Robust Domain Generalisation with Causal Invariant Bayesian Neural Networks

Gaël Gendron[1], Michael Witbrock[1], Gillian Dobbie[1]

[1]NAOInstitute, University of Auckland, New Zealand
Correspondence: gael.gendron@auckland.ac.nz

Abstract

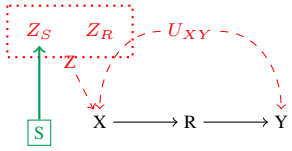
Deep neural networks can obtain impressive performance on various tasks under the assumption that their training domain is identical to their target domain. Performance can drop dramatically when this assumption does not hold. One explanation for this discrepancy is the presence of spurious domain-specific correlations in the training data that the network exploits. Causal mechanisms, in the other hand, can be made invariant under distribution changes using transportability theory as it allows disentangling the domain-specific and stable factors underlying the data generation. Yet, learning transportable causal mechanisms to improve out-of-distribution generalisation in deep neural networks remains an under-explored area. We propose a Bayesian neural architecture that disentangles the learning of the data distribution from the inference process mechanisms. We show theoretically and experimentally that our model approximates reasoning under causal interventions. We demonstrate the performance of our method, outperforming point estimate-counterparts, on out-of-distribution image recognition tasks where the data distribution acts as strong adversarial confounders.

1 Introduction

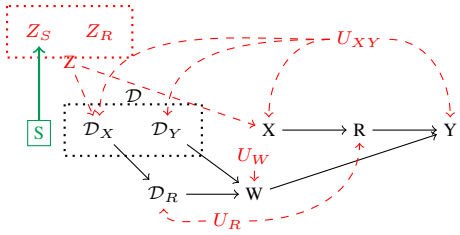
The training of deep neural networks commonly relies on the assumption that the distribution of the training data is representative of the distribution at inference. Despite being widely adopted, this assumption has been heavily criticised as it is often challenged in practice [18, 4, 1, 2, 14]. Indeed, despite tremendous progress on many vision tasks over recent years, deep neural networks face limitations and reduced performance on tasks requiring the model to shift from its training distribution at test time [24].

Causality theory under the do-calculus framework [12, 25] provides tools to explain these limitations. Neural networks extract correlation patterns but do not possess knowledge on the cause-effect relationships from the data; causal links and potentially spurious correlations are learned alike. However, the two relationships fundamentally differ as correlations can be non-causal: e.g. a correlation between X and Y can be explained by a causal link $X \leftarrow Z \rightarrow Y$. This type of correlation can be specific to the distribution if Y depends on the domain. Direct causal links make these relationships explicit and are therefore more robust to changes in the distribution [28]. In particular, the *Independent Causal Mechanisms* principle states that causal relationships are only sparsely connected and altering one should not modify the others [26, 28], allowing some learned mechanisms to be *transported* to new environments. Indeed, changes in the distribution can be modelled via the notion of *transportability* of causal mechanisms [3, 14]. As such, transportable causal effects are equivalent to domain-invariant features. These principles motivate the search for factorised neural models composed of independent modules representing either domain-specific or domain-invariant conditional distributions.

38th Conference on Neural Information Processing Systems (NeurIPS 2024).



(a) Target causal mechanism. R is a latent representation of X estimating Z_R. This representation allows computing interventional queries transportable across domains using R. However, it is not directly accessible via supervised learning.



(b) Causal graph during training. The inference mechanism determining Y is estimated using a function $f_w(r)$ parametrised by the weights $w \sim W$ and with an input representation $r \sim R$. W is learned from the training data $\mathcal{D}$ (training target $\mathcal{D}_Y$ and representation $\mathcal{D}_R$ learned from $\mathcal{D}_X$) and stochastic processes represented by the random variable U_W.

Figure 1: Target causal representation and actual causal graph during training. X is the input image and Y is the output class. Z is the variable representing the factors of distribution generating the input X, it is composed of domain-specific factors Z_S and robust domain-invariant factors Z_R. U_{XY} represents the shared factors of variations between X and Y. The variables in red are unobserved. S is a selection variable determining the current domain. The target causal graph is different from the one used in supervised deep learning as it omits the influence of the datasets $\mathcal{D} = \{\mathcal{D}_X, \mathcal{D}_Y\}$. The datasets depend on the same factors as X and Y and add spurious correlations (U_{XY} and U_R) between the training and target domains that are not captured when using the target representation.

Bayesian neural networks (BNNs) [21, 6, 22] are another line of work attempting to model robust representations of mechanisms by learning the distribution W of a parametric function $y = f_w(x), w \sim W$ instead of a single point estimate w. BNNs are particular suited to express uncertainty in their response and reduce overconfidence. By differentiating the modelling of uncertainty in the data distribution (aleatoric uncertainty) from the uncertainty in the process distribution (epistemic uncertainty), BNNs can better capture the latter [15]. However, BNNs are not fundamentally more robust against distribution shifts than their point-estimate counterparts [13]. This problem can be explained by the intractability in the optimisation of the posterior [22]. Motivated by the use of partially stochastic networks [29], we argue that BNNs should also be integrated in a wider framework estimating causal mechanisms.

We propose a Causal-Invariant Bayesian (CIB) neural network taking advantage of the causal structure of the supervised learning process to differentiate domain-specific and domain-invariant mechanisms. Compared to previous work, we integrate a more realistic causal graph that takes into account the Bayesian model update during learning. We use variational inference and partially-stochastic Bayesian neural networks to model the causal paths in an interventional setting. We verify theoretically and experimentally the applicability of our method. In particular, we perform tests on standard out-of-distribution (o.o.d) image recognition tasks. Our contributions are as follows:

- We reformulate the Bayesian inference problem to include causal interventions and the supervised learning mechanism and propose a factorised model explicitly modelling domain-invariant mechanisms in an unsupervised fashion,
- We propose an architecture wrapped around a neural network inspired by this factorised model and test it on challenging tasks requiring domain shifts,
- We show that adding our model can increase the i.i.d and o.o.d performance of the underlying neural network and reduce overfitting,
- Additionally, we find with our model that adding unsupervised contextual information with a Bayesian classifier to a standard ResNet improves its robustness and training stability, even with a small number of context and Bayesian weight samples.

Our code is available at this anonymous repository: `https://anonymous.4open.science/r/cibnn-1B0B`.

2 Background and Related Work

Causal Inference and Transportability Causal inference methods aim to estimate the result of causal queries. Answering these queries require an inference model to have a partial knowledge of the causal mechanisms underlying the studied system. The problem of formally representing causal quantities is addressed by the do-calculus framework, which introduces the $do(\cdot)$ operator corresponding to an intervention on a causal variable [25]. For example, given a treatment T and background covariates X, the probability of getting cured given by C can be expressed with the query $P(C|do(T), X)$. The do-operation implies that the treatment is received in an unbiased manner (i.e. as in a double-blind study). An interventional query as described in the example can be reduced to observations using the rules of do-calculus. We describe them in Appendix B.1. Recent work has attempted to model domain generalisation in classification tasks using causal representations [23, 20, 8, 30, 33]. However, they handle domain shifts in different ways. For instance Mahajan et al. [23] considers the domain as an observed causal variable while it in unobserved in Wang et al. [33].

Transportability theory [3] provides a unified representation of domain shifts in causal graphs and allows representing causal mechanisms across multiple domains by differentiating domain-specific and domain-invariant causal variables. Assuming the domains can be represented with a unified causal graph, domain-specific mechanisms can be modelled by a *selection variable* whose value depends on the current domain. The *S-admissibility* criterion [14] states that the value of a causal variable A is invariant when shifting from a domain $\mathcal{M}_i$ to a domain $\mathcal{M}_j$ if it can be *d-separated* from all selection variables S_{ij} (given observations Z), i.e. $A \perp\!\!\!\perp S_{ij}|Z \implies P^i(A|Z) = P^j(A|Z)$. Mao et al. [24] also integrate causal transportability for vision tasks but do not include the Bayesian learning process. They combine input and contextual information differently and only select context with the same label during training. Conversely, we make the assumption that the context should be diverse and representative of the label distribution. We use label mixup to this end. We also include new regularisation techniques.

Bayesian Neural Networks and Variational Inference Bayesian deep learning methods teach neural networks to simulate Bayesian reasoning when learning new information. This approach allows dealing with uncertainty and has been argued to help mitigate overconfidence and improve robustness in neural networks [22]. Bayesian Neural Networks (BNNs) model the distribution of parametric functions solving a task: the parameters w of a BNN are not directly optimised but sampled from a learned posterior distribution $P(w|\mathcal{D})$. This distribution can be obtained via the *Bayes objective*: $P(w|\mathcal{D}) = \frac{P(\mathcal{D}|w)P(w)}{P(\mathcal{D})}$. However, training BNNs is challenging because the computation of the denominator (called the *evidence*) is often intractable [13]. A standard way to circumvent this issue is to approximate the posterior $P(w|\mathcal{D})$ with a *variational distribution* $q(w)$. This distribution can be estimated by maximising the *Evidence Lower Bound* (ELBO) [6]. This quantity can have many expressions, the one most suitable for our work is shown in Equation 1. It simultaneously maximises the likelihood of the data $\mathcal{D} = \{\mathcal{D}_x, \mathcal{D}_y\}$ given the model w while maintaining the distribution $q(w)$ close to the prior $P(w)$.

$$\text{ELBO}(q) = \mathbb{E}_{q(w)}[\log P(\mathcal{D}_y|\mathcal{D}_x, w)] - \text{KL}(q(w)||P(w)) \tag{1}$$

Izmailov et al. [13] showed that the posterior distribution of BNNs have a high functional diversity and can outperform their *point-estimate* counterparts on downstream tasks. However, they do not necessarily show a high diversity in the parameters space and strictly following Bayesian posteriors can lead to reduced robustness against distribution shifts. Sharma et al. [29] further showed that partially-stochastic BNNs outperform networks containing only Bayesian layers. Roy et al. [27] recently showed that inducing BNNs to assign similar posterior densities to reparametrisations of the same functions (i.e. different weights realising the same function) could improve their ability to fit the data. These findings motivate our work on integrating BNNs as a part of a larger causal neural network.

3 Interventional Bayesian Inference

We propose to learn deep domain-invariant representations that can be leveraged to solve o.o.d tasks. In this section, we show the necessary assumptions needed to learn this representation in a supervised

fashion and identify an interventional query that can solve the task. We show that this query can be answered using Bayesian Neural Networks [21, 6, 22]. Section 4 describes our proposed architecture.

Interventional queries The causal graph in Figure 1b shows the causal dependencies between the variables involved in the learning process. The standard inference query $P(Y|x)$ is affected by spurious correlations as they do not distinguish the causal links $X \overset{\cdots}{\longrightarrow} Y$ (i.e. $X \rightarrow \cdots \rightarrow Y$) and their common causes $Z \overset{\cdots}{\longleftarrow} U_{XY} \overset{\cdots}{\longrightarrow} Y$. In causality theory, the $do(\cdot)$ operator removes the incoming dependencies of the target variable [25]. Therefore, the interventional query $P(Y|do(x))$ solves the confounding issue [24]. However, it does not take into account the conditioning on the training data $\mathcal{D}$. We specify an interventional query that makes the dependency on the data explicit. Note that we simplify the formalism by considering the label Y_{true} and predicted value Y_{pred} with a single variable Y. This is equivalent to having a prediction loss $\mathcal{L}(Y_{\text{pred}}, Y_{\text{true}}) = 0$. Conditional distributions independent from the domain selection variable S are transportable across domains [14]. This independence can be obtained using the $do(\cdot)$ operation by removing the outgoing dependencies of S. The conditional distribution $P(y|do(x), do(\mathcal{D}_X), \mathcal{D}_Y)$ is transportable across domains but is not tractable as it requires marginalising over all possible input distributions (via the frontdoor criterion). We give more details in Appendix B.3. Instead, we use a partially transportable query $P(y|do(x), \mathcal{D}_X, \mathcal{D}_Y)$. Equation 2 formulates it using observations only (proof in Appendix B.2):

$$
\begin{aligned}
&P(y|do(x), \mathcal{D}_X, \mathcal{D}_Y) \\
&= \underbrace{\int_w P(w|\mathcal{D}_X, \mathcal{D}_Y)}_{\text{marginalisation over W}} \underbrace{\int_r P(r|x, w, \mathcal{D}_X, \mathcal{D}_Y)}_{\text{marginalisation over R}} \underbrace{\int_{x'} P(x'|\mathcal{D}_X, \mathcal{D}_Y)}_{\text{marginalisation over X}} \overbrace{P(y|x', r, w, \mathcal{D}_X, \mathcal{D}_Y)}^{\text{inference}} \, dx' \, dr \, dw
\end{aligned}
\tag{2}
$$

The interventional query must be decomposed into four components to be represented using observations only. The marginalisation over W is equivalent to an application of the backdoor criterion and ensures that no backdoor paths exist between Y and W. The marginalisations over X and R are applications of the frontdoor criterion. The combination of these mechanisms ensure that no backdoor paths exist between the input X and the output Y that could bias the training.

Transportability We have identified an unbiased interventional query from observational information only. We now study the transportability of each component across domains using the $\mathcal{S}$-admissibility criterion [14]: conditional probabilities independent from the domain selection variable S are transportable. We assume a training source domain $\mathcal{M}^s$ and a target test domain $\mathcal{M}^t$, it follows graphically that:

$$(W \perp\!\!\!\perp S|\mathcal{D}_X, \mathcal{D}_Y)_{\mathcal{G}^{\Delta_{st}}} \tag{3}$$

$$(R \perp\!\!\!\perp S|X, W, \mathcal{D}_X, \mathcal{D}_Y)_{\mathcal{G}^{\Delta_{st}}} \tag{4}$$

$$(X \not\perp\!\!\!\perp S|\mathcal{D}_X, \mathcal{D}_Y)_{\mathcal{G}^{\Delta_{st}}} \tag{5}$$

$$(Y \not\perp\!\!\!\perp S|X, R, W, \mathcal{D}_X, \mathcal{D}_Y)_{\mathcal{G}^{\Delta_{st}}} \tag{6}$$

The first two quantities are $\mathcal{S}$-admissible and can be written as $P^s(w|\mathcal{D}_{X_s}, \mathcal{D}_{Y_s}) = P^t(w|\mathcal{D}_{X_t}, \mathcal{D}_{Y_t}) = P^*(w|\mathcal{D}_X, \mathcal{D}_Y)$ and $P^s(r|x, w, \mathcal{D}_{X_s}, \mathcal{D}_{Y_s}) = P^t(r|x, w, \mathcal{D}_{X_t}, \mathcal{D}_{Y_t}) = P^*(r|x, w, \mathcal{D}_X, \mathcal{D}_Y)$. The last two quantities are not $\mathcal{S}$-admissible and require to have access to information $\mathcal{D}_{X_t}, \mathcal{D}_{Y_t}$ on the target domain. This is an expected result for $P(x'|\mathcal{D}_X, \mathcal{D}_Y)$. Indeed, the input X directly depends on the distribution of its factors of variations. We must learn a suitable representation of X to use it for inference. Note that this probability is independent of Y and does not require access to labelled data. Under the current SCM problem formulation, $P(y|x', r, w, \mathcal{D}_X, \mathcal{D}_Y)$ is not $\mathcal{S}$-admissible because of the backdoor path between the training data $\mathcal{D}$ and the inference target Y. However, in out-of-distribution settings, the training and target domains can differ and the backdoor path may not exist. This is a reasonable expectation as the converse of the $\mathcal{S}$-admissibility criterion does not hold: non $\mathcal{S}$-admissibility does not implies non-transportability.

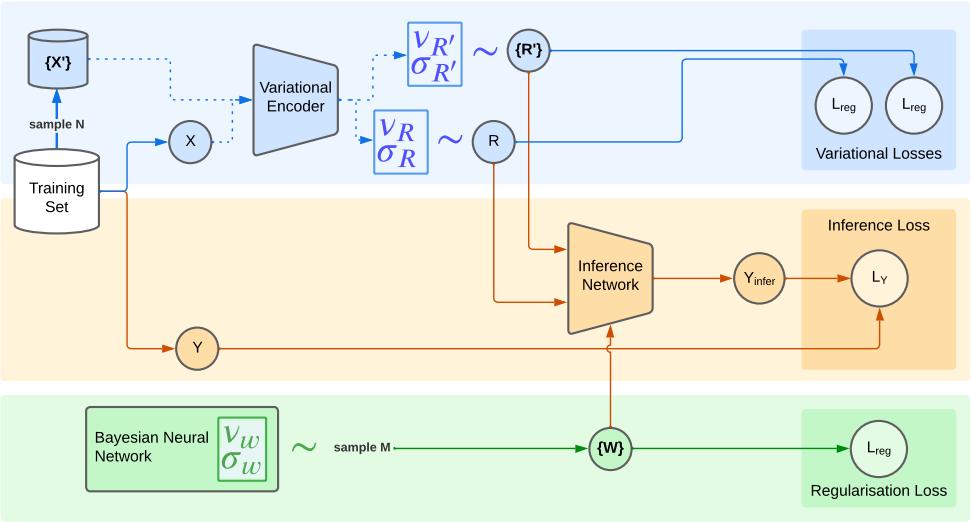


Figure 2: Architecture of the Causal-Invariant Bayesian (CIB) neural network. At the top (in blue), a variational encoder generates the respective parameters of the distributions of the intermediate representations R and $\{R'_i\}_{i=1}^N$ of X and the contextual information $\{X'_i\}_{i=1}^N$. R and $\{R'_i\}_{i=1}^N$ are then provided to the inference module (in orange) that retrieves Y. This procedure aims to disentangle the learning of the representation R from the learning of the inference mechanisms and force the inference module to only learn the latter. The weights W of the inference module are sampled from a distribution learned using variational inference (in green). The weight sampling and the variational encoding are regularised using an ELBO loss.

4 Causal-Invariant Bayesian Neural Network

We now propose our Causal-Invariant Bayesian (CIB) architecture estimating the posterior probability defined above using sampling strategies for each quantity. The architecture is summarised in Figure 2. It can be decomposed into three components:

- A variational encoder (in blue) learning domain-invariant information R by jointly optimising the inference loss with an ELBO loss,
- An inference module (in orange) combining domain-specific and domain-invariant information to solve the task,
- A weight distribution W (in green) learned by optimising the inference objective with an ELBO loss and from which are sampled the parameters of the inference module.

The variational encoder is used to learn the intermediate domain-invariant representations R and $\{R'_i\}_{i=1}^N$. The network weights are learned by optimising an ELBO loss. Existing work attempting to disentangle domain-specific and domain-invariant features typically use a Variational Autoencoder (VAE) and add a reconstruction loss [10, 34, 35]. However, we find that a single encoder can yield better performance. By jointly giving the input and the context to the encoder, we teach it to discard irrelevant domain-specific features at once. We use a pre-trained ResNet-18 [11] without the final classification layer as the encoder. For comparison, we perform additional experiments with the Causal Transportability architecture that integrates a VAE and a reconstruction loss [24]. Following the methodology of partially-stochastic neural networks [29], we use a point-estimate network for the encoder and only use a Bayesian network for the inference network. This choice allows to maintain the expressivity on the level of uncertainty inherent to Bayesian neural networks while having a limited impact on the optimisation efficiency. For the inference model, we use a dense network with stochastic layers for the input and output and a single point-estimate hidden layer. The inference module takes batches of N inputs. Each input is a sum of the input representation R and one context

Table 1: Accuracy on CIFAR-10. The CIFAR-10 column shows the results on the i.i.d test set while the columns on the right show the results with an increasing level of perturbation of the test set. The mean and standard deviation across three runs are shown. Our model outperforms the baselines and the ablated models. Particularly, mixup information has a significant impact on the final accuracy.

	CIFAR-10	CIFAR-10-o.o.d-perturb.		
		0.1	0.2	0.4
ResNet-18	0.739 ± 0.005	0.599 ± 0.017	0.471 ± 0.022	0.284 ± 0.010
ResNet-18-CT	0.682 ± 0.004	0.507 ± 0.010	0.397 ± 0.006	0.244 ± 0.010
-w/-pretrainedVAE	0.641 ± 0.004	0.464 ± 0.017	0.358 ± 0.015	0.229 ± 0.010
CIBResNet-18 (ours)	$\mathbf{0.763 \pm 0.004}$	$\mathbf{0.646 \pm 0.006}$	$\mathbf{0.510 \pm 0.012}$	$\mathit{0.301 \pm 0.006}$
-w/o-mixup	0.761 ± 0.004	0.625 ± 0.013	0.494 ± 0.009	0.297 ± 0.004
-w/o-weight-func	$\mathit{0.762 \pm 0.009}$	$\mathit{0.635 \pm 0.010}$	$\mathit{0.505 \pm 0.009}$	$\mathbf{0.305 \pm 0.007}$

representation R'_i. We select the mean of the results with respect to N to estimate the marginalisation over the context.

We use a reparametrisation trick [5] to maintain the full differentiability of the network. Following Elfwing et al. [9], we change the activation layers from ReLU to SiLU. We add a label-mixing regularisation scheme to complement the contextual information during training, following the procedure used in Mixup [36] and Manifold Mixup [32]. We also regularise the weights sampling of the inference module to enhance the diversity of the weights while aligning it with functional diversity Roy et al. [27]. The loss function comprises a cross entropy prediction loss and the additional regularisation terms from the ELBO losses and the mechanisms described above. Additional details are provided in Appendix A.

5 Experiments

5.1 Experimental Setup

We perform experiments on i.i.d and o.o.d image recognition tasks. We first use the CIFAR10 dataset [17]. We build the o.o.d sets by performing random translations of the input images. We also use the OFFICEHOME dataset [31] which contains four subsets of images containing objects in various configurations (real world, product sheet, art, clipart). We train on one configuration and evaluate o.o.d in another configuration. We compare our method against a base ResNet-18 and the causal-Transportability (CT) algorithm of Mao et al. [24]. More details about this model and the experimental setup are given in Appendices C and D.

5.2 Results

We compare our model against ResNet-18 and ResNet-18-CT on CIFAR10. The results are shown in Table 1. Curiously, the base ResNet-18 outperforms the ResNet-18-CT models. This result reinforces our hypothesis that a reconstruction loss is not necessary and can even be adversarial in some situations. We further study how our model behaves on a more challenging dataset and compare it with ResNet-18. Figure 3 shows the results on OFFICEHOME. Our model systematically outperforms the baselines. Ablation studies in Table 1 demonstrate the importance of adding contextual mixup information to the label during training to help the model incorporate the context images during learning. We perform additional experiments in Appendix E that show that our model yields higher stability during training and that increasing the amount of weight sampling improves the data efficicency of our method.

6 Conclusion

We propose a Causal-Invariant Bayesian neural network architecture to improve domain generalisation on visual recognition tasks. We build a theoretical analysis based on causality theory and taking into account the parameter learning process. We show experimentally that following causal principles can improve robustness to distribution shifts and reduce overfitting. We perform our analysis under the

	RealWorld	Product	Art	Clipart
RealWorld	0.253 (±0.019)	0.124 (±0.01)	0.091 (±0.016)	0.084 (±0.006)
Product	0.034 (±0.013)	0.509 (±0.061)	0.023 (±0.005)	0.023 (±0.009)
Art	0.06 (±0.001)	0.019 (±0.003)	0.125 (±0.017)	0.035 (±0.001)
Clipart	0.064 (±0.01)	0.03 (±0.007)	0.038 (±0.006)	0.462 (±0.025)

(a) ResNet-18.

	RealWorld	Product	Art	Clipart
RealWorld	0.303 (±0.009)	0.191 (±0.016)	0.121 (±0.009)	0.123 (±0.004)
Product	0.052 (±0.003)	0.574 (±0.015)	0.036 (±0.002)	0.042 (±0.007)
Art	0.076 (±0.019)	0.033 (±0.015)	0.159 (±0.023)	0.034 (±0.009)
Clipart	0.083 (±0.009)	0.055 (±0.01)	0.046 (±0.012)	0.517 (±0.009)

(b) CIBResNet-18.

Figure 3: Domain transfer results on the OfficeHome dataset. Rows show the training category and columns show the test category. Accuracy with a random guess is 0.015. In the right figure, a cell is shown in green if its value is higher than the baseline. We show the mean and standard deviation across three runs. Our proposed model (right) systematically outperforms the baseline (left).

SCM framework [25]. The underlying causal model is represented by a DAG that does not allow feedback loops. As the training of a neural network is an iterative process, a representation that allows cycles could better represent the inner causal mechanisms behind the learning process and help improve generalisation. This is a challenging task [7] that we will tackle in our future work. This investigation also provides elements to motivate developing a causal theory of deep learning that follows the Independent Causal Mechanisms principle [26, 28] as it could improve robustness and generalisation and provide additional benefits such as improved interpretability and modularity.

References

[1] Isabela Albuquerque, João Monteiro, Mohammad Darvishi, Tiago H Falk, and Ioannis Mitliagkas. Generalizing to unseen domains via distribution matching. *arXiv preprint arXiv:1911.00804*, 2019.

[2] Martín Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz. Invariant risk minimization. *CoRR*, abs/1907.02893, 2019. URL `http://arxiv.org/abs/1907.02893`.

[3] Elias Bareinboim and Judea Pearl. A general algorithm for deciding transportability of experimental results. *CoRR*, abs/1312.7485, 2013. URL `http://arxiv.org/abs/1312.7485`.

[4] Shai Ben-David, John Blitzer, Koby Crammer, and Fernando Pereira. Analysis of representations for domain adaptation. In Bernhard Schölkopf, John C. Platt, and Thomas Hofmann (eds.), *Advances in Neural Information Processing Systems 19, Proceedings of the Twentieth Annual Conference on Neural Information Processing Systems, Vancouver, British Columbia, Canada, December 4-7, 2006*, pp. 137–144. MIT Press, 2006. URL `https://proceedings.neurips.cc/paper/2006/hash/b1b0432ceafb0ce714426e9114852ac7-Abstract.html`.

[5] Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. Weight uncertainty in neural networks. *CoRR*, abs/1505.05424, 2015. URL `http://arxiv.org/abs/1505.05424`.

[6] Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. Weight uncertainty in neural network. In *International conference on machine learning*, pp. 1613–1622. PMLR, 2015.

[7] Stephan Bongers, Patrick Forré, Jonas Peters, and Joris M. Mooij. Foundations of structural causal models with cycles and latent variables. *The Annals of Statistics*, 49(5):2885 – 2915, 2021. doi: 10.1214/21-AOS2064. URL `https://doi.org/10.1214/21-AOS2064`.

[8] Jin Chen, Zhi Gao, Xinxiao Wu, and Jiebo Luo. Meta-causal learning for single domain generalization. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2023, Vancouver, BC, Canada, June 17-24, 2023*, pp. 7683–7692. IEEE, 2023. doi: 10.1109/CVPR52729.2023.00742. URL `https://doi.org/10.1109/CVPR52729.2023.00742`.

[9] Stefan Elfwing, Eiji Uchibe, and Kenji Doya. Sigmoid-weighted linear units for neural network function approximation in reinforcement learning. *Neural Networks*, 107:3–11, 2018. doi: 10.1016/J.NEUNET.2017.12.012. URL https://doi.org/10.1016/j.neunet.2017.12.012.

[10] Aviv Gabbay, Niv Cohen, and Yedid Hoshen. An image is worth more than a thousand words: Towards disentanglement in the wild. In Marc'Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pp. 9216–9228, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/4cef5b5e6ff1b3445db4c013f1d452e0-Abstract.html.

[11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pp. 770–778. IEEE Computer Society, 2016. doi: 10.1109/CVPR.2016.90. URL https://doi.org/10.1109/CVPR.2016.90.

[12] Yimin Huang and Marco Valtorta. Pearl's calculus of intervention is complete. In *UAI '06, Proceedings of the 22nd Conference in Uncertainty in Artificial Intelligence, Cambridge, MA, USA, July 13-16, 2006*. AUAI Press, 2006. URL https://dslpitt.org/uai/displayArticleDetails.jsp?mmnu=1&smnu=2&article_id=1301&proceeding_id=22.

[13] Pavel Izmailov, Sharad Vikram, Matthew D. Hoffman, and Andrew Gordon Wilson. What are bayesian neural network posteriors really like? In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pp. 4629–4640. PMLR, 2021. URL http://proceedings.mlr.press/v139/izmailov21a.html.

[14] Kasra Jalaldoust and Elias Bareinboim. Transportable representations for domain generalization. In Michael J. Wooldridge, Jennifer G. Dy, and Sriraam Natarajan (eds.), *Thirty-Eighth AAAI Conference on Artificial Intelligence, AAAI 2024, Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence, IAAI 2024, Fourteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2014, February 20-27, 2024, Vancouver, Canada*, pp. 12790–12800. AAAI Press, 2024. doi: 10.1609/AAAI.V38I11.29175. URL https://doi.org/10.1609/aaai.v38i11.29175.

[15] Alex Kendall and Yarin Gal. What uncertainties do we need in bayesian deep learning for computer vision? In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett (eds.), *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pp. 5574–5584, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/2650d6089a6d640c5e85b2b88265dc2b-Abstract.html.

[16] Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. In Yoshua Bengio and Yann LeCun (eds.), *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014. URL http://arxiv.org/abs/1312.6114.

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

[18] John Langford. Tutorial on practical prediction theory for classification. *J. Mach. Learn. Res.*, 6:273–306, 2005. URL http://jmlr.org/papers/v6/langford05a.html.

[19] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. URL https://openreview.net/forum?id=Bkg6RiCqY7.

[20] Fangrui Lv, Jian Liang, Shuang Li, Bin Zang, Chi Harold Liu, Ziteng Wang, and Di Liu. Causality inspired representation learning for domain generalization. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, pp. 8036–8046. IEEE, 2022. doi: 10.1109/CVPR52688.2022.00788. URL https://doi.org/10.1109/CVPR52688.2022.00788.

[21] David JC MacKay. A practical bayesian framework for backpropagation networks. *Neural computation*, 4(3):448–472, 1992.

[22] Martin Magris and Alexandros Iosifidis. Bayesian learning for neural networks: an algorithmic survey. *Artif. Intell. Rev.*, 56(10):11773–11823, 2023. doi: 10.1007/S10462-023-10443-1. URL https://doi.org/10.1007/s10462-023-10443-1.

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

[24] Chengzhi Mao, Kevin Xia, James Wang, Hao Wang, Junfeng Yang, Elias Bareinboim, and Carl Vondrick. Causal transportability for visual recognition. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, pp. 7511–7521. IEEE, 2022. doi: 10.1109/CVPR52688.2022.00737. URL https://doi.org/10.1109/CVPR52688.2022.00737.

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

[26] Jonas Peters, Dominik Janzing, and Bernhard Schölkopf. *Elements of causal inference: foundations and learning algorithms*. The MIT Press, 2017.

[27] Hrittik Roy, Marco Miani, Carl Henrik Ek, Philipp Hennig, Marvin Pförtner, Lukas Tatzel, and Søren Hauberg. Reparameterization invariance in approximate bayesian inference. *CoRR*, abs/2406.03334, 2024. doi: 10.48550/ARXIV.2406.03334. URL https://doi.org/10.48550/arXiv.2406.03334.

[28] Bernhard Schölkopf, Francesco Locatello, Stefan Bauer, Nan Rosemary Ke, Nal Kalchbrenner, Anirudh Goyal, and Yoshua Bengio. Toward causal representation learning. *Proc. IEEE*, 109 (5):612–634, 2021. doi: 10.1109/JPROC.2021.3058954. URL https://doi.org/10.1109/JPROC.2021.3058954.

[29] Mrinank Sharma, Sebastian Farquhar, Eric T. Nalisnick, and Tom Rainforth. Do bayesian neural networks need to be fully stochastic? In Francisco J. R. Ruiz, Jennifer G. Dy, and Jan-Willem van de Meent (eds.), *International Conference on Artificial Intelligence and Statistics, 25-27 April 2023, Palau de Congressos, Valencia, Spain*, volume 206 of *Proceedings of Machine Learning Research*, pp. 7694–7722. PMLR, 2023. URL https://proceedings.mlr.press/v206/sharma23a.html.

[30] Paras Sheth, Raha Moraffah, K. Selçuk Candan, Adrienne Raglin, and Huan Liu. Domain generalization - A causal perspective. *CoRR*, abs/2209.15177, 2022. doi: 10.48550/ARXIV.2209.15177. URL https://doi.org/10.48550/arXiv.2209.15177.

[31] Hemanth Venkateswara, Jose Eusebio, Shayok Chakraborty, and Sethuraman Panchanathan. Deep hashing network for unsupervised domain adaptation. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pp. 5385–5394. IEEE Computer Society, 2017. doi: 10.1109/CVPR.2017.572. URL https://doi.org/10.1109/CVPR.2017.572.

[32] Vikas Verma, Alex Lamb, Christopher Beckham, Amir Najafi, Ioannis Mitliagkas, David Lopez-Paz, and Yoshua Bengio. Manifold mixup: Better representations by interpolating hidden states. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pp. 6438–6447. PMLR, 2019. URL http://proceedings.mlr.press/v97/verma19a.html.

[33] Xinyi Wang, Michael Saxon, Jiachen Li, Hongyang Zhang, Kun Zhang, and William Yang Wang. Causal balancing for domain generalization. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023. URL https://openreview.net/forum?id=F91SROvVJ_6.

[34] Mengyue Yang, Furui Liu, Zhitang Chen, Xinwei Shen, Jianye Hao, and Jun Wang. Causalvae: Disentangled representation learning via neural structural causal models. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*, pp. 9593–9602. Computer Vision Foundation / IEEE, 2021. doi: 10.1109/CVPR46437.2021.00947. URL https://openaccess.thecvf.com/content/CVPR2021/html/Yang_CausalVAE_Disentangled_Representation_Learning_via_Neural_Structural_Causal_Models_CVPR_2021_paper.html.

[35] Hanlin Zhang, Yi-Fan Zhang, Weiyang Liu, Adrian Weller, Bernhard Schölkopf, and Eric P. Xing. Towards principled disentanglement for domain generalization. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, pp. 8014–8024. IEEE, 2022. doi: 10.1109/CVPR52688.2022.00786. URL https://doi.org/10.1109/CVPR52688.2022.00786.

[36] Hongyi Zhang, Moustapha Cissé, Yann N. Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018. URL https://openreview.net/forum?id=r1Ddp1-Rb.

A Additional Details on the Causal-Invariant Bayesian Neural Network

This section describes additional implementation details of our proposed model.

Tractable Approximation The quantities desribed in equation 2 are not directly tractable and must be estimated. The marginalisation over W is a standard technique from Bayesian inference [22]. The integral can be approximated using Markov Chain Monte Carlo (MCMC) or Variational Inference [6, 22]. Specifically:

$$
\begin{aligned}
&P(y|do(x), \mathcal{D}_X, \mathcal{D}_Y) \\
&= \int_w P(w|\mathcal{D}_X, \mathcal{D}_Y) \int_r P(r|x, w, \mathcal{D}_X, \mathcal{D}_Y) \int_{x'} P(x'|\mathcal{D}_X, \mathcal{D}_Y) P(y|x', r, w, \mathcal{D}_X, \mathcal{D}_Y)\, dx'\, dr\, dw \\
&\approx \frac{1}{M} \sum_{j=1}^{M} \int_r P(r|x, w_j, \mathcal{D}_X, \mathcal{D}_Y) \int_{x'} P(x'|\mathcal{D}_X, \mathcal{D}_Y) P(y|x', r, w_j, \mathcal{D}_X, \mathcal{D}_Y)\, dx'\, dr
\end{aligned}
$$

where $w_j \sim P(w|\mathcal{D}_X, \mathcal{D}_Y)$ and M is the amount of weight samples. The distribution of the weights W can be learned by a Bayesian Neural Network (BNN) [21, 6, 22]. The network does not directly learn the weights w but instead estimate the parameters of the distribution in which the weights belong and sample the weights from this distribution at inference. The learning objective cannot be directly optimised because of the high-dimensional and non-convex posterior distribution $P(w|\mathcal{D})$. Instead, MCMC [13] and Variational Inference [6] algorithms are preferred. The marginalisation over R is approximated similarly:

$$
\begin{aligned}
&\int_r P(r|x, w_j, \mathcal{D}_X, \mathcal{D}_Y) \int_{x'} P(x'|\mathcal{D}_X, \mathcal{D}_Y) P(y|x', r, w_j, \mathcal{D}_X, \mathcal{D}_Y)\, dx'\, dr \\
&\approx \frac{1}{L} \sum_{k=1}^{L} \int_{x'} P(x'|\mathcal{D}_X, \mathcal{D}_Y) P(y|x', r_k, w_j, \mathcal{D}_X, \mathcal{D}_Y)\, dx'
\end{aligned}
$$

where $r_k \sim P(r|x, w_j, \mathcal{D}_X, \mathcal{D}_Y)$ and L is the amount of latent representation samples. Variational Autoencoders (VAEs) [16] are commonly used to estimate this distribution. In practice, we consider a single representation R per input and context image, i.e. $L = 1$. Finally, the last marginalisation can be obtained by sampling input images from the current data distribution:

$$
\int_{x'} P(x'|\mathcal{D}_X, \mathcal{D}_Y) P(y|x', r, w_j, \mathcal{D}_X, \mathcal{D}_Y)\, dx' \approx \frac{1}{N} \sum_{k=1}^{N} P(x_i|\mathcal{D}_X, \mathcal{D}_Y) P(y|x_i, r_k, w_j, \mathcal{D}_X, \mathcal{D}_Y)
$$

where N is the amount of input image samples.

Context Regularisation We add a label-mixing regularisation scheme to complement the contextual information during training, following the procedure used in Mixup [36] and Manifold Mixup [32]. We mix the true instance label $\mathbf{y_t}$ with the labels $\mathbf{Y_{x_i}}$ of the N context images x_i, as described in Equation 7.

$$
\mathbf{y} = \alpha \cdot \mathbf{y_t} + (1 - \alpha) \cdot \frac{1}{N} \sum_{i=1}^{N} \mathbf{y_{x_i}} \tag{7}
$$

Weight Function Regularisation We further regularise the weights sampling of the inference module to enhance the diversity of the weights while aligning it with functional diversity. Following the work of Roy et al. [27], we add a regularisation loss inducing the network to assign the same posterior distribution to weights realising an identical function. For a batch of inputs $\mathbf{x}$ and a set of weight samples $\mathcal{W} = \{w_i\}_{i=1}^{M}$, we define our *weight function regularisation* loss in Equation 8. $\text{Comb}(\mathcal{W}, 2)$ denotes the set of 2-combinations from $\mathcal{W}$.

$$\mathcal{L}_{weight_func} = \sum_{(w_i,w_j)\in \text{Comb}(\mathcal{W},2)} ||f_{w_i}(\mathbf{x}) - f_{w_j}(\mathbf{x})||^2 \tag{8}$$

Loss function The final loss function comprises the mixed labels loss, the weight function regularisation and the regularisation terms of the ELBO losses. The mixed label loss is a cross entropy loss between the predicted distribution and the true mixed distribution as described above. We add the weight function regularisation loss described in the previous paragraph, weighted by a hyperparameter β. Then, KL divergence losses are added for regularising the variational parameters for the input representation (ν_R, σ_R), the context representation $(\nu_{R'}, \sigma_{R'})$ and the Bayesian weights (ν_W, σ_W). These additional losses are weighted by their respective hyperparameters γ, μ and ϵ. The complete loss is shown in Equation 9.

$$\begin{aligned}\mathcal{L} = &\text{CrossEntropy}(\mathbf{y_{pred}}, \mathbf{y}) + \beta \cdot \mathcal{L}_{weight_func} + \gamma \cdot \text{KL}(\mathcal{N}(\nu_R, \sigma_R)||\mathcal{N}(0,1)) \\ &+ \mu \cdot \text{KL}(\mathcal{N}(\nu_{R'}, \sigma_{R'})||\mathcal{N}(0,1)) + \epsilon \cdot \text{KL}(\mathcal{N}(\nu_W, \sigma_W)||\mathcal{N}(0,1))\end{aligned} \tag{9}$$

B Proofs for the Interventional Bayesian Inference

B.1 Rules of do-calculus

We first describe the three rules of do-calculus [25] as they are required for the proofs in Sections B.2 and B.3. The first rule states that an observation z can be ignored if it does not affect the outcome y of the query given the current causal graph. The second rule states that an intervention on a variable $do(z)$ can be considered an observation z if there are no backdoor paths linking it to the outcome y, i.e. if the variables Z and Y do not share any common ancestors not captured by the observation z. The third rule states that an intervention $do(z)$ can be ignored if there are no direct paths between Z and Y or backdoor paths between Y and observed descendants W of Z.

The three rules are formally defined as follows. For each rule, the equality holds if the independence test between the variable of interest Z and the outcome Y is verified in the given causal graph. The graph $\mathcal{G}_{\overline{X}}$ corresponds to the initial graph with the incoming links to X removed, corresponding to an intervention on X. The graph $\mathcal{G}_{\overline{X}}$ furthermore removes the outgoing links of Z. The term $\overline{Z(W)}$ removes the incmoing links of Z if Z is not an ancestor of the observed variable W.

- **Rule 1** Deletion of observation

$$P(y|do(x), z, w) = P(y|do(x), w) \text{ if } (Y \perp\!\!\!\perp Z|X, W)_{\mathcal{G}_{\overline{X}}} \tag{10}$$

- **Rule 2** Reduction of intervention to observation

$$P(y|do(x), do(z), w) = P(y|do(x), z, w) \text{ if } (Y \perp\!\!\!\perp Z|X, W)_{\mathcal{G}_{\overline{X}\underline{Z}}} \tag{11}$$

- **Rule 3** Deletion of intervention

$$P(y|do(x), do(z), w) = P(y|do(x), w) \text{ if } (Y \perp\!\!\!\perp Z|X, W)_{\mathcal{G}_{\overline{X},\overline{Z(W)}}} \tag{12}$$

B.2 Identifiability of $P(y|do(x), \mathcal{D}_X, \mathcal{D}_Y)$

Proof. Proof of Equation 2. We start by marginalising on W to block the backdoor path between R and Y ($R \leftarrow U_R \rightarrow \mathcal{D}_R \rightarrow W \rightarrow Y$) and use the rules of do-calculus to simplify the quantities. This step ensures that we can later apply the frontdoor criterion with R and block the backdoor paths between X and Y.

$$\begin{aligned}
&P(y|do(x), \mathcal{D}_X, \mathcal{D}_Y) \\
&= \int_w P(y|do(x), \mathcal{D}_X, \mathcal{D}_Y, w) P(w|do(x), \mathcal{D}_X, \mathcal{D}_Y)\, dw && \textit{Marginalisation over W} \\
&= \int_w P(y|do(x), \mathcal{D}_X, \mathcal{D}_Y, w) P(w|\mathcal{D}_X, \mathcal{D}_Y)\, dw && \textit{Rule 3: } (W \perp\!\!\!\perp X|\mathcal{D}_X, \mathcal{D}_Y)_{\mathcal{G}_{\overline{X}}}
\end{aligned}$$

We now focus on the quantity in the red box. We must use the frontdoor criterion to block all backdoor paths between X and Y: $X \leftarrow U_{XY} \rightarrow Y$, $X \leftarrow Z \rightarrow \mathcal{D}_X \leftarrow U_{XY} \rightarrow Y$. We marginalise on R.

$$
\begin{aligned}
& \boxed{P(y|do(x), \mathcal{D}_X, \mathcal{D}_Y, w)} \\
&= \int_r P(y|do(x), \mathcal{D}_X, \mathcal{D}_Y, w, r) P(r|do(x), \mathcal{D}_X, \mathcal{D}_Y, w)\, dr && \textit{Marginalisation over R} \\
&= \int_r \boxed{P(y|do(x), \mathcal{D}_X, \mathcal{D}_Y, w, r)} P(r|x, \mathcal{D}_X \mathcal{D}_Y, w)\, dr && \textit{Rule 2: } (R \perp\!\!\!\perp X | \mathcal{D}_X, \mathcal{D}_Y, W)_{\mathcal{G}_{\underline{X}}}
\end{aligned}
$$

Again, we focus on the quantity in the blue box. We continue to apply the frontdoor criterion and marginalise on X after switching the intervened variable using do-calculus rules.

$$
\begin{aligned}
& \boxed{P(y|do(x), \mathcal{D}_X, \mathcal{D}_Y, w, r)} \\
&= P(y|do(x), do(r), \mathcal{D}_X, \mathcal{D}_Y, w) && \textit{Rule 2: } (Y \perp\!\!\!\perp R | X, \mathcal{D}_X, \mathcal{D}_Y, W)_{\mathcal{G}_{\overline{X}\underline{R}}} \\
&= P(y|do(r), \mathcal{D}_X, \mathcal{D}_Y, w) && \textit{Rule 3: } (Y \perp\!\!\!\perp X | R, \mathcal{D}_X, \mathcal{D}_Y, W)_{\mathcal{G}_{\overline{XR}}} \\
&= \int_{x'} P(y|do(r), \mathcal{D}_X, \mathcal{D}_Y, w, x') P(x'|do(r), \mathcal{D}_X, \mathcal{D}_Y, w)\, dx' && \textit{Marginalisation over X} \\
&= \int_{x'} P(y|r, \mathcal{D}_X, \mathcal{D}_Y, w, x') P(x'|do(r), \mathcal{D}_X, \mathcal{D}_Y, w)\, dx' && \textit{Rule 2: } (Y \perp\!\!\!\perp R | X, \mathcal{D}_X, \mathcal{D}_Y, W)_{\mathcal{G}_{\underline{R}}} \\
&= \int_{x'} P(y|r, \mathcal{D}_X, \mathcal{D}_Y, w, x') P(x'|\mathcal{D}_X, \mathcal{D}_Y, w)\, dx' && \textit{Rule 3: } (X \perp\!\!\!\perp R | \mathcal{D}_X, \mathcal{D}_Y, W)_{\mathcal{G}_{\overline{R}}} \\
&= \int_{x'} P(y|r, \mathcal{D}_X, \mathcal{D}_Y, w, x') P(x'|\mathcal{D}_X, \mathcal{D}_Y)\, dx' && \textit{Rule 1: } (X \perp\!\!\!\perp W | \mathcal{D}_X, \mathcal{D}_Y)_{\mathcal{G}}
\end{aligned}
$$

We put together all the quantities and obtain Equation 2:

$$
\begin{aligned}
& P(y|do(x), \mathcal{D}_X, \mathcal{D}_Y) \\
&= \int_w P(w|\mathcal{D}_X, \mathcal{D}_Y) \int_r P(r|x, w, \mathcal{D}_X, \mathcal{D}_Y) \int_{x'} P(x'|\mathcal{D}_X, \mathcal{D}_Y) P(y|x', r, w, \mathcal{D}_X, \mathcal{D}_Y)\, dx'\, dr\, dw
\end{aligned}
$$

□

B.3 Intractability of $P(y|do(x), do(\mathcal{D}_X), \mathcal{D}_Y)$

Proof. Proof of the intractability of $P(y|do(x), do(\mathcal{D}_x), \mathcal{D}_y)$. We show that estimating this query from observational data requires to marginalise on $\mathcal{D}_X$. We take as a postulate that this is an intractable operation because it implies to access all possible input domains. We also note that estimating this operation via sampling can be achieved but goes against the purpose of domain generalisation as we aim to adapt to new domains from limited information.

From the causal graph in Figure 1b, we observe two causal paths linking $\mathcal{D}_X$ to Y :

1. A direct path: $\mathcal{D}_X \rightarrow \mathcal{D}_R \rightarrow W \rightarrow Y$
2. A backdoor path: $\mathcal{D}_X \leftarrow Z \rightarrow X \rightarrow R \rightarrow Y$
3. A backdoor path: $\mathcal{D}_X \leftarrow U_{XY} \rightarrow Y$

The third path is blocked by the do operation on X but the first two paths realise the frontdoor criterion as shown in the simplified causal graph in Figure 4. The expression can therefore be simplified as follows:

$$
\begin{aligned}
& P(y|do(x), do(\mathcal{D}_X), \mathcal{D}_Y) \\
&= \int_a P(a|\mathcal{D}_X, do(x), \mathcal{D}_Y) \int_{\mathcal{D}'_X} P(y|do(x), \mathcal{D}'_X, \mathcal{D}_Y) P(\mathcal{D}'_X|do(x), \mathcal{D}_Y)\, d\mathcal{D}'_X && \textit{frontdoor criterion}
\end{aligned}
$$

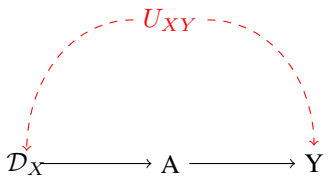


Figure 4: Simplified causal graph linking the input domain $\mathcal{D}_X$ to the output label Y. The paths can be simplified as a backdoor path via U_{XY} and a direct path through an an abstract variable A.

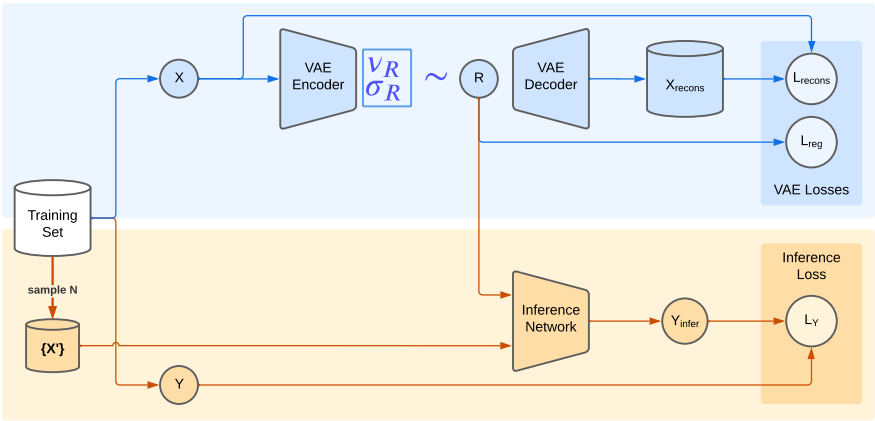


Figure 5: Illustration of the Causal Transportability baseline model.

Answering this query from observations requires marginalising on D_X, as highlighted in blue.

We can see graphically that this path cannot be blocked by any other variable than $\mathcal{D}_X$ because the variable U_{XY} is not observed, therefore marginalising on $\mathcal{D}_X$ is necessary.

□

C Details on the Causal Transportability Model

We implement the Causal Transportability algorithm defined by Mao et al. [24] as a baseline. The model is illustrated in Figure 5. The algorithm realises the following quantity:

$$P(y|do(x)) = P(r|x) \sum_{k=1}^{N} P(y|r, x_i) P(x_i) \tag{13}$$

The sampling strategy is the same as the one used in our work. The authors use a small network composed of three convolution layers followed by two fully connected layers to realise the quantity $P(y|r, x_i)$. For a fair comparison with our model, we replace it with a modified ResNet-18. We alter the first layer is modified to take a concatenation of an input image x with the representation r. The rest of the network is left untouched. The quantity $P(r|x)$ can be obtained via several methods. We use a VAE in our implementation, as the main method described in the original paper. We use the model in two settings. First, we jointly train the VAE and the inference network. Second, we compare this baseline against a second model where we train the VAE separately on a reconstruction task and then include the weights to the full model. We allow the training of the VAE weights at the second stage as we observed that freezing the weights leads to decreased performance.

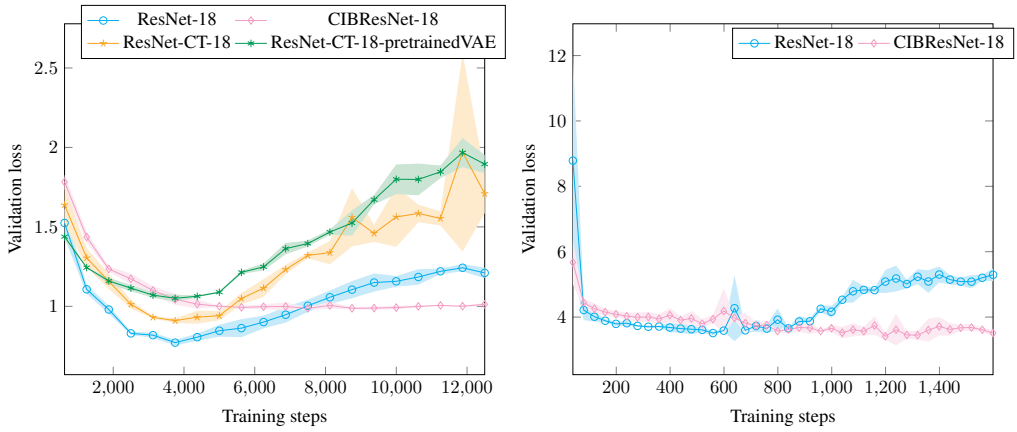
D Experimental Details

We compare our CIBResNet-18 against ResNet-18 and ResNet-18-CT in i.i.d and o.o.d settings. The second baseline is the causal-Transportability (CT) algorithm of Mao et al. [24]. Following their methodology, we use a VAE for the $P(R|X)$ encoder. Their proposed inference network is a three-layers convolution network followed by a two-layers dense classifier. For a fair comparison, we use a modified ResNet-18 (the same size as our model) that takes input and contextual information. More details are given in Appendix C. We conduct our experiments on a single GPU Nvidia A100 with the AdamW optimiser [19]. We initialise the Bayesian classifier parameters from the Normal distribution $\mathcal{N}(0, 1)$. On CIFAR-10, we train our model and the baselines for 20 epochs. We find that all models usually do not show improvements after 10 epochs. On OFFICEHOME, we train the models for 80 epochs. The training sets contain less samples than CIFAR-10 and more epochs are required for convergence. ResNet-18 stops improving after 40 epochs while CIBResNet-18 keep improving until around 60 epochs. We use hyperparameter grid search on the validation set of CIFAR-10 to set the hyperparameters. We use the default learning rate of 0.005 for the baselines and 0.01 for CIBResNet-18. We use a batch size of 64. By default, we use $N = 16$, $M = 16$, $\alpha = 0.4$, $\beta = 0.01$, $\gamma = 10^{-6}$, $\nu = 10^{-6}$ and $\epsilon = 10^{-6}$. The remaining hyperparameters are set to their default values. When evaluating the models in o.o.d settings, we use the batch statistics to build the mean and variance in the batch normalisation layers instead of the values learned on the training set that follow a different distribution.

E Additional Experiments

E.1 Training Stability

We compare the training of our model with the baseline on CIFAR-10 and the first domain of OFFICEHOME in Figure 6. We observe that the validation accuracy of the baselines first decreases to reach a plateau but then increases again, highlighting an overfitting to the training data. However, this behaviour is not observed with CIBResNet-18, which steadily decreases across the entire training, demonstrating higher stability. We can also note that our proposed model yields a lower standard deviation in the accuracy.



(a) CIFAR-10 dataset.

(b) OFFICEHOME-RealWorld dataset.

Figure 6: Evolution of the validation loss during training. The mean and standard deviation across three runs are shown. After a period of improvement, the ResNet-18 and ResNet-18-CT baselines overfit as the training progresses. This behaviour is not observed with CIBResNet-18, which also demonstrates a lower standard deviation.

E.2 Impact of Sampling on Posterior

We perform additional experiments to investigate the impact of the number of samples. We vary the quantities N and M of context images and inference weights, respectively. Figure 7 shows the

evolution of the accuracy of our model when varying N and M. The main contributing factor for performance is the number of context samples: using four context samples instead of one significantly improves accuracy. We hypothesize that a lower number is not sufficient to be representative of the distribution and have an adversarial effect instead. Figure 8 shows the evolution of the accuracy during training for several values of N and M. We observe that increasing the number of weight samples M improves sample efficiency and reduces the amount of training steps required to converge.

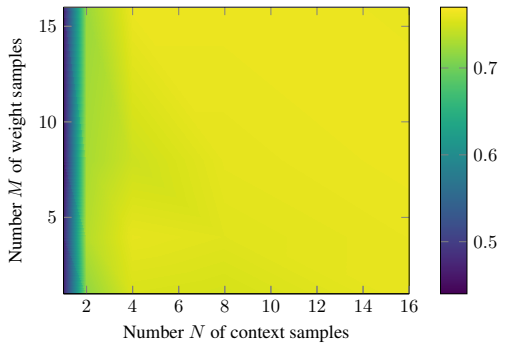


Figure 7: Accuracy heatmap of CIBResNet-18 on the CIFAR-10 test set as a function of the number of weight and context samples. The amount of weight samples has a negligible effect on performance. Only increasing the context size improves accuracy.

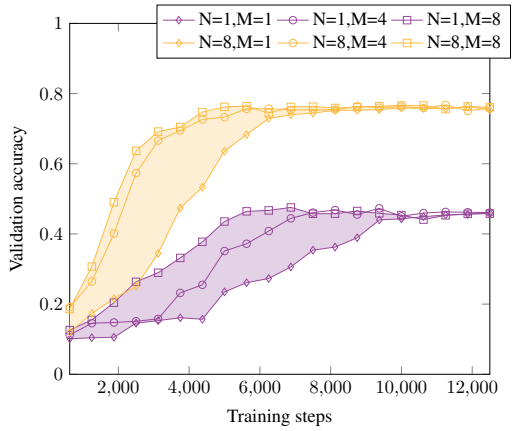


Figure 8: Evolution of the validation accuracy of CIBResNet-18 on CIFAR-10 during training with varying context images N and weight samples M. Increasing the weight samples reduces the amount of training steps required for learning.