsymbol{z}^{\mathcal{G}}_{cmb})^s\right)\sim\mathcal{N}(\mu_s,\sigma_s^2).$$

The entropy of $p\left(y|(\boldsymbol{z}^{\mathcal{G}}_{cmb})^s\right)$ can be computed via

$$\mathcal{H}[p\left(y|(\boldsymbol{z}^{\mathcal{G}}_{cmb})^s\right)]=\frac{1}{2}\log\left(2\pi e\sigma_s^2\right). \tag{15}$$

To calculate the entropy of a mixture Gaussian model with S components where each component has equal weights $\frac{1}{S}$, i.e., $\mathcal{H}\left[\frac{1}{S}\sum_{s=1}^{S}p(y|\left(\boldsymbol{z}^{\mathcal{G}}_{cmb}\right)^s)\right]$, we employ a closed-form Gaussian approximation, which is often used in Deep Ensembles for uncertainty estimation. We first compute the mixture mean and variance as follow,

$$\begin{aligned}\mu_{mix}=&\frac{1}{S}\sum_{s=1}^{S}\mu_s,\\\sigma^2_{mix}=&\underbrace{\frac{1}{S}\sum_{s=1}^{S}\sigma_s^2}_{\text{Average Variance}}+\underbrace{\frac{1}{S}\sum_{s=1}^{S}(\mu_s-\mu_{mix})^2}_{\text{Variance of Means}}.\end{aligned} \tag{16}$$

Hence, we have,

$$\mathcal{H}\left[\frac{1}{S}\sum_{s=1}^{S}p(y|\left(\boldsymbol{z}^{\mathcal{G}}_{cmb}\right)^s)\right]\approx\frac{1}{2}\log(2\pi e\sigma^2_{mix}). \tag{17}$$

Substituting Eq. (15) and Eq. (17) into Eq. (14), we have

$$\begin{aligned}&\mathcal{U}_e(\boldsymbol{x}^t|\mathcal{D})\\=&\frac{1}{2}\log\left(2\pi e(\frac{1}{S}\sum_{s=1}^{S}\sigma_s^2+\frac{1}{S}\sum_{s=1}^{S}(\mu_s-\frac{1}{S}\sum_{s=1}^{S}\mu_s))\right)\\&-\frac{1}{2S}\sum_{s=1}^{S}\log(2\pi e\sigma_S^2)\end{aligned} \tag{18}$$

For classification task, we assume $p\left(y|(\boldsymbol{z}^{\mathcal{G}}_{cmb})^s\right)$ follows categorical distribution. Let C be the total number of classes. Let $p_{s,c}$ be the probability that the s^{th} sample assigns to class c:

$$p_{s,c}=P(y=c|(\boldsymbol{z}^{\mathcal{G}}_{cmb})^s)$$

The entropy is

$$\mathcal{H}[p\left(y|(\boldsymbol{z}^{\mathcal{G}}_{cmb})^s\right)]=-\sum_{c=1}^{C}p_{s,c}\log p_{s,c}.$$

The entropy of the mixture of models:

$$\begin{aligned}&\mathcal{H}\left[\frac{1}{S}\sum_{s=1}^{S}p\left(y|(\boldsymbol{z}^{\mathcal{G}}_{cmb})^S\right)\right]\\=&-\sum_{c=1}^{C}\left(\frac{1}{S}\sum_{s=1}^{S}p_{s,c}\right)\log\left(\frac{1}{S}\sum_{s=1}^{S}p_{s,c}\right).\end{aligned} \tag{19}$$

Hence we have,

$$\begin{aligned}&\mathcal{U}_e(\boldsymbol{x}^t|\mathcal{G})\\=&-\sum_{c=1}^{C}\left(\frac{1}{S}\sum_{s=1}^{S}p_{s,c}\right)\log\left(\frac{1}{S}\sum_{s=1}^{S}p_{s,c}\right)\\&+\frac{1}{S}\sum_{s=1}^{S}\left(-\sum_{c=1}^{C}p_{s,c}\log p_{s,c}\right).\end{aligned} \tag{20}$$

A.4. Derivation of Eq. (6)

$$\begin{aligned}&\underbrace{\mathcal{H}\left[p(y|\boldsymbol{x}^t,\mathcal{D})\right]}_{\text{Total Uncertainty }\mathcal{U}_t(\boldsymbol{x}^t|\mathcal{D})}\\=&\mathcal{H}\left[\int_{\mathcal{G}}p(y|\boldsymbol{x}^t,\mathcal{G},\mathcal{D})p(\mathcal{G}|\boldsymbol{x}^t,\mathcal{D})\,\mathrm{d}\mathcal{G}\right]\\=&\mathcal{H}\left[\frac{\mathbb{E}_{\mathcal{G}\sim p(\mathcal{G}|\mathcal{D})}\left[p(y|\boldsymbol{x}^t,\mathcal{G})p(\boldsymbol{x}^t|\mathcal{G})\right]}{\mathbb{E}_{\mathcal{G}\sim p(\mathcal{G}|\mathcal{D})}\left[p(\boldsymbol{x}^t|\mathcal{G})\right]}\right]\\\approx&\mathcal{H}\left[\sum_{l=1}^{L}p(y|\boldsymbol{x}^t,\mathcal{G}^l)\frac{p(\boldsymbol{x}^t|\mathcal{G}^l)}{\sum_{l^*=1}^{L}p(\boldsymbol{x}^t|\mathcal{G}^{l^*})}\right],\mathcal{G}^l\sim p(\mathcal{G}|\mathcal{D}).\\=&\mathcal{H}\left[\sum_{l=1}^{L}\omega^l p(y|\boldsymbol{x}^t,\mathcal{G}^l)\right],\omega^l=\frac{p(\boldsymbol{x}^t|\mathcal{G}^l)}{\sum_{l^*=1}^{L}p(\boldsymbol{x}^t|\mathcal{G}^{l^*})}.\end{aligned} \tag{21}$$

$$
\begin{aligned}
&\underbrace{\mathbb{E}_{p(\mathcal{G}|\boldsymbol{x}^t,\mathcal{D})}\left[\mathcal{H}[p(y|\boldsymbol{x}^t,\mathcal{G})]\right]}_{\text{Aleatoric Uncertainty } \mathcal{U}_a(\boldsymbol{x}^t|\mathcal{D})}\\
=&\mathbb{E}_{p(\mathcal{G}|\mathcal{D})}\left[\frac{p(\boldsymbol{x}^t|\mathcal{G})\mathcal{H}[p(y|\boldsymbol{x}^t,\mathcal{G})]}{\mathbb{E}_{\mathcal{G}\sim p(\mathcal{G}|\mathcal{D})}\left[p(\boldsymbol{x}^t|\mathcal{G})\right]}\right]\\
=&\sum_{l=1}^{L}\left[\frac{p(\boldsymbol{x}^t|\mathcal{G}^l)}{\sum_{l^*=1}^{L}p(\boldsymbol{x}^t|\mathcal{G}^{l^*})}\mathcal{H}[p(y|\boldsymbol{x}^t,\mathcal{G}^l)]\right],\mathcal{G}^l\sim p(\mathcal{G}|\mathcal{D})\\
\approx&\sum_{l=1}^{L}\left[\omega^l\mathcal{H}[p(y|\boldsymbol{x}^t,\mathcal{G}^l)]\right]
\end{aligned}
\tag{22}
$$

Substituting Eqs. (21) and (22) into Eq. (6), we have:

$$
\begin{aligned}
&\mathcal{U}_e(\boldsymbol{x}^t|\mathcal{D})\\
=&\mathcal{U}_t(\boldsymbol{x}^t|\mathcal{D})-\mathcal{U}_a(\boldsymbol{x}^t|\mathcal{D})\\
=&\mathcal{H}\left[\sum_{l=1}^{L}p(y|\boldsymbol{x}^t,\mathcal{G}^l)\frac{p(\boldsymbol{x}^t|\mathcal{G}^l)}{\sum_{l^*=1}^{L}p(\boldsymbol{x}^t|\mathcal{G}^{l^*})}\right]\\
&-\sum_{l=1}^{L}\left[\frac{p(\boldsymbol{x}^t|\mathcal{G}^l)}{\sum_{l^*=1}^{L}p(\boldsymbol{x}^t|\mathcal{G}^{l^*})}\mathcal{H}[p(y|\boldsymbol{x}^t,\mathcal{G}^l)]\right]\\
&\mathcal{G}^l\sim p(\mathcal{G}|\mathcal{D})
\end{aligned}
\tag{23}
$$

Similarly, for regression task, if we assume

$$
p(y|\boldsymbol{x}^t,\mathcal{G}^l)\sim\mathcal{N}(\mu_l,\sigma_l^2),
$$

then

$$
\mathcal{H}[p(y|\boldsymbol{x}^t,\mathcal{G})]\approx\frac{1}{2}\log(2\pi e\sigma_l^2),
$$

and the mixture mean and variances are

$$
\begin{aligned}
\mu_{mix}=&\sum_{l=1}^{L}\omega^l\mu_l,\\
\sigma_{mix}^2=&\sum_{l=1}^{L}\omega^l\sigma_l^2+\sum_{l=1}^{L}\omega^l(\mu_l-\mu_{mix})^2,
\end{aligned}
\tag{24}
$$

the entropy of the mixture Gaussian model is

$$
\mathcal{H}\left[\sum_{l=1}^{L}\omega^l p(y|\boldsymbol{x}^t,\mathcal{G}^l)\right]=\frac{1}{2}\log(2\pi e\sigma_{mix}^2).
$$

The uncertainty is:

$$
\begin{aligned}
&u_e(\boldsymbol{x}^t|\mathcal{D})\\
=&\frac{1}{2}\log(2\pi e\sigma_{mix}^2)-\frac{1}{2}\sum_{l=1}^{L}\omega^l\log(2\pi e\sigma_l^2)\\
=&\frac{1}{2}\log\left(2\pi e\big(\sum_{l=1}^{L}\omega^l\sigma_l^2+\sum_{l=1}^{L}\omega^l(\mu_l-\mu_{mix})^2\big)\right)\\
&-\frac{1}{2}\sum_{l=1}^{L}\omega^l\log(2\pi e\sigma_l^2)
\end{aligned}
\tag{25}
$$

For classification task, if we assume $p(y|\boldsymbol{x}^t,\mathcal{G}^l)$ follows categorical distribution. Let C be the total number of classes. Let $p_{l,c}$ be the probability that the l^{th} sample assigns to class c:

$$
p_{l,c}=p(Y=c|\boldsymbol{X}=\boldsymbol{x}^t,\mathcal{G}^l).
$$

The entropy is

$$
\mathcal{H}[p(y|\boldsymbol{x}^t,\mathcal{G}^l)]=-\sum_{c=1}^{C}p_{l,c}\log p_{l,c}.
$$

The entropy of the mixture of models:

$$
\begin{aligned}
&\mathcal{H}\left[\sum_{l=1}^{L}\omega^l p(y|\boldsymbol{x}^t,\mathcal{G}^l)\right]\\
=&-\sum_{c=1}^{C}\left(\sum_{l=1}^{L}\omega^l p_{l,c}\right)\log\left(\sum_{l=1}^{L}\omega^l p_{l,c}\right).
\end{aligned}
\tag{26}
$$

Then the uncertainty is:

$$
\begin{aligned}
&\mathcal{U}_e(\boldsymbol{x}^t|\mathcal{D})\\
=&-\sum_{c=1}^{C}\left(\sum_{l=1}^{L}\omega^l p_{l,c}\right)\log\left(\sum_{l=1}^{L}\omega^l p_{l,c}\right)\\
&+\sum_{l=1}^{L}\omega^l\left(\sum_{c=1}^{C}p_{l,c}\log p_{l,c}\right)
\end{aligned}
\tag{27}
$$

A.5. Derivation of Eq. (7)

We start with the joint distribution of observed variables $\boldsymbol{X},Y,U$, i.e., $p(\boldsymbol{X},Y,U)$.

$$
\begin{aligned}
&-\log p(\boldsymbol{x},y,u)\\
=&-\log p(\boldsymbol{x}|y,u)p(y,u)\\
&\ y,u \text{ are observed variables. } p(y,u) \text{ is known.}\\
=&-\log p(y,u)-\log p(\boldsymbol{x}|y,u)\\
=&-\log p(\boldsymbol{x}|y,u)+c\quad -\log p(y,u)\text{ is constant.}\\
=&-\log\int_{\boldsymbol{z}}p(\boldsymbol{x},\boldsymbol{z}|y,u)\,\mathrm{d}\boldsymbol{z}+c\\
=&-\log\int_{\boldsymbol{z}}p(\boldsymbol{x}|\boldsymbol{z},y,u)p(\boldsymbol{z}|y,u)\,\mathrm{d}\boldsymbol{z}+c\\
=&-\log\int_{\boldsymbol{z}}p(\boldsymbol{x}|\boldsymbol{z})p(\boldsymbol{z}|y,u)\,\mathrm{d}\boldsymbol{z}+c\quad \boldsymbol{X}\perp\!\!\!\perp Y,U|\boldsymbol{Z}\\
=&-\log\int_{\boldsymbol{z}}\frac{p(\boldsymbol{x}|\boldsymbol{z})p(\boldsymbol{z}|y,u)}{q(\boldsymbol{z}|\boldsymbol{x})}q(\boldsymbol{z}|\boldsymbol{x})\,\mathrm{d}\boldsymbol{z}+c\\
=&-\log\mathbb{E}_{q(\boldsymbol{z}|\boldsymbol{x})}\left[\frac{p(\boldsymbol{x}|\boldsymbol{z})p(\boldsymbol{z}|y,u)}{q(\boldsymbol{z}|\boldsymbol{x})}\right]+c\\
\leq&-\mathbb{E}_{q(\boldsymbol{z}|\boldsymbol{x})}\left[\log\frac{p(\boldsymbol{x}|\boldsymbol{z})p(\boldsymbol{z}|y,u)}{q(\boldsymbol{z}|\boldsymbol{x})}\right]+c\\
=&-\mathbb{E}_{q(\boldsymbol{z}|\boldsymbol{x})}\left[\log p(\boldsymbol{x}|\boldsymbol{z})+\log p(\boldsymbol{z}|y,u)-\log q(\boldsymbol{z}|\boldsymbol{x})\right]+c
\end{aligned}
\tag{28}
$$

According to Eq. (28), $-\mathbb{E}_{q(\boldsymbol{z}|\boldsymbol{x})}\big[\log p(\boldsymbol{x}|\boldsymbol{z}) + \log p(\boldsymbol{z}|y,u) - \log q(\boldsymbol{z}|\boldsymbol{x})\big] + c$ is upper bound of the negative log joint likelihood over observed variables $\boldsymbol{X}, Y, U$. We ignore the constant term in training since there is no parameter to optimize. Its expected value over data observations from the training distribution $p_{\mathcal{D}}$ is defined as ELBO loss $\mathcal{L}_{\text{ELBO}}$ in Eq. (7).

Training Details for Our iVAE: As described in Eq. (7), the training objective comprises an ELBO loss $\mathcal{L}_{\text{ELBO}}$ and a score matching loss $\mathcal{L}_{\text{SM}}$. The ELBO loss $\mathcal{L}_{\text{ELBO}}$ optimizes over encoder and decoder parameters $(\boldsymbol{\theta}, \boldsymbol{\psi})$, while the score matching loss $\mathcal{L}_{\text{SM}}$ minimizes over prior parameters (T, λ). The parameters $(\boldsymbol{T}, \boldsymbol{\lambda})$ are constants in $\mathcal{L}_{\text{ELBO}}$, and $\boldsymbol{\theta}, \boldsymbol{\psi}$ are constants in $\mathcal{L}_{\text{SM}}$. The score matching loss $\mathcal{L}_{\text{SM}}$ minimizes over prior parameters $(\boldsymbol{T}, \boldsymbol{\lambda})$. The parameters $\boldsymbol{T}, \boldsymbol{\lambda}$ are constants in $\mathcal{L}_{\text{ELBO}}$, and $\boldsymbol{\theta}, \boldsymbol{\psi}$ are constants in $\mathcal{L}_{\text{SM}}$.

$$\mathcal{L}_{\text{iVAE}}(\boldsymbol{\theta}, \boldsymbol{\psi}, \boldsymbol{T}, \boldsymbol{\lambda}) \coloneqq \mathcal{L}_{\text{ELBO}}(\boldsymbol{\theta}, \boldsymbol{\psi}, \hat{\boldsymbol{T}}, \hat{\boldsymbol{\lambda}}) + \mathcal{L}_{\text{SM}}(\hat{\boldsymbol{\theta}}, \hat{\boldsymbol{\psi}}, \boldsymbol{T}, \boldsymbol{\lambda})$$

B. Theoretical Analysis

B.1. General Form of SCM Assumptions

The general form of SCM we proposed in Figure 1 should satisfy the assumptions in **Assumption 1**.

Assumption 1. *(a) U is the root node and does not have direct links with Y or $\boldsymbol{X}$. (b) Z_i is generated by either Y or U for any i; (c) $\boldsymbol{Z}_p$, $\boldsymbol{Z}_c$, and $\boldsymbol{Z}_s$ collectively form the Causal Markov Blanket (CMB) set of target Y. The CMB set of Y does not contain U. Y does not have a direct link to $\boldsymbol{X}$. (d) $\boldsymbol{X}$ is the child of Z_i for any i. $\boldsymbol{X}$ is the leaf node in the graph. (e) The causal graph over $\{\boldsymbol{X}, Y, \boldsymbol{Z}, U\}$ is a DAG.*

B.2. Latent Variable Learning

We adopt the NF-iVAE framework [41] and tailor it to our purpose. NF-iVAE trains the iVAE using a prior distribution on $\boldsymbol{Z}$ that aligns with **Assumption 1(b)** and belongs to the general exponential family, i.e.,

$$p_{\boldsymbol{T},\boldsymbol{\lambda}}(\boldsymbol{Z}|Y,U) = \frac{\mathcal{Q}(\boldsymbol{Z})}{\mathcal{C}(Y,U)} \exp[\boldsymbol{T}(\boldsymbol{Z})^T \boldsymbol{\lambda}(Y,U)]$$

where $\mathcal{Q}$ is the base measure. $\mathcal{C}$ is the normalizing constant. $\boldsymbol{\lambda}$ is the arbitrary function. $\boldsymbol{T}$ is the sufficient statistics.[4]

From the joint distribution $p(\boldsymbol{x}, y, u) \propto \int_{\boldsymbol{z}} p(\boldsymbol{x} \mid \boldsymbol{z}) p(\boldsymbol{z} \mid y, u)\, \mathrm{d}\boldsymbol{z}$, it is evident that the prior $p(\boldsymbol{z} \mid y, u)$ is consistent with our SCM. Specifically, based on our SCM, $p(\boldsymbol{z} \mid y, u)$ can be further decomposed into a product of conditional probabilities, introducing additional sparsity in $\boldsymbol{\lambda}$. This distinction differentiates our learned model from existing works that are based on alternative SCMs [27, 28, 41].

However, since the causal graph is generally unknown *a priori*, we cannot predefine the sparsity of $\boldsymbol{\lambda}$. Consequently, in our learning algorithm, we treat $p(\boldsymbol{z} \mid y, u)$ as a generic prior consistent with **Assumption 1(b)**. Without a fully specified causal graph, we use this generic prior to guiding the iVAE training, obtaining $\boldsymbol{Z}$ without prior knowledge of their causal identities. The assumptions and identifiability results are summarized in **Theorem 1**.

Theorem 1. *Assume the data is sampled from a generative model described by*

$$p_{\boldsymbol{\xi}=(\boldsymbol{\theta},\boldsymbol{T},\boldsymbol{\lambda})}(\boldsymbol{X}, \boldsymbol{Z}|Y,U) = p_{\boldsymbol{\theta}}(\boldsymbol{X}|\boldsymbol{Z}) p_{\boldsymbol{T},\boldsymbol{\lambda}}(\boldsymbol{Z}|Y,U)$$
$$p_{\boldsymbol{\theta}}(\boldsymbol{X}|\boldsymbol{Z}) = p_{\boldsymbol{\epsilon}}(\boldsymbol{X} - g_{\boldsymbol{\theta}}(\boldsymbol{Z}))$$

Assume the following holds: (i) Denote the characteristic function of $p_{\boldsymbol{\epsilon}}$ as $\varphi_{\boldsymbol{\epsilon}}$, $\{\boldsymbol{X}|\varphi_{\boldsymbol{\epsilon}}(\boldsymbol{X}) = 0\}$ has measure zero. (ii) g has second-order cross derivatives and is injective. (iii) The sufficient statistics $\boldsymbol{T}(\boldsymbol{Z}) = [T_1(Z_1)^T, \cdots, T_N(Z_N)]^T$ have all second-order own derivatives, and all the $T_i(Z_i)$ have dimension larger or equal to 2. (iv) There exist $k+1$ distinct points $(Y^0, U^0), (Y^1, U^1), \cdots, (Y^k, U^k)$ such that the matrix $L = \Big(\boldsymbol{\lambda}(Y^1, U^1) - \boldsymbol{\lambda}(Y^0, U^0), \cdots, \boldsymbol{\lambda}(Y^k, U^k) - \boldsymbol{\lambda}(Y^0, U^0)\Big)$ of size $k \times k$ is invertible, where k is the dimension of $\boldsymbol{T}$. Then, the following holds: $\boldsymbol{\xi}$ is identifiable up to a permutation and component-wise transformation.

Theorem 1 is the same to **Theorem 1** in NF-iVAE. Please refer to [41] for detailed proof. Our contribution is not in introducing new assumptions to prove the component-wise identifiability of the latent variables. Instead, we aim to demonstrate that the iVAE can be trained using the same general prior distribution $p(\boldsymbol{Z} \mid Y, U)$, as it adheres to the data generation process defined by our proposed SCM.

As discussed in Section 4, we use a different encoder distribution, $q(\boldsymbol{Z} \mid \boldsymbol{X})$, rather than the $q(\boldsymbol{Z} \mid \boldsymbol{X}, Y, U)$ used in NF-iVAE. Consequently, we derive a distinct theorem for obtaining the true parameters $\boldsymbol{\xi}^*$ within our proposed learning framework.

Theorem 2. *Assume the following assumptions hold: (i) The family of distributions $q_{\boldsymbol{\psi}}(\boldsymbol{Z}|\boldsymbol{X})$ contains $p_{\boldsymbol{\xi}}(\boldsymbol{Z}|\boldsymbol{X}, Y, U)$, and $q_{\boldsymbol{\psi}}(\boldsymbol{Z}|\boldsymbol{X}) > 0$ everywhere. (ii) The NF-iVAE learning framework, which minimizes $\mathcal{L}_{iVAE}(\boldsymbol{\psi}, \boldsymbol{\xi})$ in Eq. (29) with respect to both $\boldsymbol{\xi}$ and $\boldsymbol{\psi}$, can learn the true parameters $\boldsymbol{\xi}^*$ up to a permutation and simple transformation of the latent variable $\boldsymbol{Z}$ in the limit of infinite data.*

Proof. We recall from the loss function in Phase I is as follows:

$$\mathcal{L}_{\text{iVAE}}(\boldsymbol{\theta}, \boldsymbol{\psi}, \boldsymbol{T}, \boldsymbol{\lambda}) \coloneqq \mathcal{L}_{\text{ELBO}}(\boldsymbol{\theta}, \boldsymbol{\psi}, \hat{\boldsymbol{T}}, \hat{\boldsymbol{\lambda}}) + \mathcal{L}_{\text{SM}}(\hat{\boldsymbol{\theta}}, \hat{\boldsymbol{\psi}}, \boldsymbol{T}, \boldsymbol{\lambda}) \quad (29)$$

[4] Arbitrary function $\boldsymbol{\lambda}$ and sufficient statistics $\boldsymbol{T}$ are modeled by neural networks with ReLU activation due to their universal approximation ability.

$$\mathcal{L}_{\text{ELBO}} \coloneqq -\mathbb{E}_{p_{\mathcal{D}}}\left[\mathbb{E}_{q_{\boldsymbol{\psi}}(\boldsymbol{z}|\boldsymbol{x})}[\log p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z}) + \log p_{\boldsymbol{T},\boldsymbol{\lambda}}(\boldsymbol{z}|y,u) - \log q_{\boldsymbol{\psi}}(\boldsymbol{z}|\boldsymbol{x})]\right] \tag{30}$$

$$\mathcal{L}_{\text{SM}} \coloneqq \mathbb{E}_{p_{\mathcal{D}}}\left[\mathbb{E}_{q_{\boldsymbol{\psi}}(\boldsymbol{z}|\boldsymbol{x})}[\|\nabla_{\boldsymbol{z}} \log q_{\boldsymbol{\psi}}(\boldsymbol{z}|\boldsymbol{x}) - \nabla_{\boldsymbol{z}} \log p_{\boldsymbol{T},\boldsymbol{\lambda}}(\boldsymbol{z}|y,u)\|^2]\right] \tag{31}$$

If the family of $q_{\boldsymbol{\psi}}(\boldsymbol{Z}|\boldsymbol{X})$ is flexible enough to contain $p_{\boldsymbol{\xi}}(\boldsymbol{Z}|\boldsymbol{X},Y,U)$, then by optimizing the $\mathcal{L}_{\text{iVAE}}$ over its parameter $\boldsymbol{\xi}$, the score matching term $\mathcal{L}_{\text{SM}}$ is minimized and eventually reach zero. If we assume that the model is not degenerate and that $q_{\boldsymbol{\psi}} > 0$ everywhere, then we have

$$\begin{aligned} \mathcal{L}_{\text{SM}} = 0 &\implies \nabla_{\boldsymbol{z}} \log q_{\boldsymbol{\psi}}(\boldsymbol{z}|\boldsymbol{x}) = \nabla_{\boldsymbol{z}} \log p_{\boldsymbol{T},\boldsymbol{\lambda}}(\boldsymbol{z}|y,u) \\ &\implies \log q_{\boldsymbol{\psi}}(\boldsymbol{z}|\boldsymbol{x}) = \log p_{\boldsymbol{T},\boldsymbol{\lambda}}(\boldsymbol{z}|y,u) + c \end{aligned} \tag{32}$$

for some constant c. c is zero because both $q_{\boldsymbol{\psi}}(\boldsymbol{z}|\boldsymbol{x})$ and $p_{\boldsymbol{T},\boldsymbol{\lambda}}(\boldsymbol{z}|y,u)$ are pdf's. Therefore, the $\mathcal{L}_{\text{iVAE}}$ will be equal to the log-likelihood. Under such circumstances, the estimation in Eq. (29) inherits all the properties of maximum likelihood estimation (MLE). Since our identifiability is guaranteed up to a permutation and componentwise transformation, the consistency of MLE indicates that we converge to the true parameters $\boldsymbol{\xi}^*$ up to a permutation and componentwise transformation in the limit of infinite data. □

Automatically, we will have Theorem 3 that proves the identifiability of learned $\boldsymbol{Z}^*$.

Theorem 3. *Assume that Theorem 1 and Theorem 2 hold, then in the limit of infinite data, the true latent variables $\boldsymbol{Z}^*$ are identifiable up to a permutation and componentwise transformation.*

Proof. **Theorem 1** and **Theorem 2** guarantee that in the limit of infinite data, iVAE can obtain the true parameters $\boldsymbol{\xi}^* \coloneqq (\boldsymbol{\theta}^*, \boldsymbol{T}^*, \boldsymbol{\lambda}^*)$ up to a permutation and componentwise transformation of the latent variables. We denote the parameters obtained from NF-iVAE as $\hat{\boldsymbol{\xi}} \coloneqq (\hat{\boldsymbol{\theta}}, \hat{\boldsymbol{T}}, \hat{\boldsymbol{\lambda}})$, i.e., $(\hat{\boldsymbol{\phi}}, \hat{\boldsymbol{T}}, \hat{\boldsymbol{\lambda}})$ and $(\boldsymbol{\phi}^*, \boldsymbol{T}^*, \boldsymbol{\lambda}^*)$ are identifiable up to a permutation and component-wise transformation. If there were no noise, we have $\hat{\boldsymbol{Z}} = g^{-1}_{\hat{\boldsymbol{\theta}}}(\boldsymbol{X})$ that are equal to $\boldsymbol{Z}^* = g^{-1}_{\boldsymbol{\theta}^*}(\boldsymbol{X})$ up to a permutation and componentwise transformation. If with noise, we can obtain the posterior distribution of the latent variables up to an analogous indeterminacy. □

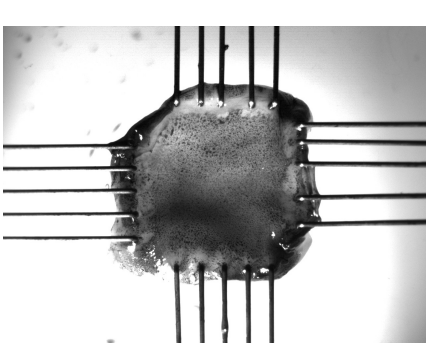
B.3. BCD Related Work

Bayesian Causal Discovery. Causal discovery is the task of inferring causal structure from observational data, and has traditionally been approached through several primary lenses: constraint-based methods, which rely on conditional independence tests, and score-based methods, which search for a graph that maximizes a scoring functions. While existing constraint-based and score-based methods are efficient, they typically yield a single estimated graph without quantifying uncertainty. In contrast, Bayesian causal discovery aims to *infer a full posterior distribution over directed acyclic graphs (DAGs)*, $P(\mathcal{G}|\mathcal{D})$, thereby capturing the epistemic uncertainty inherent in finite data.

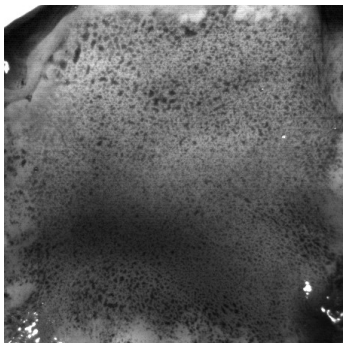
MCMC and order/partial-order samplers. Early and influential Bayesian methods use Markov chain Monte Carlo (MCMC) to explore graph space. Structure-MCMC [43] directly proposes local edge additions/deletions/reversals, but mixes poorly because the space of DAGs is vast and discrete. To improve mixing, methods that sample over node orderings (or partial orders) [17] have been proposed; sampling orders reduces the complexity of the search space and yields more regular posterior landscapes. These order-based MCMC approaches (and Metropolis-coupled variants such as MC^3) remain a practical baseline for posterior approximation and model averaging.

Differentiable and Variational Methods. The advent of continuous optimization for causal discovery, pioneered by Zheng et al. [69], reformulated the discrete combinatorial problem of DAG learning into a continuous optimization task. This relaxation paved the way for differentiable Bayesian methods that scale significantly better than MCMC. DiBS [40] maps the discrete space of DAGs to a continuous latent space and uses Stein variational gradient descent to approximate the posterior. Similarly, BCD Nets [14] employ variational inference, parameterizing the distribution over DAGs using a specialized factorization of the adjacency matrix to enable efficient stochastic optimization. More recently, BayesDAG [7] has combined the strengths of stochastic gradient MCMC with variational inference to improve posterior approximation accuracy while maintaining scalability, avoiding the restrictive DAG regularization penalties used in prior works.

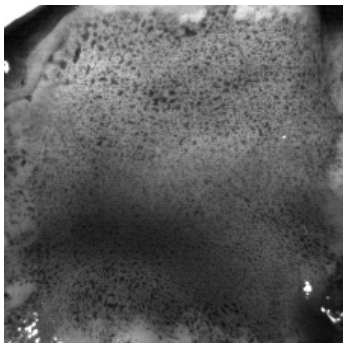
Generative Flow Networks (GFlowNets). A recent paradigm shift involves the use of Generative Flow Networks (GFlowNets) for causal discovery. Unlike MCMC methods that rely on local moves, GFlowNets learn a stochastic policy to construct DAGs sequentially, treating graph generation as a flow-matching problem. DAG-GFlowNet [15] demonstrates that this approach can sample proportional to the posterior distribution more efficiently than MCMC, particularly in identifying multimodal posteriors where traditional samplers might get stuck.



(a) Raw acquisition frame (960×1280).



(b) Extracted region of interest (450×450).



(c) Standardized model input (224×224).

Figure 4. Visualization of the data preprocessing pipeline for the Biaxial Loading Test (BLT) dataset. The raw high-resolution inputs (Figure 4(a)) are processed to isolate the region of interest (Figure 4(b)) and standardized for model training (Figure 4(c)).

C. Additional Empirical Results

C.1. Implementation Details

For the CMNIST dataset, we adopt the architectural configurations from one of our baselines, as described by Lu et al. [41], utilizing multilayer perceptrons (MLP) as both encoder and decoder components. For BLT, PACS, VLCS, and OfficeHome datasets, we leverage a pre-trained ResNet-50 on ImageNet as the encoder backbone, complemented by a decoder of comparable complexity. For the classifier, we use the same 2-layer MLP as [41, 45]. The main results include mean and standard deviation over 5 runs with different initialization. All experiments are conducted on a 5090 GPU. It is worth noting that our propose framework is backbone-agnostic and extensible to high-resolution architectures like transformers or diffusion models. We utilize the iVAE for its identifiability, providing theoretical gaurantees for the reproducibility and transferability of learned causal features.

C.2. CMNIST Data Settings

We follow the data generation procedure in Arjovsky et al. [8] and add noise to the preliminary label by flipping it with 25 percent probability to construct the final labels. The label is set to 0 if the digit is between $0-4$ and 1 if the digit is between $5-9$. We sample the color label by flipping the final labels with probability p_e, where $p_e = 0.2$ in one training domain and $p_e = 0.1$ in another. We set $p_e = 0.9$ in the test domain. We color the digit red if the color label is 1 and green if it is 0.

C.3. Biaxial Loading Test Data

To further evaluate CGU-BAYES on challenging tasks, we construct a dataset from real-world experiment on biaxial loading test (BLT) (https://github.com/fishmoon1234/Biaxial_Loading_Test_VisionTask). Biaxial testing is a popular technique to evaluate the mechanical response of materials. In this test, a material specimen is subjected to increasing simultaneous stress along two perpendicular axes, and a camera will record the corresponding deformation of this material. As such, each loading stress would produce one image. To provide a thorough picture of the mechanical property and microstructure of complex materials, usually multiple protocols (corresponding to different fixed biaxial stress ratios) are needed. However, for delicate materials such as biotissues, damage occurs in the microscale during the testing procedure. Hence, it is of interest to reduce the number of samples in the later protocols, so as to minimize the specimen damage and corresponding errors.

Table 4. Statistical breakdown of the processed dataset.

Domain Index	Stress Ratio	# of observations
1	1:1	3918
2	1:0.66	3798
3	1:0.33	3540
4	0.66:1	4014
5	0.33:1	4176
6	0.05:1	3540
7	1:0.1	3540
Total	-	26526

In this dataset, we have conducted biaxial loading test and collected image/stress data pairs from a representative tricuspid valve anterior leaflet (TVAL) specimen from a porcine heart. Seven biaxial stress ratio protocols ($P_{11} : P_{22} = 1:1, 1:0.66, 1:0.33, 0.66:1, 0.33:1, 0.05:1, 1:0.1$) were performed, with 3539∼4175 samples in each protocol and 26,524 samples in total. The details are provided in Table 4. Each protocol will be treated as a domain. The data between different domains is anticipated to have distribution shift due to the change of loading ratio. The goal is to predict the corresponding stress along the x and y axis, using the recorded image. Beyond the

challenge from limited samples in each domain, this dataset also possess challenge due to the graduate microscale damage on the tissue. As a result, the tissue would be loosen and deform permanently in later protocols, introducing non-uniform variability and heteroskedastic noise across different domains.

As illustrated in Figure 4, data preprocessing for the BLT dataset involved a two-step transformation to standardize the visual inputs. We first cropped the original 960×1280 high-resolution images (Figure 4(a)) to a central 450×450 region of interest (Figure 4(b)), effectively eliminating background noise and focusing exclusively on the essential tissue sections. These cropped segments were subsequently resized to the standard resolution of 224×224 pixels (Figure 4(c)) to align with model input requirements. The downstream task is formulated as a multi-target regression problem, where the model utilizes these processed images to predict the corresponding mechanical load components, specifically corresponding stresses along the x and y axis.

C.4. Detailed Empirical Results for PACS, VLCS, and OfficeHome

We provided the detailed empirical results for each domain of VLCS, PACS, and Officehome datasets in table 5 and 6. For algorithms with read-to-use implementations, we run the algorithms for 5 trials and report the mean and std in the tables. For algorithms without implementations, such as iCaRL and PTG, we use the reported results in the original paper. We provide the baseline implementations as follows:

- IRM, F-IRM GAME, V-IRM GAME: We employ the implementation from the original paper [3](https://github.com/IBM/OoD).
- CTRANS: We employ the implementation from its original paper [45](https://github.com/cvlab-columbia/CT4Recognition).
- CAUSALREP: We employ the implementation from its original paper [60](https://github.com/yixinwang/representation-causal-public).
- FACT: We employ the implementation from its original paper [63] (https://github.com/MediaBrain-SJTU/FACT).
- CIRL: We employ the implementation from its original paper [42] (https://github.com/BIT-DA/CIRL).
- CASN: We employ the implementation from its original paper [65] (https://github.com/ymy4323460/CaSN).
- BITEBAYES: We employ the implementation from its original paper [62] (https://github.com/zzzx1224/A-Bit-More-Bayesian).
- The other baselines are from the domainBed package (https://github.com/facebookresearch/DomainBed).

C.5. Identifiability of Representations

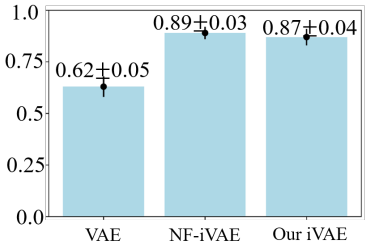


Figure 5. Average MCC on CMNIST

To reuse the learned encoder distribution for uncertainty estimation and Bayesian inference, we replace $q(\boldsymbol{z}|\boldsymbol{x}, y, u)$ in the original NF-iVAE with $q(\boldsymbol{z}|\boldsymbol{x})$. We empirically assess the identifiability of our estimated representations using $q(\boldsymbol{z}|\boldsymbol{x})$ on the CMNIST dataset, comparing it to the unidentifiable VAE [38] and NF-iVAE [41]. Following the standard protocol in Khemakhem et al. [27], we compute the mean correlation coefficient (MCC) between representations learned by models with different random initializations, where higher MCC values indicate stronger identifiability. Representation dimensionality is set to $|\boldsymbol{Z}| = 10$. The results in Figure 5 show that replacing the encoder with a less expressive variant results in a slight reduction in identifiability; however, our estimated representations still achieve a high degree of identifiability, significantly outperforming the unidentifiable VAE.

C.6. Ablation Study on Hyperparameters

We conduct an extensive ablation study to investigate: (1) the sensitivity of prediction performance to the number of sampled causal graphs L; (2) alternative metrics for quantifying the relationship between uncertainty $\mathcal{U}_e(\boldsymbol{x}^t|\mathcal{G})$ and $p(\boldsymbol{x}^t|\mathcal{G})$; (3) the impact of different Bayesian causal discovery approaches; and (4) the sensitivity to the latent representation dimensionality $N = |\boldsymbol{Z}|$.

We also conduct a brief ablation study on α in Eq. (3). In practice, choosing α only requires the estimated uncertainties, not ground-truth labels. By scaling $\mathcal{U}_e(\boldsymbol{x}^t|\mathcal{G})$ with α, we avoid two extremes in the weights: nearly uniform and overly high-contrast. We test $\alpha \in \{0.1, 0.5, 1, 2, 5\}$ and find that $\alpha = 1$ typically yields robust performance across all classification tasks.

Sensitivity of the number of sampled causal graph L. We perform the ablation study on the CMNIST dataset, employing NF-iVAE with $N = |\boldsymbol{Z}| = 10$ in Phase 1 and DAG-GFlowNet in Phase 2. In Phase 3, we varied the number

Table 5. Empirical comparison on the VLCS and PACS benchmarks reporting OOD prediction accuracy (%). The optimal performance is indicated in bold.

Algorithms	VLCS					PACS				
	C	L	S	V	Avg	A	C	P	S	Avg
ERM	98.0±0.4	62.6±0.9	70.8 ±1.9	77.5 ±1.9	77.2	84.8±1.3	76.4±1.1	96.7±0.6	76.1±1.0	83.5
GROUPDRO	98.1± 0.3	66.4 ± 0.9	71.0 ± 0.3	76.1 ± 1.4	77.9	83.5±0.9	79.1±0.6	96.7±0.3	78.3±2.0	84.4
MLDG	98.5± 0.3	61.7± 1.2	73.6± 1.8	75.0± 0.8	77.2	85.5±1.4	80.1±1.7	97.4±0.3	76.6±1.1	84.9
CORAL	96.9± 0.9	65.7± 1.2	73.3± 0.7	78.7± 0.8	78.7	88.3±0.2	80.0±0.7	97.5±0.3	78.8±1.3	86.2
MMD	98.3± 0.1	65.6± 0.7	69.7± 1.0	75.7± 0.9	77.3	86.1±1.4	79.4±0.9	96.6±0.2	76.5±0.7	84.6
RSC	97.5± 0.6	63.1± 1.2	73.0± 1.3	76.2± 0.5	77.5	85.4±0.8	79.7±1.8	97.6±0.3	78.2±1.2	85.2
MIXUP	98.4± 0.3	63.4± 0.7	72.9± 0.8	76.1± 1.2	77.7	86.1±0.7	78.9±0.8	97.6±0.1	75.8±1.8	84.6
DANN	98.5± 1.3	64.9± 1.3	72.6± 1.4	78.7± 1.7	78.2	86.4±0.8	77.4±0.8	97.3±0.4	73.5±2.3	83.6
CDANN	97.6± 0.6	65.2± 0.8	73.4± 1.4	76.9± 0.5	78.3	84.6±1.8	75.5±0.9	96.8±0.3	73.5±0.6	82.6
MTL	97.6± 0.6	60.6± 1.3	71.0± 1.2	77.2± 0.7	76.6	87.5±0.8	77.1±0.7	96.4±0.8	77.3±1.8	84.6
ARM	97.2± 0.5	62.7± 1.	70.6± 0.6	75.8± 0.9	76.6	86.8±0.6	76.8±0.7	97.4±0.3	79.3±1.2	85.1
IRM	98.6±0.1	66.0 ±0.9	72.3 ±0.6	77.3 ±0.9	78.5	84.7±0.4	80.0±0.6	97.2±0.3	79.3±1.0	85.5
SAGNET	97.3± 0.4	61.6± 0.8	73.4± 1.9	77.6± 0.4	77.5	87.4±1.0	80.7±0.6	97.1±0.1	80.0±0.4	86.3
ICARL	-	-	-	-	81.8	-	-	-	-	88.7
FACT	97.5± 0.5	65.7± 0.7	72.6± 0.8	77.4± 0.4	78.3	**90.9** ±0.4	83.6 ±0.6	**97.8** ±0.1	86.2 ±0.7	89.6
CIRL	98.5±0.4	66.3 ±1.2	72.6 ±0.7	77.2 ±0.8	78.5	90.7 ±0.2	84.3 ±0.7	**97.8** ±0.5	**87.7** ±0.8	90.1
CASN	98.1 ±0.3	67.5±0.8	72.9±0.7	78.3±0.9	79.1	88.5±0.6	83.2±1.0	97.2±0.3	80.4±0.5	87.3
CMBRL	98.7 ±0.3	71.2±0.4	77.1±0.7	82.1±0.5	82.3	89.2±0.7	85.3±1.5	97.7±0.5	84.1±0.5	89.1
BITEBAYES	97.3± 0.2	67.2± 0.1	73.0± 0.2	78.8± 0.1	79.1	83.9±0.7	81.6±81.6	96.0±0.2	80.3±0.9	85.5
PTG	-	-	-	-	76.1	-	-	-	-	83.7
CGU-BAYES	98.6±0.1	69.6 ±0.7	79.5±0.4	**82.3**±0.4	82.5	89.2±0.7	**85.6**±1.2	97.6±0.5	83.6±0.6	89.0
CGU-BAYES++	**98.8**±0.2	**70.3** ±0.4	**80.2**±0.3	**82.3**±0.2	**82.9**	**90.9**±0.5	**85.6**±1.0	**97.8**±0.3	**87.7**±0.7	**90.5**

Table 6. Empirical results on OfficeHome datasets in terms of OOD prediction accuracy (%). The optimal performance is indicated in bold.

Algorithms	Domains				Avg
	A	C	P	R	
ERM	61.3 ± 0.7	52.4 ± 0.3	75.8 ± 0.1	76.6 ± 0.3	66.5
GROUPDRO	60.4 ± 0.7	52.7 ± 1.0	75.0 ± 0.7	76.0 ± 0.7	66.0
MLDG	61.5 ± 0.9	53.2 ± 0.6	75.0 ± 1.2	77.5 ± 0.4	66.8
CORAL	65.3 ± 0.4	54.4 ± 0.5	76.5 ± 0.1	78.4 ± 0.5	68.7
MMD	60.4 ± 0.2	53.3 ± 0.3	74.3 ± 0.1	77.4 ± 0.6	66.3
RSC	60.7 ± 1.4	51.4 ± 0.3	74.8 ± 1.1	75.1 ± 1.3	65.5
MIXUP	62.4 ± 0.8	54.8 ± 0.6	77.3 ± 0.3	**79.2** ±0.2	68.4
DANN	59.9 ± 1.3	53.0 ± 0.3	73.6 ± 0.7	76.9 ± 0.5	65.9
CDANN	61.5 ± 1.4	50.4 ± 2.4	74.4 ± 0.9	76.6 ± 0.8	65.8
MTL	61.5 ± 0.7	52.4 ± 0.6	74.9 ± 0.4	76.8 ± 0.4	66.4
ARM	58.9 ± 0.8	51.0 ± 0.5	74.1 ± 0.1	75.2 ± 0.3	64.8
IRM	58.9 ± 2.3	52.2 ± 1.6	72.1 ± 2.9	74.0 ± 2.5	64.3
SAGNET	**63.4**±0.2	54.8± 0.4	75.8± 0.4	78.3± 0.3	68.1
CIRL	61.5± 0.2	55.3± 0.4	75.1± 0.7	76.6± 0.5	67.1
FACT	60.3± 0.4	54.9± 0.5	74.5± 0.9	76.5± 0.5	66.6
CASN	60.7± 1.0	53.5± 0.6	74.9± 0.4	76.5± 0.7	66.4
CMBRL	61.3± 0.8	54.9± 0.4	76.5± 0.4	75.3± 1.1	67.0
BITEBAYES	61.8 ±0.4	53.3±0.4	74.3±0.4	76.3±0.2	66.4
PTG	-	-	-	-	61.6
CGU-BAYES	63.1 ±0.1	**56.9**±0.2	**78.8**±0.2	79.1±0.1	**69.5**
CGU-BAYES++	**63.4** ±0.2	56.7±0.3	78.4±0.2	**79.2**±0.4	**69.5**

of sampled graphs used to construct the predictors, specifically setting $L \in \{3, 5, 10, 15\}$. Table 7 summarizes the in-distribution and OOD prediction performance. The results indicate that in-distribution accuracy remains stable across different values of L. We attribute this to the fact that the graph posterior is conditioned on in-distribution data, yielding sampled graphs that are structurally similar and equally effective. Conversely, OOD performance improves with larger L. Table 7 reveals that while in-distribution performance is insensitive to L, OOD robustness benefits significantly from a higher sample count. This dichotomy arises because the posterior is sharpened by in-distribution data; thus, small samples suffice for internal consistency. However, for OOD generalization, a larger L increases the probability of recovering the invariant causal mechanism essential for accurate CMB representation. In practice, we set $L = 10$ for all datasets.

Table 7. Ablation study of varying number of graph samples and functions for quantifying $p(\boldsymbol{x}^t|\mathcal{G})$ using $\mathcal{U}_e(\boldsymbol{x}^t|\mathcal{G})$.

Methods	L	In-distribution Accuracy		OOD Accuracy	
		$\frac{1}{\mathcal{U}_e(\boldsymbol{x}^t\|\mathcal{G})}$	$e^{-\alpha\mathcal{U}_e(\boldsymbol{x}^t\|\mathcal{G})}$	$\frac{1}{\mathcal{U}_e(\boldsymbol{x}^t\|\mathcal{G})}$	$e^{-\alpha\mathcal{U}_e(\boldsymbol{x}^t\|\mathcal{G})}$
CGU-BAYES(W/O UQ)	-	70.4		55.8	
CGU-BAYES	3	72.7	72.5	63.8	63.3
	5	72.1	72.5	65.1	67.7
	10	72.5	**72.8**	66.5	**70.5**
	15	72.4	72.5	66.5	69.5

Alternative approach for quantifying $p(\boldsymbol{x}^t|\mathcal{G})$ using $\mathcal{U}_e(\boldsymbol{x}^t|\mathcal{G})$. In addition to the exponential relationship between $p(\boldsymbol{x}^t|\mathcal{G})$ and $\mathcal{U}_e(\boldsymbol{x}^t|\mathcal{G})$ defined in Eq. (3), we investigated an alternative inverse formulation, $p(\boldsymbol{x}^t|\mathcal{G}) \propto \frac{1}{\mathcal{U}_e(\boldsymbol{x}^t|\mathcal{G})}$. The comparative results are presented in Table 7. As shown, the exponential function yields superior prediction accuracy. Empirically, this is because the exponential form assigns significantly sharper weights than the inverse function, allowing the model to more effectively prioritize CMB features that fit the test data well.

Impact of different Bayesian causal discovery approaches. We conducted this ablation study on the CMNIST dataset, utilizing NF-iVAE with $N = |\boldsymbol{Z}| = 32$ in Phase 1. We compared our chosen backbone, DAG-GFLOWNETS (a sequential decision-based approach), against several distinct baselines: MC3 (MCMC sampling),

DIBS and BCD-NETS (variational inference), and BAYESDAG (a mixture approach). For each method, we sampled $L = 10$ graphs from the estimated posterior to construct the corresponding OOD predictors. As shown in Table 8, DAG-GFLOWNETS outperforms all baselines with an OOD accuracy of 74.1%, justifying its adoption in our framework. Notably, differentiable methods such as DIBS and BCD-NETS learned significantly sparser graphs (selecting only 6 and 3 features, respectively) compared to DAG-GFLOWNETS (10 features). This tendency toward over-sparsity in differentiable baselines is consistent with results reported by Deleu et al. [15] on real-world Sachs data.

Table 8. Comparison of OOD prediction accuracy and feature sparsity across different Bayesian causal discovery methods.

Methods	# Selected Features	OOD Accuracy (%)
MC3	15/32	21.7
DIBS	6/32	69.5
BCD-NETS	3/32	58.5
BAYESDAG	11/32	60.8
DAG-GFLOWNETS	10/32	**74.1**

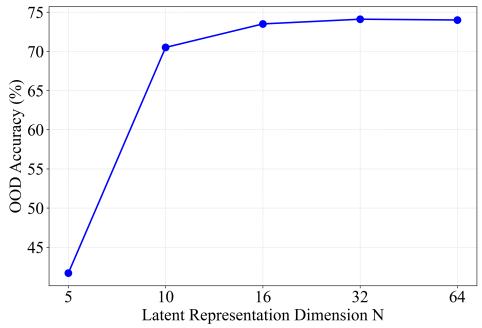


Figure 6. Sensitivity analysis of OOD prediction accuracy with respect to the latent representation dimensionality N on the CMNIST dataset.

Sensitivity to the latent representation dimensionality N. We conducted this ablation study on the CMNIST dataset, varying the latent dimension $N \in \{5, 10, 16, 32, 64\}$ within the NF-iVAE (Phase 1). Our results in Figure 6 indicate that higher dimensionality generally yields more robust OOD performance. We attribute this to the increased capacity of a larger $\boldsymbol{Z}$ to capture information from $\boldsymbol{X}$, which promotes better disentanglement of distinct semantic concepts. This disentanglement is crucial, as it facilitates the separation of invariant CMB features from spurious ones in subsequent steps. However, we observed diminishing returns; once $|\boldsymbol{Z}|$ is sufficiently large, further increases do not guarantee improved identifiability or information incorporation. As optimal selection is challenging and data-dependent, we empirically set $N = 32$ for CMNIST and $N = 50$ for the BLT, PACS, VLCS, and OfficeHome datasets.

C.7. Efficiency Justification

We provide the wall-clock training times and inference latencies for CGU-Bayes and the CMBRL baseline on the CMNIST. For CGU-Bayes, the training time for NF-iVAE, BCD, and the L classifiers training are approximately 21, 18, and 15 minutes, with an inference latency of 1.4e-2s per sample. In comparison, CMBRL requires 21 minutes for iVAE, 10 minutes for CMB feature selection, and 11 minutes for its single classifier, with an inference latency of 1.1e-2s. Notably, although our approach utilizes L classifiers, we optimize their execution in **parallel** rather than sequentially. This ensures that the performance gap remains marginal (15 vs. 11 minutes for training; 1.4e-2 vs. 1.1e-2 s for inference). While the BCD component is more computationally intensive than the deterministic selection in CMBRL, the Bayesian framework is essential for sampling accurate CMB estimations, particularly for challenging data. We contend that in challenging cases where predictive accuracy is the primary goal, this slight efficiency overhead is a well-justified trade-off.

References

[1] Alessandro Achille and Stefano Soatto. Emergence of invariance and disentanglement in deep representations. *The Journal of Machine Learning Research*, 19(1):1947–1980, 2018. 3

[2] Idan Achituve, Idit Diamant, Arnon Netzer, Gal Chechik, and Ethan Fetaya. Bayesian uncertainty for gradient aggregation in multi-task learning. *arXiv preprint arXiv:2402.04005*, 2024. 2

[3] Kartik Ahuja, Karthikeyan Shanmugam, Kush Varshney, and Amit Dhurandhar. Invariant risk minimization game. In *International Conference on Machine Learning*, 2020. 3, 8, 7

[4] Kartik Ahuja, Ethan Caballero, Dinghuai Zhang, Yoshua Bengio, Ioannis Mitliagkas, and Irina Rish. Invariance principle meets information bottleneck for out-of-distribution generalization. In *Advances in Neural Inf. Proc. Systems*, 2021.

[5] Kartik Ahuja, Jun Wang, Amit Dhurandhar, Karthikeyan Shanmugam, and Kush R. Varshney. Empirical or invariant risk minimization? a sample complexity perspective. In *International Conference on Learning Representations*, 2021. 3

[6] Mark Andrews and Thom Baguley. Bayesian data analysis. *The Cambridge encyclopedia of child development*, pages 165–169, 2017. 2, 4

[7] Yashas Annadani, Nick Pawlowski, Joel Jennings, Stefan Bauer, Cheng Zhang, and Wenbo Gong. Bayesdag: Gradient-based posterior inference for causal discovery. *Advances in Neural Information Processing Systems*, 36:1738–1763, 2023. 5

[8] Martin Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz. Invariant risk minimization. *arXiv preprint arXiv:1907.02893*, 2019. 1, 3, 6, 7

[9] Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1798–1828, 2013. 3

[10] Christopher P Burgess, Irina Higgins, Arka Pal, Loic Matthey, Nick Watters, Guillaume Desjardins, and Alexander Lerchner. Understanding disentangling in β-vae. *arXiv preprint arXiv:1804.03599*, 2018.

[11] Ricky TQ Chen, Xuechen Li, Roger B Grosse, and David K Duvenaud. Isolating sources of disentanglement in variational autoencoders. *Advances in neural information processing systems*, 31, 2018. 3

[12] Peng Cui and Susan Athey. Stable learning establishes some common ground between causal inference and machine learning. *Nature Machine Intelligence*, 4(2):110–115, 2022. 3

[13] Peng Cui, Zheyan Shen, Sheng Li, Liuyi Yao, Yaliang Li, Zhixuan Chu, and Jing Gao. Causal inference meets machine learning. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 3527–3528, 2020. 3

[14] Chris Cundy, Aditya Grover, and Stefano Ermon. Bcd nets: Scalable variational approaches for bayesian causal discovery. *Advances in Neural Information Processing Systems*, 34:7095–7110, 2021. 5

[15] Tristan Deleu, António Góis, Chris Emezue, Mansi Rankawat, Simon Lacoste-Julien, Stefan Bauer, and Yoshua Bengio. Bayesian structure learning with generative flow networks. In *Uncertainty in Artificial Intelligence*, pages 518–528. PMLR, 2022. 5, 6, 9

[16] Stefan Depeweg, Jose-Miguel Hernandez-Lobato, Finale Doshi-Velez, and Steffen Udluft. Decomposition of uncertainty in bayesian deep learning for efficient and risk-sensitive learning. In *International conference on machine learning*, pages 1184–1193. PMLR, 2018. 4

[17] Nir Friedman and Daphne Koller. Being bayesian about network structure. a bayesian approach to structure discovery in bayesian networks. *Machine learning*, 50(1):95–125, 2003. 5

[18] Pascal Germain, Francis Bach, Alexandre Lacoste, and Simon Lacoste-Julien. Pac-bayesian theory meets bayesian inference. *Advances in Neural Information Processing Systems*, 29, 2016. 2

[19] Dan Hendrycks, Norman Mu, Ekin D Cubuk, Barret Zoph, Justin Gilmer, and Balaji Lakshminarayanan. Augmix: A simple data processing method to improve robustness and uncertainty. *arXiv preprint arXiv:1912.02781*, 2019. 3

[20] Zeyi Huang, Haohan Wang, Eric P Xing, and Dong Huang. Self-challenging improves cross-domain generalization. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*, pages 124–140. Springer, 2020. 8

[21] Kasra Jalaldoust and Elias Bareinboim. Transportable representations for domain generalization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 12790–12800, 2024. 3

[22] Kasra Jalaldoust, Alexis Bellot, and Elias Bareinboim. Partial transportability for domain generalization. *Advances in Neural Information Processing Systems*, 37:137768–137805, 2024.

[23] Dominik Janzing. Causal regularization. *Advances in Neural Information Processing Systems*, 32, 2019.

[24] Yibo Jiang and Victor Veitch. Invariant and transportable representations for anti-causal domain shifts. *arXiv preprint arXiv:2207.01603*, 2022. 3

[25] Alex Kendall and Yarin Gal. What uncertainties do we need in bayesian deep learning for computer vision? *Advances in neural information processing systems*, 30, 2017. 4

[26] Salman Khan, Munawar Hayat, Syed Waqas Zamir, Jianbing Shen, and Ling Shao. Striking the right balance with uncertainty. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 103–112, 2019. 4

[27] 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. 3, 4, 7

[28] Ilyes Khemakhem, Ricardo Monti, Diederik Kingma, and Aapo Hyvarinen. Ice-beem: Identifiable conditional energy-based deep models based on nonlinear ica. *Advances in Neural Information Processing Systems*, 33:12768–12778, 2020. 4

[29] Polina Kirichenko, Pavel Izmailov, and Andrew Gordon Wilson. Last layer re-training is sufficient for robustness to spurious correlations. *arXiv preprint arXiv:2204.02937*, 2022. 3

[30] Lingjing Kong, Shaoan Xie, Weiran Yao, Yujia Zheng, Guangyi Chen, Petar Stojanov, Victor Akinwande, and Kun Zhang. Partial disentanglement for domain adaptation. In *International Conference on Machine Learning*, pages 11455–11472. PMLR, 2022. 1

[31] Masanori Koyama and Shoichiro Yamaguchi. When is invariance useful in an out-of-distribution generalization problem? *arXiv preprint arXiv:2008.01883*, 2020. 3

[32] Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy M Hospedales. Deeper, broader and artier domain generalization. In *Proceedings of the IEEE international conference on computer vision*, pages 5542–5550, 2017. 6

[33] Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy Hospedales. Learning to generalize: Meta-learning for domain generalization. In *Proceedings of the AAAI conference on artificial intelligence*, 2018. 3

[34] Xiaotong Li, Yongxing Dai, Yixiao Ge, Jun Liu, Ying Shan, and Ling-Yu Duan. Uncertainty modeling for out-of-distribution generalization. *arXiv preprint arXiv:2202.03958*, 2022. 2

[35] Quande Liu, Cheng Chen, Jing Qin, Qi Dou, and Pheng-Ann Heng. Feddg: Federated domain generalization on medical image segmentation via episodic learning in continuous frequency space. In *Proceedings of the IEEE/CVF Conference*

on Computer Vision and Pattern Recognition, pages 1013–1023, 2021. 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 *international conference on machine learning*, pages 4114–4124. PMLR, 2019. 3

[37] Francesco Locatello, Ben Poole, Gunnar Rätsch, Bernhard Schölkopf, Olivier Bachem, and Michael Tschannen. Weakly-supervised disentanglement without compromises. In *International Conference on Machine Learning*, pages 6348–6359. PMLR, 2020. 3

[38] Romain Lopez, Jeffrey Regier, Michael I Jordan, and Nir Yosef. Information constraints on auto-encoding variational bayes. *Advances in neural information processing systems*, 31, 2018. 7

[39] David Lopez-Paz, Robert Nishihara, Soumith Chintala, Bernhard Scholkopf, and Léon Bottou. Discovering causal signals in images. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6979–6987, 2017. 1

[40] Lars Lorch, Jonas Rothfuss, Bernhard Schölkopf, and Andreas Krause. Dibs: Differentiable bayesian structure learning. *Advances in Neural Information Processing Systems*, 34:24111–24123, 2021. 5

[41] Chaochao Lu, Yuhuai Wu, José Miguel Hernández-Lobato, and Bernhard Schölkopf. Invariant causal representation learning for out-of-distribution generalization. In *International Conference on Learning Representations*, 2021. 1, 3, 5, 6, 7, 4

[42] Fangrui Lv, Jian Liang, Shuang Li, Bin Zang, Chi Harold Liu, Ziteng Wang, and Di Liu. Causality inspired representation learning for domain generalization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8046–8056, 2022. 7

[43] David Madigan and Jeremy York. Bayesian graphical models for discrete data. *International Statistical Review / Revue Internationale de Statistique*, 63(2):215–232, 1995. 5

[44] Chengzhi Mao, Augustine Cha, Amogh Gupta, Hao Wang, Junfeng Yang, and Carl Vondrick. Generative interventions for causal learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3947–3956, 2021. 1

[45] Chengzhi Mao, Kevin Xia, James Wang, Hao Wang, Junfeng Yang, Elias Bareinboim, and Carl Vondrick. Causal transportability for visual recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7521–7531, 2022. 1, 7, 6

[46] Hyeonseob Nam, HyunJae Lee, Jongchan Park, Wonjun Yoon, and Donggeun Yoo. Reducing domain gap by reducing style bias. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8690–8699, 2021. 3

[47] Ian Osband, Zheng Wen, Seyed Mohammad Asghari, Vikranth Dwaracherla, Morteza Ibrahimi, Xiuyuan Lu, and Benjamin Van Roy. Epistemic neural networks. *Advances in Neural Information Processing Systems*, 36, 2024. 5, 1

[48] Jonas Peters, Peter Bühlmann, and Nicolai Meinshausen. Causal inference by using invariant prediction: identification and confidence intervals. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 78(5):947–1012, 2016. 1

[49] Fengchun Qiao and Xi Peng. Uncertainty-guided model generalization to unseen domains. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6790–6800, 2021. 2

[50] Shikai Qiu, Andres Potapczynski, Pavel Izmailov, and Andrew Gordon Wilson. Simple and fast group robustness by automatic feature reweighting. In *International Conference on Machine Learning*, pages 28448–28467. PMLR, 2023. 3

[51] Elan Rosenfeld, Pradeep Ravikumar, and Andrej Risteski. The risks of invariant risk minimization. In *International Conference on Learning Representations*, 2021. 3

[52] Shiori Sagawa, Pang Wei Koh, Tatsunori B Hashimoto, and Percy Liang. Distributionally robust neural networks for group shifts: On the importance of regularization for worst-case generalization. *arXiv preprint arXiv:1911.08731*, 2019. 1, 8

[53] Shiyu Shen, Bin Pan, Tianyang Shi, Tao Li, and Zhenwei Shi. Bayesian domain invariant learning via posterior generalization of parameter distributions. *arXiv preprint arXiv:2310.16277*, 2023. 2, 3, 7

[54] Xinwei Shen, Furui Liu, Hanze Dong, Qing Lian, Zhitang Chen, and Tong Zhang. Weakly supervised disentangled generative causal representation learning. *The Journal of Machine Learning Research*, 23(1):10994–11048, 2022. 2, 3

[55] Yang Shu, Zhangjie Cao, Chenyu Wang, Jianmin Wang, and Mingsheng Long. Open domain generalization with domain-augmented meta-learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9624–9633, 2021. 3

[56] Antonio Torralba and Alexei A Efros. Unbiased look at dataset bias. In *CVPR 2011*, pages 1521–1528. IEEE, 2011. 6

[57] Hemanth Venkateswara, Jose Eusebio, Shayok Chakraborty, and Sethuraman Panchanathan. Deep hashing network for unsupervised domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5018–5027, 2017. 6

[58] Riccardo Volpi, Hongseok Namkoong, Ozan Sener, John C Duchi, Vittorio Murino, and Silvio Savarese. Generalizing to unseen domains via adversarial data augmentation. *Advances in neural information processing systems*, 31, 2018. 3

[59] Haotao Wang, Chaowei Xiao, Jean Kossaifi, Zhiding Yu, Anima Anandkumar, and Zhangyang Wang. Augmax: Adversarial composition of random augmentations for robust training. *Advances in neural information processing systems*, 34: 237–250, 2021. 3

[60] Yixin Wang and Michael I Jordan. Desiderata for representation learning: A causal perspective. *arXiv preprint arXiv:2109.03795*, 2021. 7

[61] Zifan Wang, Nan Ding, Tomer Levinboim, Xi Chen, and Radu Soricut. Improving robust generalization by direct

pac-bayesian bound minimization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16458–16468, 2023. 2

[62] Zehao Xiao, Jiayi Shen, Xiantong Zhen, Ling Shao, and Cees Snoek. A bit more bayesian: Domain-invariant learning with uncertainty. In *International Conference on Machine Learning*, pages 11351–11361. PMLR, 2021. 2, 3, 7

[63] Qinwei Xu, Ruipeng Zhang, Ya Zhang, Yanfeng Wang, and Qi Tian. A fourier-based framework for domain generalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14383–14392, 2021. 7

[64] Shen Yan, Huan Song, Nanxiang Li, Lincan Zou, and Liu Ren. Improve unsupervised domain adaptation with mixup training. *arXiv preprint arXiv:2001.00677*, 2020. 8

[65] Mengyue Yang, Yonggang Zhang, Zhen Fang, Yali Du, Furui Liu, Jean-Francois Ton, Jianhong Wang, and Jun Wang. Invariant learning via probability of sufficient and necessary causes. *Advances in Neural Information Processing Systems*, 36, 2024. 7

[66] Naiyu Yin, Hanjing Wang, Yue Yu, Tian Gao, Amit Dhurandhar, and Qiang Ji. Integrating markov blanket discovery into causal representation learning for domain generalization. In *European Conference on Computer Vision*, 2024. 1, 3, 6, 7

[67] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6023–6032, 2019. 3

[68] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017. 3

[69] Xun Zheng, Chen Dan, Bryon Aragam, Pradeep Ravikumar, and Eric Xing. Learning sparse nonparametric dags. In *International Conference on Artificial Intelligence and Statistics*, pages 3414–3425. PMLR, 2020. 5