HIERARCHICAL VARIATIONAL AUTO-ENCODING FOR UNSUPERVISED DOMAIN GENERALIZATION

Xudong Sun *
Department of Statistics
Ludwig Maximilian's University of Munich
Germany
smilesun.east@gmail.com

Florian Buettner
Siemens AG
Germany
buettner.florian@siemens.com

ABSTRACT

We address the task of domain generalization, where the goal is to train a predictive model such that it is able to generalize to a new, previously unseen domain. We choose a generative approach within the framework of variational autoencoders and propose an unsupervised algorithm that is able to generalize to new domains without supervision. We show that our method is able to learn representations that disentangle domain-specific information from class-label specific information even in complex settings where domain structure is not observed during training. Our interpretable method outperforms previously proposed generative algorithms for domain generalization and achieves competitive performance compared to state-of-the-art approaches, which rely on observing domain-specific information during training, on the standard domain generalization benchmark dataset PACS. Additionally, we proposed weak domain supervision which can further increase the performance of our algorithm in the PACS dataset.

1 BACKGROUND AND MOTIVATION

One big challenge of deploying a neural network model in real world use-cases is domain shift. In many real world applications, data seen by a deployed model is drawn from a distribution that is different from the training distribution and often unknown at train time. Domain Generalization aims at training a model from a set of domains (i.e. related distributions) such that the model is able to generalize to a new, unseen domain at test time. domain shift can manifest itself in a continuous manner, where for example the data distribution seen by an industrial asset can change due to wear and tear or due to maintenance procedures. Similarly, domain sub-structures are not always observable during training due to data privacy concerns (in particular when patient data is used). In these latter scenarios, it is difficult to train standard domain generalization algorithms since they are based on the notion of clearly separable domains that are observable during model training. In many of these use cases, interpretability and human oversight of machine learning models is key. Generative models allow for learning disentangled representations that correspond to specific and interpretable factors of variation, thereby facilitating transparent predictions.

We propose a new generative model that solves domain generalization problems in an interpretable manner without requiring domain labels during training. We build on previous work using autoencoder-based models for domain generalization (Kingma & Welling, 2013; Ilse et al., 2019) and propose a Hierarchical Domain Unsupervised Variational Auto-encoding that we refer to as HDUVA. Our major contributions include:

- We present an unsupervised algorithm for domain generalization that is able to learn in setting with incomplete or hierarchical domain information. Our algorithm only need to use extended ELBO as model selection criteria, instead of relying on the validation set.
- Our method is able to learn representations that disentangle domain-specific information from class-label specific information without domain supervision even in complex settings.

*Work partially done during research internship at Siemens AG. see https://arxiv.org/pdf/2101.09436.pdf for potential paper update

- Our algorithm generates interpretable domain predictions that reveal connections between domains.
- We constructed several hierarchical and sequential domain generalization benchmark datasets with doubly colored mnist for the domain generalization community.

2 METHODS AND TECHNICAL SOLUTION

2.1 PROBLEM STATEMENT AND NOTATION

Domain generalization aims to generalize models to unseen domains without knowledge about the target distribution during training. A domain d consists of a joint distribution $p(x, y)$ on $\mathcal{X} \times \mathcal{Y}$, with $\mathcal{X}$ being the input space and $\mathcal{Y}$ being the output space (Muandet et al., 2013). For our modelling approach, we employ the framework of variational autoencoders (VAEs) (Kingma & Welling, 2013). We use z to represent the latent representation of a VAE and use three independent latent representations to disentangle variability in inputs X related to domain-specific sources, label-specific sources and residual variation. We use probabilistic graphical models to illustrate the conditional dependecies of random variables, observables and hyperparameters in Figure 1. In the graphical model of Figure 1, solid circles represent observations and white circles represent latent variables. We use half-shaded circles to represent a variable can either be observed or act as latent variable, which is typical in semi-supervised learning. Small solid circles in Figure 1 represent fixed hyper-parameters. Subscripts represent components of a variable, while we use super-script to index samples and domains. We use solid arrows to represent generative path, and dashed arrows to represent variational inference part. Plates represent repetitions of random variables. We use θ to represent learnable parameters of priors/decoders and ϕ to represent learnable parameters of variational posterior distributions/encoders.

2.2 HDUVA OVERVIEW

To overcome the limitations of current autoencoder-based ethods, we propose a hierarchical probabilistic graphical model called Hierarchical Domain Unsupervised Variational Autoencoding(HDUVA). Our model is based on three latent variables are used to model distinct sources of variation that are denoted as z_y, z_d and z_x. z_y represents class specific information, z_d represents domain specific information and z_x models residual variance of the input. We model the prior distribution of z_y as conditional distribution based on class label y. These settings constitutes the inductive bias of the model (Locatello et al., 2019). We introduce an additional hierarchical level and use a continuous latent representation s to model (potentially unobserved) domain structure. This means that we can encourage disentanglement of the latent variables through conditional priors without the need of conditioning on a one-hot-encoded, observed domain label.
More specifically, we first place a Dirichlet prior on s such that it can be interpreted as a soft, topic-like, version of the standard one-hot encoded domain d. We then use z_d to capture domain-specific variation by conditioning its prior on s. Note that in our model this domain s is not an observable but instead a latent variable to be inferred from data. For clarity, we refer to an observed domain as nominal domain. Borrowing from topic models in NLP (Srivastava & Sutton, 2017), we refer to s as **topic**. We illustrate HDUVA in form of a graphical model in Figure 1, where we form a hierarchical path Klushyn et al. (2019) from topic s to z_d to observation x. We use K to denote the dimension of the domain representation or topic vector s, i.e. $dim(s) = K$. We use k to index each component of s, i.e. $s^{(l)} = [s_1^{(l)}, s_2^{(l)}, \cdots, s_K^{(l)}]$, with l indexing a domain. Note that in our case, K can be either greater, smaller or equal to the number of domains L, while in supervised approaches, the one-hot encoded domain label is always the size of L. This is beneficial not only in setting with unobserved domain observation, but also for problems with a large number of domains which lie on a lower-dimensional manifold (e.g. thousands assets in an predictive maintenance task). In this case, when choosing the topic dimension K to be smaller than the number of training domains, our algorithm can be interpreted as an eigen-domain decomposition algorithm. We use stochastic gradient descent to train our model. Accordingly, in Figure 1, the batch size is denoted by $M^{(l)}$ for the lth domain, with a total of $N^{(l)}$ batches for domain l. We use i to index a batch and j to index a sample. For simplicity, i and j are omitted whenever convenient and not causing confusion. Details

on model implementation and inference are described in the supplement. We also provide details on extensions for weak supervision to our model in the supplement.

Taken together, we present a novel approach for probabilistic domain generalization without the need for observed domain labels. Further details about the prior setting and inference process for our method can be found in the supplement material.

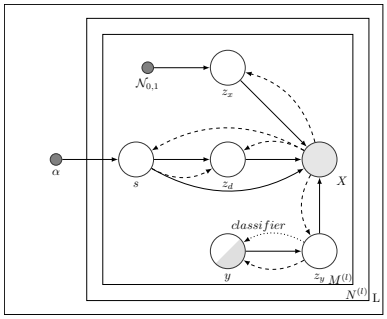


Figure 1: HDUVA:Hierarchical Domain Unsupervised Variational Auto-encoding

Table 1: **Malaria Virtual Hospital from Malaria Dataset**. Patients with ID starting with C1, C6, C8, C9 are grouped to form 4 virtual hospitals as 4 nominal domains. We report result on the test domain corresponding to virtual hospital C1. 20 repetitions are done where each repetition sub-sample 20 percent of the training domains data as training data and the rest as validation. Comparison algorithms are DIVA (Ilse et al., 2019) and Match-DG (Mahajan et al., 2020), while Deep-All is used as baseline by pooling all training domains together. Further detail can be found in supplement B. Data source: (Rajaraman et al., 2018)

Malaria Cell Classification	**Test Accuracy**
DIVA	0.83 ± 0.06
HDUVA	$\mathbf{0.87} \pm 0.05$
Deep-All	0.84 ± 0.05
MatchDG	0.85 ± 0.09

3 Empirical Evaluation

We conduct experiments, trying to answer the following questions:

- Could HDUVA mitigate the limitations of standard supervised approaches for domain generalization in terms of domain-substructure or incomplete domain information? We conduct experiments based on a real-world dataset for the classification of Malaria patients.
- How does HDUVA perform under standard domain generalization benchmarks where information on clearly separated domain is available, compared with other state-of-the-art algorithms? See Section 3.2.

In addition we provide results from additional experiments in the supplement, where we use simulated data to further illustrate that our algorithm outperforms standard domain generalization approaches in case of incomplete domain information. In the supplement we also visualize topics from overlapping nominal domains to illustrate why HDUVA improves upon supervised approaches.

3.1 Domain Generalization to Medical Image Classification

Trustworthy prediction is essential for biomedical data where domain generalization poses a great challenge (Gossmann et al., 2019; 2020). For example, medical imaging datasets usually come from a multitude of patients and devices where both the patient and devices can form domains. In this study, as suggested by Ilse et al. (2019), we consider hospital as domains, which consist of patients as sub-domains. This correspond to hierarchical domains and has practical implications. Since there can be thousands of patients, and having thousands of domain labels can be impractical and many patients can share common features, e.g. coming from nearby areas, but it can also be true that two hospitals can have similar patients. To simulate such a setting, we construct virtual hospitals by using the Malaria dataset as described in Table 1. The Malaria dataset (Rajaraman et al., 2018) consist of thin blood smear slide images of segmented cells from Malaria patients. We group patients by their IDs for form hospitals. Table 1 shows the out of domain classification accuracy across different algorithms. Our approach is able to implicitly learn the unobserved domain substructure of the data,

Table 2: **Domain Generalization in PACS Dataset with AlexNet Classifier**

Methods	**Art Painting**	**Cartoon**	**Photo**	**Sketch**	**Ave.**
Factorization (Li et al., 2017)	0.63	0.67	0.90	0.58	0.69
MLDG (Li et al., 2018)	0.66	0.67	0.88	0.59	0.70
SourceCombo (Mancini et al., 2018)	0.64	0.67	0.90	0.60	0.70
MetaReg (Balaji et al., 2018)	**0.70**	0.70	**0.91**	0.59	0.73
GLCM (Wang et al., 2019)	0.67	0.70	0.88	0.56	0.70
Jigsaw (Carlucci et al., 2019)	0.68	**0.72**	0.89	**0.65**	**0.74**
AFLAC (Akuzawa et al., 2018)	0.61	0.64	0.83	0.59	0.67
MatchDG (Mahajan et al., 2020)	0.67 $\pm$0.01	0.69 $\pm$ 0.01	0.89 $\pm$ 0.01	0.63 $\pm$ 0.02	0.72
DIVA (Ilse et al., 2019)	0.64 $\pm$ 0.02	0.66 $\pm$ 0.003	0.87 $\pm$ 0.01	0.58 $\pm$ 0.03	0.69
Deep-All	0.64 $\pm$ 0.01	0.67 $\pm$ 0.02	0.85 $\pm$ 0.02	0.56 $\pm$ 0.02	0.68
HDUVA	0.65 $\pm$ 0.01	0.66 $\pm$ 0.01	0.87 $\pm$ 0.01	0.58 $\pm$ 0.01	0.69
WHDUVA*	0.64 $\pm$ 0.01	0.67 $\pm$ 0.02	0.88 $\pm$ 0.01	0.60 $\pm$ 0.02	0.70
HDUVA-no-zx****	0.65$\pm$ 0.01	0.67$\pm$0.02	0.87 $\pm$ 0.01	0.61 $\pm$ 0.03	0.70
LHDUVA**	0.65 $\pm$ 0.003	0.69 $\pm$ 0.02	0.87 $\pm$ 0.004	0.55 $\pm$ 0.002	0.69
HDUVA-CTR***	0.65 $\pm$ 0.01	0.66 $\pm$ 0.02	0.88 $\pm$ 0.01	0.63 $\pm$ 0.003	0.71

* WHDUVA: weak domain-supervision added to HDUVA as explained in Appendix H.
**LHDUVA: Ladder Hierarchical Domain Unsupervised Variational Auto-encodingexplained in Appendix G.
***HDUVA-CTR: Use the contrastive learning phase (pretrain phase) of Mahajan et al. (2020) as initialization for AlexNet of HDUVA.
****HDUVA-no-zx: HDUVA without z_x variable, see Figure 7.
Part of the table is adapted from `https://domaingeneralization.github.io/`.

resulting is substantially better accuracy on unseen test domains (i.e. a new hospital) compared to state-of-the-art approaches DIVA and MatchDG. The latter require explicit domain labels during training and fail to perform well in scenarios with domain substructure. Our approach also performs substantially better than a standard baseline where information across all domains is pooled together. Additionally, we only use the ELBO in Equation 8 as our model selection criteria, without using the validation set.

3.2 State of the art Domain Generalization benchmark

We finally compare HDUVA to state-of-the-art domain generalization algorithms for a standard domain generalization task, where domain information is available on largely different domains. Table 2 shows algorithm performance on the PACS dataset (Li et al., 2017) which is a popular domain generalization benchmark.We use AlexNet (Krizhevsky et al., 2012; 2017) as the neural network architecture for $q_{\phi_y}(z_y|x)$ and $q_\omega(y|z_y)]$ in our model in Equation 8. For fair comparison, the rest of the algorithms also use AlexNet as classifier.

Table 2 shows that the performance of HDUVA is comparable to state-of-the-art performances on the PACS dataset. Notably, HDUVA without using domain label ties DIVA, and with weak domain supervision variant introduced in Appendix H which we coin WHDUVA, the performance improves over DIVA. With the contrastive pretrain phase of Mahajan et al. (2020) as the initialization for the AlexNet of HDUVA, the performance over the sketch test domain further improves by 3 percent. Deep-All by pooling all training domain together remain a strong baseline where we outperform Deep-All in 3 out of 4 test domains and ties the other one. While overall performance of methods such as JIGSAW is consistently better than HDUVA, it is based on complex image manipulations. In contrast, HDUVA is an interpretable model that can be used for different data modalities and a larger number of tasks including domain prediction and sample generation. Importantly, HDUVA achieves competitive performance without using domain labels during training and without using validation set (we conduct model selection using extended ELBO in Equation 8, see supplement J). This enables domain generalization for a much wider range of use-cases than standard algorithms.

4 Conclusion

See section A.

REFERENCES

Kei Akuzawa, Yusuke Iwasawa, and Yutaka Matsuo. Domain generalization via invariant representation under domain-class dependency. 2018.

Yogesh Balaji, Swami Sankaranarayanan, and Rama Chellappa. Metareg: Towards domain generalization using meta-regularization. In *Advances in Neural Information Processing Systems*, pp. 998–1008, 2018.

Fabio M Carlucci, Antonio D'Innocente, Silvia Bucci, Barbara Caputo, and Tatiana Tommasi. Domain generalization by solving jigsaw puzzles. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2229–2238, 2019.

Xi Chen, Diederik P Kingma, Tim Salimans, Yan Duan, Prafulla Dhariwal, John Schulman, Ilya Sutskever, and Pieter Abbeel. Variational lossy autoencoder. *arXiv preprint arXiv:1611.02731*, 2016.

Alexej Gossmann, Kenny H Cha, and Xudong Sun. Variational inference based assessment of mammographic lesion classification algorithms under distribution shift. In *NeurIPS 2019 workshop: Medical Imaging meets NeurIPS 2019*, 2019.

Alexej Gossmann, Kenny H Cha, and Xudong Sun. Performance deterioration of deep neural networks for lesion classification in mammography due to distribution shift: an analysis based on artificially created distribution shift. In *Medical Imaging 2020: Computer-Aided Diagnosis*, volume 11314, pp. 1131404. International Society for Optics and Photonics, 2020.

Arthur Gretton, Karsten M Borgwardt, Malte J Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *The Journal of Machine Learning Research*, 13(1):723–773, 2012.

Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. 2016.

Haodi Hou, Jing Huo, and Yang Gao. Cross-domain adversarial auto-encoder. *arXiv preprint arXiv:1804.06078*, 2018.

Maximilian Ilse, Jakub M Tomczak, Christos Louizos, and Max Welling. Diva: Domain invariant variational autoencoders. *arXiv preprint arXiv:1905.10427*, 2019.

Martin Jankowiak and Fritz Obermeyer. Pathwise derivatives beyond the reparameterization trick. In *International conference on machine learning*, pp. 2235–2244. PMLR, 2018.

Weonyoung Joo, Wonsung Lee, Sungrae Park, and Il-Chul Moon. Dirichlet variational autoencoder. *Pattern Recognition*, 107:107514, 2020.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Alexej Klushyn, Nutan Chen, Richard Kurle, Botond Cseke, and Patrick van der Smagt. Learning hierarchical priors in vaes. *arXiv preprint arXiv:1905.04982*, 2019.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25:1097–1105, 2012.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84–90, 2017.

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*, pp. 5542–5550, 2017.

Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy M Hospedales. Learning to generalize: Meta-learning for domain generalization. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

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*, pp. 4114–4124. PMLR, 2019.

Divyat Mahajan, Shruti Tople, and Amit Sharma. Domain generalization using causal matching. *arXiv preprint arXiv:2006.07500*, 2020.

Massimiliano Mancini, Samuel Rota Bulo, Barbara Caputo, and Elisa Ricci. Best sources forward: domain generalization through source-specific nets. In *2018 25th IEEE International Conference on Image Processing (ICIP)*, pp. 1353–1357. IEEE, 2018.

Luke Metz, Ben Poole, David Pfau, and Jascha Sohl-Dickstein. Unrolled generative adversarial networks. *arXiv preprint arXiv:1611.02163*, 2016.

Krikamol Muandet, David Balduzzi, and Bernhard Schölkopf. Domain generalization via invariant feature representation. In *International Conference on Machine Learning*, pp. 10–18, 2013.

Sivaramakrishnan Rajaraman, Sameer K Antani, Mahdieh Poostchi, Kamolrat Silamut, Md A Hossain, Richard J Maude, Stefan Jaeger, and George R Thoma. Pre-trained convolutional neural networks as feature extractors toward improved malaria parasite detection in thin blood smear images. *PeerJ*, 6:e4568, 2018.

Danilo Jimenez Rezende and Fabio Viola. Taming vaes. *arXiv preprint arXiv:1810.00597*, 2018.

Casper Kaae Sønderby, Tapani Raiko, Lars Maaløe, Søren Kaae Sønderby, and Ole Winther. Ladder variational autoencoders. In *Advances in neural information processing systems*, pp. 3738–3746, 2016.

Akash Srivastava and Charles Sutton. Autoencoding variational inference for topic models. *arXiv preprint arXiv:1703.01488*, 2017.

Jakub Tomczak and Max Welling. Vae with a vampprior. In *International Conference on Artificial Intelligence and Statistics*, pp. 1214–1223. PMLR, 2018.

Aäron Van Den Oord, Nal Kalchbrenner, Oriol Vinyals, Lasse Espeholt, Alex Graves, and Koray Kavukcuoglu. Conditional image generation with pixelcnn decoders. *Advances in Neural Information Processing Systems*, (Nips):4797–4805, 2016. ISSN 10495258. arXiv: 1606.05328.

Riccardo Volpi and Vittorio Murino. Addressing model vulnerability to distributional shifts over image transformation sets. 2019.

Riccardo Volpi, Hongseok Namkoong, Ozan Sener, John C Duchi, Vittorio Murino, and Silvio Savarese. Generalizing to unseen domains via adversarial data augmentation. In *Advances in neural information processing systems*, pp. 5334–5344, 2018.

Haohan Wang, Zexue He, Zachary C Lipton, and Eric P Xing. Learning robust representations by projecting superficial statistics out. *arXiv preprint arXiv:1903.06256*, 2019.

Qizhe Xie, Zihang Dai, Yulun Du, Eduard Hovy, and Graham Neubig. Controllable invariance through adversarial feature learning. In *Advances in Neural Information Processing Systems*, pp. 585–596, 2017.

Tingting Zhao, Zifeng Wang, Aria Masoomi, and Jennifer G. Dy. Adaptive nonparametric variational autoencoder. *arXiv*, 2019. ISSN 23318422. arXiv: 1906.03288v2.

Supplemental Materials

A CONCLUSION

We proposed an Hierarchical Domain Invariant Variational Autoencoder, with the following improvements:

- Our approach does not require observed domain labels during training, facilitating domain generalization for a much wider range of applications. Additionally, our approach does not need validation set for model selection but only use extended ELBO for model selection.
- In the presence of domain-substructure, our algorithm is able to robustly disentangle domain-specific variation from class-label specific variation.
- Our algorithm is able to model domain overlap via interpretable topics and generalize to settings with continuous domain shift.
- Our algorithm has a competitive performance even in standard domain generalization tasks, where observed domain information is available on clearly separated domains.
- We proposed evaluation dataset for benchmarking hierarchical and sequential domain shift.

In the supplementary material, to facilitate easy reference, we use consecutive Figure and Table numbering following the main article. In section B, we introduce further details about the malaria experiment setting. In section C we introduce related work. In section D, we introduce technical details for HDUVA. In section E, we introduce further experimental settings and results which further shows the benefit of our algorithm. In section G we explain an alternative inference algorithm inspired by Ladder-VAE (Sønderby et al., 2016) for our proposed model. In section H, we introduce weak domain supervision methods for both inference algorithms. In section J, we list further details on experimental settings.

B FURTHER DETAIL ABOUT EXPERIMENT SETTING OF THE VIRTUAL MALARIA HOSPITAL

Virtual hospital C6 has 10 patients with 1061 infected cell images (in total 1748 images). Virtual hospital C8 has 10 patients with 957 infected cell images (in total 1638 images). Virtual hospital C9 has 10 patients with 1284 infected cell images (in total 1964 images). Virtual hospital C1 has 90 patients with 8023 infected cell images (in total 14190 images). Each time, we combine the C6, C8, C9 virtual hospital domain as 3 training domains and sample 20 percent of the images for training. 20 random repetitions are done. We report result on the test domain corresponding to virtual hospital C1.

C RELATED WORK

In this section, we provide a taxonomy of existing solutions in domain generalization. In general, domain generalisation approaches can be divided into the following main categories, that we describe below.

Invariant Feature Learning While observations from different domains follow different distributions, Invariant Feature Learning approaches try to map the observations from different domains into a common feature space, where domain information is minimized (Xie et al., 2017; Akuzawa et al., 2018). The method works in a mini-max game fashion in that there is a domain classifier trying to classify domains from the common feature space, while a feature extractor tries to fool this domain classifier and help the target label classifier to classify class label correctly. Li et al. (2017) presented a related approach and used tensor decomposition to learn a low rank embedding for a set of domain specific models as well as a base model. We classify this method into invariant feature learning because the base model is domain-invariant.

Image Processing Based Method Carlucci et al. (2019) divided the image into small patches and generated permutations of those small patches. They then used a deep classifier to predict the predefined permutation index so that the model learned the global structure of an image instead

of local textures. Wang et al. (2019) used a gray level co-occurence matrix to extract superficial statistics. They presented two methods to encourage the model to ignore the superficial statistics and thereby learn robust representations. This group of methods has been developed for image classification tasks, and it is not clear how it can be extended to other data types.

Adversarial Training Based Data Augmentation Volpi & Murino (2019) optimized a procedure to search for worst case adversarial examples to augment the training domain. Volpi et al. (2018) used Wasserstein distance to infer adversarial images that were close to the current training domain, and trained an ensemble of models with different search radius in terms of Wasserstein distance.

Meta Learning Based Method Meta learning based domain generalization method (MLDG) uses model agnostic training to tackle domain generalization as a zero-shot problem, by creating virtual train and test domains and letting the meta-optimizer choose a model with good performance on both virtual train and virtual test domains (Li et al., 2018). Balaji et al. (2018) improved upon MLDG by concatenating a fixed feature network with task specific networks. They parameterized a learnable regularizer with a neural network and trained with a meta-train and a meta-test set .

Auto-Encoder Based Method DIVA (Ilse et al., 2019) builds on variational auto-encoders and splits the latent representation into three latent variables capturing different sources of variation, namely class specific information (z_y), domain specific information (z_d) and residual variance (z_x). Disentanglement is encouraged via conditional priors, where the domain-specific latent variable z_d is condition on an observed, one-hot-encoded domain d. As auxiliary components, DIVA adds a domain classifier based on z_d, as well as a target class label classifier based on z_y. Hou et al. (2018) encoded images from different domains in a common content latent code and domain-specific latent code, while the two types of encoders share layers. Corresponding discriminators are used to predict whether the input is drawn from a prior distribution or generated from encoder.

Causality based Method Recently, Mahajan et al. (2020) proposed MatchDG with that approximates base object similarity by using a contrastive loss formulation adapted for multiple domains. The algorithm then match inputs that are similar under the invariant representation.

Comparing these families of approaches, we can see that only probabilistic auto-encoder based models inherit advantageous properties like semi-supervised learning, density estimation and variance decomposition naturally. While autoencoder-based approaches such as DIVA have a better interpretability than all other approaches, a main drawback is that explicit domain labels are required during training.This can be problematic in a number of settings. In particular, a one-hot encoding of domains does not reflect scenarios where a continuous domain shift can occur. In this case, without knowledge of the causal factor that causes the domain shift, it is not clear how such continuous shifts can be one-hot encoded in a meaningful manner. In addition,

- Domains can have a hierarchical structure reflected by related sub-domains (e.g. country $>$ factory $>$ machine). One-hot encodings as used in existing autoencoder-based approaches are not able to model such hierarchical domain structures.
- In some applications, domains are not necessarily well-separated, but significant overlap between domains can occur (e.g. a cartoon might look more similar to a pop-art painting than a photography). One-hot encoding such overlapping domains encourages separated representations, which may harm model performance.
- A one-hot encoding of domains mapping to the prior distribution of z_d may limit the generalization power of neural networks, especially when we deal with continuous domain shift.

D METHODS AND TECHNICAL SOLUTION

In this section, we provide technical details to HDUVA in the main text.

D.1 MODEL IMPLEMENTATION

In this section, we first describe the generative model with prior distributions, followed by a discussion on model inference.

D.1.1 PRIOR DISTRIBUTIONS FOR z_x, z_y AND z_s

We chose a standard isotropic Gaussian prior with zero mean and unit variance for z_x and conditional priors for for z_y and z_d. More specifically, we chose a normal prior for z_y that is conditioned on the target class label y:

$$p_{\theta_y}(z_y^{(l,i)}|y^{(l,i)}) = \mathcal{N}\left(\cdot|\mu_{\theta_y}(y^{(l,i)}), \sigma_{\theta_y}(y^{(l,i)})\right) \tag{1}$$

with μ_{θ_y} and σ_{θ_y} being learnable parameterizations of the mean and standard deviation in form of neural networks. Similarly, we choose a normal prior for z_d and condition it on s:

$$p_{\theta_d}(z_d^{(l,i)}|s^{(l,i)}) = \mathcal{N}\left(\cdot|\mu_{\theta_d}(s^{(l,i)}), \sigma_{\theta_d}(s^{(l,i)})\right) \tag{2}$$

where again μ_{θ_d} and σ_{θ_d} parameterize mean and variance of z_d.

D.1.2 PRIOR DISTRIBUTION FOR s

We would like for s to display topic-like characteristics, facilitating interpretable domain representations. Consequently, we use a Dirichlet prior on s, which is a natural prior for topic modeling (Srivastava & Sutton, 2017; Joo et al., 2020; Zhao et al., 2019).

Let α be the Dirichlet concentration parameter $\alpha = [\alpha_1, \alpha_2, \cdots, \alpha_K]$, then the prior distribution of s can be written as:

$$p(s^{(l,i)}|\alpha^l) = Dir(s^{(l,i)}|\alpha^l_{1:K}) = \frac{\prod_k (s_k^{(l,i)})^{\alpha_k^l - 1}}{\mathcal{Z}(\alpha^l_{1:K})} \tag{3}$$

where we use $\mathcal{Z}(\alpha_{1:K})$ to represent the partition function.

We do not learn the distribution parameter α, but instead, leave it as a hyper-parameter. By default, we set α to be a vector of ones, which corresponds to a uniform distribution of topics. We refer to this prior setting as flat prior. If more prior knowledge about the relation between training domains is available, an informative prior can be used instead.

D.1.3 INFERENCE FOR HDUVA

Since exact inference is intractable in such an autoencoder, we perform variational inference and introduce three separate encoders as follows:

$$\begin{aligned} &q_\phi(s^{(l,i)}, z_d^{(l,i)}, z_x^{(l,i)}, z_y^{(l,i)}|x^{(l,i)}) \\ =&q_{\phi_s}(s^{(l,i)}|x^{(l,i)}) q_{\phi_d}(z_d^{(l,i)}|s^{(l,i)}, x^{(l,i)}) q_\phi(z_x^{(l,i)}, z_y^{(l,i)}|x^{(l,i)}) \end{aligned} \tag{4}$$

For the approximate posterior distributions of z_x and z_y, we assume fully factorized Gaussians with parameters given as a function of their input:

$$q_\phi(z_x^{(l,i)}, z_y^{(l,i)}|x^{(l,i)}) = q_{\phi_x}(z_x^{(l,i)}|x^{(l,i)}) q_{\phi_y}(z_y^{(l,i)}|x^{(l,i)}) \tag{5}$$

Encoders q_{ϕ_s} , q_{ϕ_d}, q_{ϕ_y}, and q_{ϕ_x} are parameterized by ϕ_s, ϕ_d, ϕ_y, and ϕ_x using separate neural networks to model respective means and variances as function of x.

For the form of the approximate posterior distribution of the topic s we chose a Dirichlet distribution:

$$q_{\phi_s}(s^{(l,i)}|x^{(l,i)}) = Dir\left(s^{(l,i)}|\phi_s(x^{(l,i)})\right) \tag{6}$$

where ϕ_s parameterizes the concentration parameter based on x, using a neural network. We use the technique in Jankowiak & Obermeyer (2018) to reparameterize the Dirichlet distribution.

D.1.4 ELBO FOR HDUVA

Given the priors and factorization described above, we can optimize the model parameters by maximizing the evidence lower bound (ELBO). We can write the ELBO for a given input-output tupel (x, y) as:

$$\begin{aligned} ELBO(x,y) &= E_{q(z_d,s|x),q(z_x|x),q(z_y|x)} \log p_\theta(x|s,z_d,z_x,z_y) \\ &- \beta_x KL(q_{\phi_x}(z_x|x)||p_{\theta_x}(z_x)) - \beta_y KL(q_{\phi_y}(z_y|x)||p_{\theta_y} \\ &(z_y|y)) - \beta_d E_{q_{\phi_s}(s|x),q_{\phi_d}(z_d|x,s)} \log \frac{q_{\phi_d}(z_d|x,s)}{p_{\theta_d}(z_d|s)} \\ &- \beta_s E_{q_{\phi_s}(s|x)} KL(q_{\phi_s}(s|x)||p_{\theta_s}(s|\alpha)) \end{aligned} \tag{7}$$

where we use β to represent the multiplier in the Beta-VAE setting (Higgins et al., 2016), further encouraging disentanglement of the latent representations.

Finally, we add an auxiliary classifier $q_\omega(y|z)$, which is parameterized by ω, to encourage separation of classes y in z_y. The HDUVA objective then becomes:

$$\mathcal{F}(x,y) = ELBO(x,y) + \gamma_y E_{q_{\phi_y}(z_y|x)}[\log q_\omega(y|z_y)] \tag{8}$$

The whole process is described in Algorithm 1. The objective function in Equation 8 which we coin extended ELBO can also be used as a model selection criteria, thus our method does not need validation set at all, as we empirically evaluated in the experimental section in section 3.

Algorithm 1 HDUVA

while not converged or maximum epochs not reached **do**
  warm up β defined in Equation 7, as in Sønderby et al. (2016)
  fetch mini-batch {x, y} ={$x^{(l,i)}, y^{(l,i)}$}
  compute parameters for $q_{\phi_x}(z_x|x)$, $q_{\phi_y}(z_y|x)$, $q_{\phi_s}(s|x)$, $q_{\phi_d}(z_d|s,x)$
  sample latent variable z_x^q, z_y^q, s^q, z_d^q and compute $[\log q_\omega(y|z_y)]$.
  compute prior distribution for z_d using s
  compute $p_\theta(x|z_x, z_y, z_d, s)$ using sampled s, z_x^q, z_y^q, z_d^q
  compute KL divergence for z_d, z_x and z_y, s.
  aggregate loss according to Equation 8 and update model
end while

E FURTHER EXPERIMENTAL SETTINGS AND RESULTS

After a detailed explanation to HDUVA, we supplement the the experimental section further with the following scenarios.

E.1 HIERARCHICAL DOMAINS

To simulate domains with sub-structures (hierarchical domains), we create sub-domains within nominal domains. All sub-domains within one nominal domain share the same domain label. We adapt color-mnist (Metz et al., 2016; Rezende & Viola, 2018) with the modification that both its foreground and background are colored as sub-domain, as shown in Figure 2. We constructed 3 nominal domains with sub-structures as indicated in Figure 2. For baseline algorithms, we use a one-hot encoded nominal domain label as explicit domain label, since these methods require a domain label during training. For HDUVA, we do not use domain label and we only use extended ELBO in Equation 8 as model selection criteria, further experimental details can be found in supplement J.

We are interested in evaluating how our unsupervised approach and supervised generative domain generalization algorithms like DIVA Ilse et al. (2019) for domain generalization would behave under this sub-domain scenario, in terms of out-of-domain prediction accuracy and disentanglement performance. We perform a leave-one-domain-out evaluation (Li et al., 2017), where each test domain is repeated 10 times with 10 different random seeds. We report the out of domain test accuracy in

(a) 1st domain (b) 2nd domain (c) 3rd domain

Figure 2: **Random combination of Color-Mnist as Hierarchical Domains**. Mnist has both its foreground and background colored, each color combination represent one sub-domain. Each nominal domains include 2 sub-domains.

Table 3: **Out of Domain Accuracy on Color-Mnist Composed Subdomain inside Nominal Domains**

Color-Mnist (Figure 2)	Test Domain 1	Test Domain 2	Test Domain 3
HDUVA	**0.93** $\pm$ 0.02	**0.69** $\pm$ 0.12	**0.55** $\pm$ 0.03
DIVA (Ilse et al., 2019)	0.88 $\pm$ 0.05	0.56 $\pm$ 0.19	0.50 $\pm$ 0.08

Table 3. Table 3 shows that HDUVA outperforms DIVA in terms of out of domain performance on all three test domains, while retaining a very small variance compared to DIVA.

To explain such a performance difference, we further evaluate how robustly DIVA and HDUVA are able to disentangle different sources of variation under this scenario with incomplete sub-domain information.

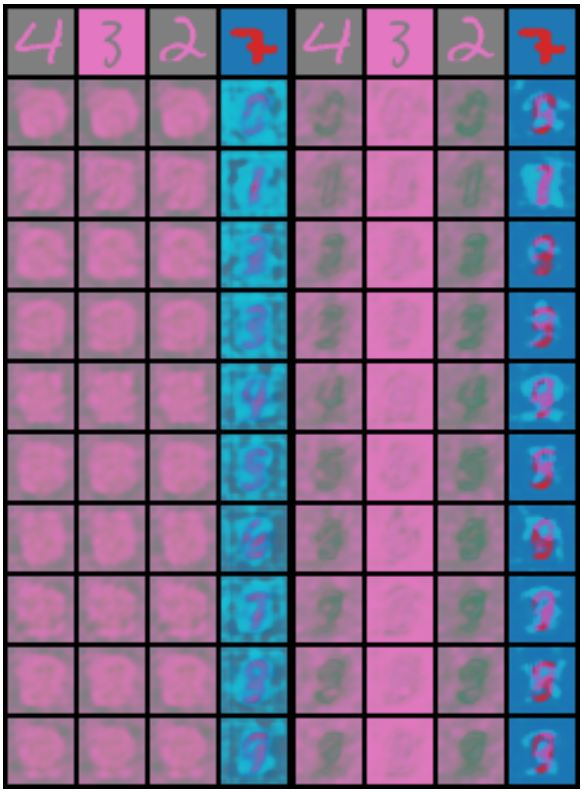
We sample seed images from different sub-domains as shown in the first row of Figure 3. We then generate new images by scanning the class label from 0 to 9 by sampling from the conditional prior distribution of z_y (i.e. $p_{\theta_y}(z_y|y)$, eq. 1). We keep the domain representation the same as in the seed image, set the noise component z_x to zero and then use the decoder network $p_\theta(x|z_d, z_x, z_y)$ to generate an image based on the three latent representations. If the models are able to disentangle domain-specific variation from class-label specific variation in z_y and z_d, we expect that the generated images have the same domain information as the seed image (foreground and background color) while generating different class labels (numbers from 0 to 9). In Figure 3 we compare DIVA and HDUVA's generative performance. Due to the sub-structure inside the nominal domains, DIVA could only reconstruct a blur of colors for the first 3 columns in Figure 3a, while HDUVA could generate different numbers for 2 of the three seed images. For the last seed image, both DIVA and HDUVA could conditionally generate numbers, but DIVA did not retain the domain information (since the background color, which is dark blue in the seed image, is light blue in the generated images). This indicates that DIVA is not able to disentangle the different sources of variation and domain information is captured by z_y as well. In contrast, HDUVA was able to separate domain information from class-label information.

E.2 Generalization under domain-drift scenarios

In some occasions, the boundaries between different domains can be ambiguous. For example, consider continuous domain drift in industry applications, some physical parameters of the same type of machine in different factories might change continuously and between two factories there can be overlap.

To simulate such a behavior, we consider a domain-drift scenario with Color-Mnist in Figure 4. By dividing a smooth color palette into 7 sub-domains (with each color corresponding to a sub-domain), we simulate a near continuous domain shift. We use this scenario to evaluate how robust our algorithm is in domain drift scenarios.

Following leave-one-domain out setting as in other experiments, we report the out-of-domain classification accuracy in Table 4, illustrating that our unsupervised approach is better able to account for continuous domain drift scenarios than standard supervised approaches that artificially categorize



(a) DIVA (b) HDUVA

Figure 3: **Comparison of Conditional Image Generation under Incomplete Domain Knowledge**. The domain composition is shown in Figure 2.

(a) 1st domain (b) 2nd domain (c) 3rd domain

Figure 4: **Sequential Color-Mnist (VLAG Palette)**. Background color taking 7 hue values spanning the `VLAG` hue ranges sequentially, with fixed saturation and lightning, foreground color takes equally spaced hue value in the complete hue circle with fixed saturation and lightning. Background and foreground colors are zipped. The first 3 color schemes representing 3 sub-domains compose the first nominal domain in Figure 4a, the 2nd nominal domain in Figure 4b takes the middle 3 color schemes with one color scheme overlap with the 1st and one color scheme overlap with 3rd nominal domain in Figure 4c. The 2nd nominal domain serves as a bridge between the other two nominal domains. Out of domain test accuracy is reported in Table 4.

the gradually shifting into distinct nominal domains. For HDUVA, we only use extended ELBO in Equation 8 as model selection criteria, further experimental details can be found in supplement J.

Table 4: **Out of Domain Accuracy for Color-Mnist (VLAG Palette) in Figure 4**. Each sub-domain in Figure 4 contains a random sample of 1000 mnist images. Random seed is shared for the different sub-domains of a nominal domain but different across nominal domains. Each repetition is with different starting random seed, 10 repetitions are done. The sub-domains are combined to form one nominal domain and 50 percent is used for training, the rest for validation. Comparison algorithms are DIVA (Ilse et al., 2019) and Match-DG (Mahajan et al., 2020), while Deep-All is used as baseline by pooling all training domain s together. The 2nd domain is a bridge domain that connect the 1st and 3rd domain, so it is not used as test domain at all.

Color-Mnist (Figure 4)	**Test Domain 1**	**Test Domain 3**
DIVA	0.63 ± 0.05	0.68 ± 0.03
HDUVA	$\mathbf{0.69} \pm 0.05$	$\mathbf{0.71} \pm 0.03$
Deep-All	0.60 ± 0.05	0.68 ± 0.04
Match-DG	0.67 ± 0.06	0.70 ± 0.03

(a) 1st domain (b) 2nd domain (c) 3rd domain

Figure 5: **Sequential Color-Mnist (Red Diverging Palette)**. Background color taking 7 hue values spanning in the area between 0 and 350 hue degrees (red spectrum) sequentially, with fixed saturation and lightning, foreground color takes equally spaced hue value in the complete hue circle with fixed saturation and lightning. Background and foreground colors are zipped. The first 3 color schemes representing 3 sub-domains compose the first nominal domain in Fig. 5a, the 2nd nominal domain take the middle 3 color schemes with one color scheme overlap with the 1st and another color scheme overlap with the 3rd nominal domain in Fig. 5b. The 2nd nominal domain serves as a bridge between the other two nominal domains. Out of domain test accuracy is reported in Table 5.

Table 5: **Out of Domain Test Accuracy for Sequential Color-Mnist (Red Diverging Palette) from Figure 5**. Each sub-domain in Figure 5 contains a random sample of 1000 mnist images. Random seed is shared for the different sub-domains of a nominal domain but different across nominal domains. Each repetition is with different starting random seed, 10 repetitions are done. The sub-domains are combined to form one nominal domain and 50 percent is used for training, the rest for validation. Comparison algorithms are DIVA (Ilse et al., 2019) and Match-DG (Mahajan et al., 2020), while Deep-All by pooling all training domains together is used as baseline. The 2nd domain is a bridge domain that connect the 1st and 3rd domain, so it is not used as test domain at all.

Color-Mnist (Figure 5)	**Test Domain 1**	**Test Domain 3**
DIVA	0.53 ± 0.05	0.63 ± 0.05
HDUVA	$\mathbf{0.56} \pm 0.05$	$\mathbf{0.68} \pm 0.05$
Deep-All	0.53 ± 0.06	0.61 ± 0.06
Match-DG	0.44 ± 0.04	0.67 ± 0.10

E.3 Domain Embedding

Here, we investigate the ability of our approach to generate meaningful domain embeddings. To this end, we adapt the standard rotated MNIST benchmark (Ilse et al., 2019) by introducing an overlap

between three nominal domains: for the first nominal domain, we use 1000 samples of MNIST and rotate them by 15, 30 and 45 degrees respectively. Thus, the first domain contains 3000 instances and each rotation angle constitutes one sub-domain. For the second domain nominal domain, we rotate the same subset of MNIST, by 30, 45 and 60 degrees respectively. In this way, each nominal domains has two rotation degrees of overlap corresponding to 2000 instances that have the same rotation. We use these 2 nominal domains for training, and simulate a sequential domain shift for testing with rotation angles of 0, 22 and 75 degrees. We sampled images from both nominal training domains as well the continuously shifted test domain and plot their topic distributions in Figure 6. We expect the topics of the training domains to overlap substantially, due to the shared rotation angles. We further expect for the topics of the test domain to span the entire range of topics from both training domains. Figure 6 illustrates that HDUVA indeed assigns similar domain topics to many instances from both training domains, while samples from the test domain span the entire range of topics.

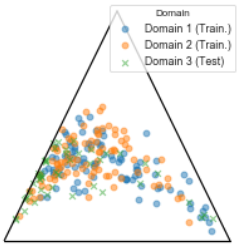


Figure 6: Topic plot of overlapped domains

F Model Ablation

Here we investigate what role z_x is playing for the performance. Without z_x, our model is shown in Figure 7. The performance is shown in Table 2 which shows the method performs equally well with the default version of HDUVA. However, we keep z_x to be default in the HDUVAsince z_x can be used to represent unobserved domain information.

G Alternative Inference Method for HDUVA

We propose an alternative inference algorithm for our model. The graphical model for the Ladder-VAE version of our model is shown in Figure 8 which we coined LHDUVA. The corresponding variational posterior and ELBO is explained below. We summarize this alternative algorithm in Algorithm 2.

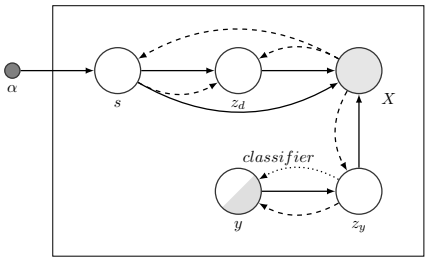


Figure 7: HDUVA without variable z_x

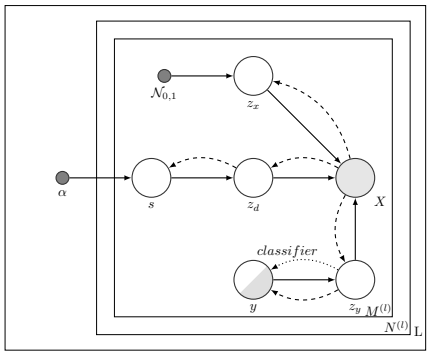


Figure 8: LHDUVA: Ladder Hierarchical Domain Unsupervised Variational Auto-encoding

G.1 Inference for LHDUVA

In Figure 8, we factorize the approximate posterior as follows:

$$\begin{aligned} &q_\phi(s^{(l,i)}, z_d^{(l,i)}, z_x^{(l,i)}, z_y^{(l,i)}|x^{(l,i)}) \\ =&q_\phi(s|z_d^{(l,i)})q_\phi(z_d^{(l,i)}, z_x^{(l,i)}, z_y^{(l,i)}|x^{(l,i)}) \end{aligned} \tag{9}$$

For the approximate posterior distributions of z_x, z_d and z_y, we follow Ilse et al. (2019) and assume fully factorized Gaussians with parameters given as a function of their input:

$$\begin{aligned} &q_\phi(z_d^{(l,i)}, z_x^{(l,i)}, z_y^{(l,i)}|x^{(l,i)}) \\ =&q_{\phi_d}(z_d^{(l,i)}|x^{(l,i)})q_{\phi_x}(z_x^{(l,i)}|x^{(l,i)})q_{\phi_y}(z_y^{(l,i)}|x^{(l,i)}) \end{aligned} \tag{10}$$

Encoders q_{ϕ_y}, q_{ϕ_d} and q_{ϕ_x} are parameterized by ϕ_y, ϕ_d and ϕ_x using separate neural networks to model respective means and variances as function of x.

For the form of the approximate posterior distribution of the topic s we chose a Dirichlet distribution:

$$q_{\phi_s}(s^{(l,i)}|z_{d_j}^{(l,i)}) = Dir\left(s^{(l,i)}|\phi_s(z_{d_j}^{(l,i)})\right) \tag{11}$$

where ϕ_s parameterizes the concentration parameter based on z_d, using a neural network.

G.2 ELBO for LHDUVA

Given the priors and factorization described above, we can optimize the model parameters by maximizing the evidence lower bound (ELBO). We can write the ELBO for a given input-output tuple (x, y) as:

$$\begin{aligned} &ELBO(x,y) = E_{q(z_d|x),q(z_x|x),q(z_y|x)} \log p_\theta(x|z_d, z_x, z_y) \\ &- \beta_x KL(q_{\phi_x}(z_x|x)||p_{\theta_x}(z_x)) - \beta_y KL(q_{\phi_y}(z_y|x)||p_{\theta_y} \\ &(z_y|y)) - \beta_d E_{q_{\phi_s}(s|x,z_d),q_{\phi_d}(z_d|x)} \log \frac{q_{\phi_d}(z_d|x)}{p_{\theta_d}(z_d|s)} \\ &- \beta_s E_{q_{\phi_d}(z_d|x)} KL(q_{\phi_s}(s|z_d)||p_{\theta_s}(s|\alpha)) \end{aligned} \tag{12}$$

where we use β to represent the multiplier in the Beta-VAE setting (Higgins et al., 2016), further encouraging disentanglement of the latent representations.

We add an auxiliary classifier $q_\omega(y|z)$, which is parameterized by ω, to encourage separation of classes y in z_y. The LHDUVA objective then becomes:

$$\mathcal{F}(x,y) = ELBO(x,y) + \gamma_y E_{q_{\phi_y}(z_y|x)}[\log q_\omega(y|z_y)] \tag{13}$$

To efficiently perform inference with the dependent stochastic variables z_d and s, we follow Sønderby et al. (2016) and adapt the ELBO using the Ladder VAE approach as detailed in the next section.

G.2.1 Dealing with Dependent Stochastic Variables

The joint posterior $q(z_d, s|x)$ can be written as:

$$q(z_d, s|x) = \frac{q(z_d, s, x)}{q(x)} = \frac{q(z_d, s, x)}{q(z_d, x)} \frac{q(z_d, x)}{q(x)}$$
$$= q(s|z_d, x)q(z_d|x) = q(s|z_d)q(z_d|x) \tag{14}$$

where conditional independence of s from x is assumed. As pointed out by Chen et al. (2016); Tomczak & Welling (2018), this can lead to inactive stochastic units. We follow Sønderby et al. (2016) and recursively correct the generative distribution by a data dependent approximate likelihood. Additionally, we implement a deterministic warm-up period of β following Sønderby et al. (2016); Ilse et al. (2019), in order to prevent the posterior of the latent representation from aligning too quickly to its prior distribution.

H Weak Supervision on domains

In many scenarios only incomplete domain information is available. For example, due to privacy concerns, data from from different customers within a region may be pooled so that information on the nominal domain at customer-level is lost and only higher-level domain information is available. In other settings, substantial heterogeneity may exist in a domain and various unobserved sub-domains may be present. We introduce two techniques for weak supervision on domains, allowing the model to infer such lower-level domains or sub-domain information in the form of a topic s.

H.1 Topic Distribution Aggregation

To indicate that a group of samples "weakly" belong to one domain, we aggregate the concentration parameter of the posterior distribution of s for all samples in a minibatch (note that all samples in a minibatch have the same nominal domain):

$$\phi_s^{agg}(z_{d_{1:M}}^{(l,i)}) = 1/M \sum_{j=1:M} \left(\phi_s(z_{d_j}^{(l,i)})\right) \tag{15}$$

We then use the aggregated concentration parameter to sample a topic from a Dirichlet distribution:

$$q^{agg}(s^{(l,i)}|z_{d_{1:M}}^{(l,i)}) = Dir\left(\cdot|\phi_s^{agg}(z_{d_{1:M}}^{(l,i)})\right) \tag{16}$$

The conditional prior of $z_d^{(l,i)}$ (equation 2) then shares this same topic for all samples in the ith mini-batch. We interpret this topic-sharing across samples in a mini-batch as a form of regularized weak supervision. In one-hot encoded approaches, all samples from the same nominal domain would share the same topic. In contrast, sharing a topic in the conditional prior of the latent representation across samples in a mini-batch provides a weak supervision, whilst allowing for an efficient optimisation via SGD. Note that concentration parameters for a mini-batch are only aggregated during training, at test time sample-specific posterior concentration parameters are used.

H.2 Weak domain distribution supervision with MMD

DIVA encourages separation of nominal domains in the latent space z_d by fitting an explicit domain classifier which might limit model performance in the case of incomplete domain information. To mitigate these limitations but still weakly encourage separation between different nominal domains, we constrain the HDUVA objective based on the Maximum-Mean-Discrepancy (MMD) (Gretton et al., 2012) between pairwise domains.

Denoting C_{mmd}^d as the minimal distance computed by MMD as an inequality constraint, we can write the constraint optimization of equation 13 as follows:

$$\underset{\theta,\phi,\omega}{argmax} \sum_{l,i} \mathcal{F}(x^{(l,i)}, y^{(l,i)})$$
$$s.t.\ MMD(q_{z_d}^{(l,i)}(\cdot)|q_{z_d}^{(l',i)}(\cdot)) \geq C_{mmd}^{(l,l')} \tag{17}$$

H.3 PRACTICAL CONSIDERATIONS

In practice, we transform the constrained optimization in Equation 17 with a Langrange Multiplier. This leads to the final loss in Equation 18, where $\gamma_d^{(l)}$ denotes the Lagrange multiplier for C_{mmd}^d (c.f. Equation 17):

$$\begin{aligned}\mathcal{L} = &\sum_{l,i} -\mathcal{F}^{(agg,ladder)}(x^{(l,i)}, y^{(l,i)}) \\ &- \gamma_d^{(l)} \sum_{i,l,l'} MMD(q_{z_d}^{(l,i)}(\cdot)|q_{z_d}^{(l',i)(\cdot)})\end{aligned} \tag{18}$$

Superscript agg and $ladder$ in Equation 18 refer to batch-wise aggregation of the concentration parameter and the ladder approach described above.

Algorithm 2 LHDUVA

while not converged or maximum epochs not reached **do**
  warm up β defined in Equation 12, as in (Sønderby et al., 2016)
  fetch mini-batch {x, y} ={$x^{(l,i)}, y^{(l,i)}$}
  compute parameters for $q_{\phi_x}(z_x|x)$, $q_{\phi_y}(z_y|x)$, $q_{\phi_d}(z_d|x)$
  sample latent variable z_x^q, z_y^q and compute $[\log q_\omega(y|z_y)]$ in equation 13
  sample z_d^q, infer concentration parameter $\phi_s(z_d)$ and aggregate according to Equation 15
  sample topic s from aggregated $\phi_s^{agg}(z_{d_{1:M}})$ according to Equation 16.
  compute prior distribution for z_d using s
  adapt posterior of $q_{\phi_d}(z_d)$ with ladder-vae method (Sønderby et al., 2016)
  sample z_d^q from adapted $q_{\phi_d}(z_d)$
  compute $p_\theta(x|z_x, z_y, z_d)$ using sampled z_x^q, z_y^q, z_d^q
  compute KL divergence for z_d, z_x and z_y, s in Equation 12
  compute pair wise MMD of the nominal domains
  aggregate loss according to 18 and update model
end while

I ARCHITECTURES

We list the neural network architectures used in the experiment. For the decoder, we use the same architecture as explained in Table 6. For image of size 224, we used Alexnet with the last layer removed as the encoder.

Table 6: Decoder Architecture
We use the same decoder architecture across all experiments. First, from latent code, we use a Gated Dense Layer Tomczak & Welling (2018) to map the latent code to the dimension of the image. Then we apply two times Gated ConvolutionVan Den Oord et al. (2016)(kernel size =3, stride=1, padding=1, dilation=1)).

	Block	Input	Output
1	GatedDense	dim(z)	3 × height × image
2	GatedConv2d	3	64
3	GatedConv2d	64	64

For the synthetic datasets as well as the malaria dataset, we use the architecture as described in Table 7.

J OTHER EXPERIMENT DETAILS

For comparing algorithms, we implemented DIVA (Ilse et al., 2019) and MatchDG (Mahajan et al., 2020), and use the same hyper-parameters suggested by the original paper. For HDUVA, we match

Table 7: Archiecture of the encoder used for small images with convolution kernel size = 5, convolution stride size = 1, max pool stride size =2.

	Block	Input	Output
1	Conv2d	3	32
2	BatchNorm2d	32	32
3	Relu		
4	MaxPool2d(2)		
4	Conv2d	32	64
5	BatchNorm2d	64	64
6	ReLu		
7	MaxPool2d(2)		

the hyper-parameters in (Ilse et al., 2019), where we take the latent dimension for each latent code is taken to be 64, i.e. $z_x = z_y = z_d = 64$. The classifier is taken to be a one layer neural network with Relu activation. For all experiments, γ_y in equation 8 is taken to be $1e5$, while the β values are taken to be 1, warm-up of KL divergence loss in Equation 7 is taken to be 100 epochs. We use topic dimension of 3 for HDUVA. For the malaria experiment, we run with maximum 1000 epochs, with early stopping tolerance of 100 epochs. For HDUVA, we use ELBO directly as model selection criteria, for the rest of the algorithms, we use validation accuracy as model selection criteria. That means, we do not use the validation set at all.

The mnist related experiments are run with maximum 500 epochs with early stopping tolerance of 100 epochs. For HDUVA, we use ELBO directly as model selection criteria, for the rest of the algorithms, we use validation accuracy as model selection criteria. That means, we do not use the validation set at all. We use a learning rate of 1e-4 for DIVA and HDUVA, a learning rate of 1e-5 (better than 1e-4) for Deep-All and the suggested learning rate for MatchDG. For experiments regarding MNIST, including MNIST rotation overlap E.3, colored mnist combination E.1, and domain overlapped color-mnist in E.2, we use random sub-samples (each contains 1000 instances) pre-sampled from `https://github.com/AMLab-Amsterdam/DIVA/tree/master/paper_experiments/rotated_mnist/dataset` with commit hash tag `ab590b4c95b5f667e7b5a7730a797356d124`.

For the PACS experiment, we run with maximum 500 epochs, with early stopping criteria of 5 epochs to save computation resources. For HDUVA, we use ELBO directly as model selection criteria, for the rest of the algorithms, we use validation accuracy as model selection criteria. That means, we do not use the validation set at all. We use a learning rate of 1e-5 for HDUVA, DIVA, Deep-All and use default learning rate of MatchDG.