Optimal Representations for Covariate Shifts

Anonymous Author(s)
Affiliation
Address
email

Abstract

Machine learning often experiences distribution shifts between training and testing. We introduce a simple objective whose optima are *exactly all* representations on which risk minimizers are guaranteed to be robust to Bayes-preserving shifts, e.g., covariate shifts. Our objective has two components. First, a representation must remain discriminative, i.e., some predictor must be able to minimize the source and target risk. Second, the representation's support should be invariant across source and target. We make this practical by designing self-supervised methods that only use unlabelled data and augmentations. Our objectives achieve SOTA on DomainBed, and give insights into the robustness of recent methods, e.g., CLIP.

1 Introduction

It is hard to build machine learning (ML) systems that are robust to distribution shifts between a source (train) and target (test) domain. One promising approach to DG is to learn representations from which predictors trained on source must perform well on target. In practice, however, no current DG method uniformly outperforms empirical source-risk minimizers (ERM) [22]. Our theoretical understanding is also lacking: while previous work has studied properties that are or are not sufficient for robust representations [7, 58, 27], the *minimal* set of requirements is not yet known.

We introduce the first, simple, objective whose optima are exactly the set of all representations on which source risk minimizers are guaranteed to generalize across distribution shifts that preserve the Bayes predictor. Our characterization implies that it is sufficient and *necessary* that an optimal representation: (a) remains discriminative, i.e., there must exist predictors that simultaneously minimize *both* source and target risk; and (b) keeps the support of its marginal distribution shift-invariant.

Optimal representations must thus seek discriminative information of targets. Even worse, we prove that without target knowledge, no representation can uniformly outperform *constant* representations, which may explain why DG methods struggle to outperform ERM. We show how to overcome these challenges using only a large set of unlabeled examples and particular data augmentations that retain discriminative information but minimal domain-specific information. Text descriptions of images are such augmentations, as they are informative for many classification tasks, but remove domain-specific information. With those augmentations, we design practical self-supervised (SSL) objectives for learning robust representations. Our objectives give insights into CLIP's robustness [43], and lead to improved CLIP-based representations that achieve SOTA on DomainBed [22].

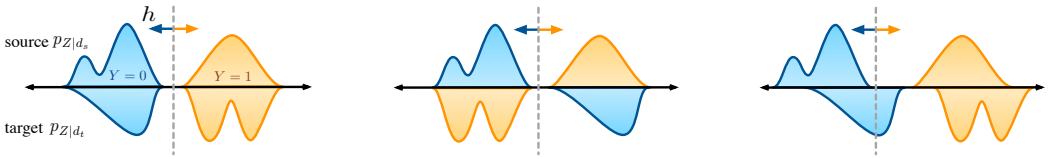


Figure 1: (Left) IDG optimal representations must have invariant supports while being simultaneously discriminative on all domains: (center) without discriminativeness, a source-risk minimizer can mispredict the target, and (right) without support match, some risk minimizer will perform poorly.

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

2 Problem statement: idealized domain generalization (IDG)

We want to learn representations Z of inputs X that are robust across distribution shifts. Specifically, we want an encoder $p_{Z \mid X}$ that ensures that predictors h from representations Z to labels Y, which are trained on a source $D_s \in \mathcal{D}$ distribution $p_{X,Y \mid D_s}$, will perform well on a target $D_t \in \mathcal{D}$ distributions $p_{X,Y \mid D_t}$. We evaluate h's performance using expected target risk $\mathrm{R}_h^{D_t}[Y \mid Z] := \mathbb{E}_{p_{X,Y \mid D_t}}[\ell(Y, f(Z))]$ with respect to a well behaved loss function ℓ, e.g., log-loss, MSE, or 0-1 loss.

To separate domain generalization from finite sample generalization, we consider an idealized DG (IDG), where predictors h are selected by minimizing the source *population* rather than empirical risk. To give uniform guarantees while reflecting uncertainty over source-target (D_s, D_t) pairs, we score Z using the expected risk of the worst source minimizer $h \in \mathcal{H}^*_{D_s} := \arg\min_h \mathrm{R}_h^{D_s}[Y \mid Z]$.

Definition. The *idealized domain generalization risk (IDG risk)* of an encoder $p_{Z \mid X}$ is the expected (over domains) worst-case (over source risk minimizers) target risk, i.e.,

$$\mathrm{R}_{\mathrm{IDG}}[Y \mid Z] := \mathbb{E}_{p_{D_s,D_t}}\left[\sup_{h \in \mathcal{H}^*_{D_s}} \mathrm{R}_h^{D_t}[Y \mid Z]\right] \tag{1}$$

A representation Z^* is *optimal for IDG* if it minimizes Eq. (1): $p_{Z^* \mid X} \in \arg\min_{p_{Z \mid X}} \mathrm{R}_{\mathrm{IDG}}[Y \mid Z]$.

Interesting DG is clearly only possible when target and source domains are related. We assume that domains are related by the following generalized covariate shift (GCS), which says that the inputs' Bayes predictor $f^* = \arg\min_f \mathbb{E}_{p_{D_t}}[\mathrm{R}_f^{D_t}[Y \mid X]]$ is uniquely optimal on all domains.

Assumption (Generalized covariate shift)**.** Domain risk minimizers $f^d \in \arg\min_f [\mathrm{R}_f^d[Y \mid X]]$ are equal to the Bayes predictor on their support: $f^d(x) = f^*(x)$ for all $d, x \in \mathrm{supp}(p_{D,X})$.

For log-loss ℓ GCS recovers standard covariate shift, i.e., $p_{Y \mid x,d} = p_{Y \mid x}$. For other losses, GCS is weaker, e.g., it only requires invariance of most likely labels for 0-1 loss, and of conditional expectations for MSE. For minor assumptions, formal statements and proofs see Appcs. A and B.

3 Characterizing optimal representations for IDG under covariate shift

IDG risk is useful to evaluate representations but gives few insights into IDG and is impractical to optimize due to the sup. in Eq. (1). Assuming GCS we can provide a simplified, equivalent objective that is easier to optimize. The intuition is that under GCS any source risk minimizer will also make optimal predictions on all target samples x that are in source domain's support. Thus, optimal representations for IDG are exactly those that (a) ensure that all domains have the same support in $\mathcal{Z}$, and (b) retain GCS from Z without sacrificing the ability to predict Y optimally. See Fig. 1.

Theorem 1. *Under our assumptions, an encoder $p_{Z^* \mid X}$ is optimal for IDG if and only if it minimizes the risk $\mathrm{R}[Y \mid Z] := \inf_h \mathbb{E}_{p_{D_t}}\left[\mathrm{R}_h^{D_t}[Y \mid Z]\right]$ while matching the support of Z across domains, i.e.,*

$$p_{Z^* \mid X} \in \arg\min_{p_{Z \mid X}} \quad \mathrm{R}[Y \mid Z] \qquad \text{s.t.} \qquad \forall\, d \in \mathcal{D},\ \mathrm{supp}(p_{Z \mid d}) = \mathrm{supp}(p_Z) \tag{2}$$

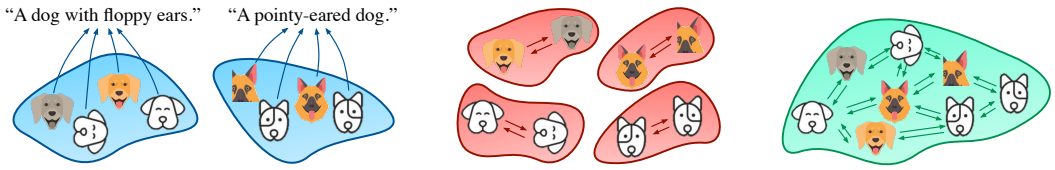
Moreover, such encoders exist and their IDG risk is the Bayes risk $\mathrm{R}_{\mathrm{IDG}}[Y \mid Z^*] = \mathrm{R}[Y \mid X]$.

Theorem 1 provides an objective to learn representations on which performing risk minimization using a single domain is as good as performing risk minimization on all domains simultaneously. Other sufficient objectives have previously been proven or hinted towards [7, 58, 13, 27], e.g., minimizing the risk while matching the representation's marginal. To our knowledge, Thm. 1 is nevertheless the first to identify the necessary and sufficient conditions for optimal representations. This gives better insights into IDG and provides a framework for deriving all objectives that describe optimal IDG.

Theorem 1 shows that one must know the target domains to learn optimal representations for IDG. Access to target domain might seem unrealistic, but without it, it is provably impossible to learn useful representations for IDG. Specifically, the following holds under minor assumptions (see Appx. B.3).

Proposition 1 (No free lunch for IDG)**.** *Let Z_{d_s} be any representation chosen on source d_s. For every "good" target domain outside the source's support on which Z_{d_s} outperforms a constant representation $C \in \mathcal{Z}$, there are many "bad" target domains on which Z_{d_s} is* strictly worse *than C.*

Proposition 1 may explain why previous DG methods fail to outperform ERM [22]: the knowledge they have access to is insufficient. So either you access target domain d_t and achieve an IDG risk that matches supervised learning (Thm. 1), or you do not and cannot do better than a constant (Prop. 1).



(a) image-text augmentations (b) standard augmentations (c) supervised augmentations

Figure 2: (a) Image-text aug. is (nearly) domain-covering by mapping images across domains to similar descriptions. (b) Standard aug. is not domain-covering. (c) Supervised aug. uniformly augments inputs inside their label class irrespective of domains. Arrows denote augmentations. Bubbles denote inputs that have the same representations by predicting the augmentations.

4 Learning optimal representations with domain-covering augmentations

Thm. 1 requires labels from all domains, which is impractical. We overcome this with self-supervised learning (SSL) and particular data augmentations $p_{A \mid X}$ from inputs X to augmentations A. The key requirement for A is to retain discriminative information about labels Y, i.e., if samples $x, x' \in \mathcal{X}$ have same augmentations $p_{A \mid x} = p_{A \mid x'}$, then they have the same Bayes predictions $f^*(x) = f^*(x')$. With such Bayes-preserving A, one can minimize $\mathrm{R}[Y \mid Z]$ by instead maximizing mutual information $\mathrm{I}[A; Z]$ [15]. However, fully optimizing $\mathrm{I}[A; Z]$ is not generally possible under the support constraint in Eq. (2). This can be addressed by a *domain-covering* assumption: if for each $d \in \mathcal{D}$, there is an input that is mapped to every augmentation distribution, i.e., $\{p_{A \mid x} \mid x \in \mathrm{supp}(p_{X \mid d})\} = \{p_{A \mid x} \mid x \in \mathcal{X}\}$.

Proposition 2. *Let $p_{A \mid X}$ be a domain-covering augmenter. Then any optimal solution $p_{Z^* \mid X}$ of the following objective is optimal for IDG:*

$$p_{Z^* \mid X} \in \operatorname*{arg\,max}_{p_{Z \mid X}} \mathrm{I}[A; Z] \quad \text{s.t.} \quad \forall\, d \in \mathcal{D},\ \mathrm{supp}(p_{Z \mid d}) = \mathrm{supp}(p_Z) \tag{3}$$

Proposition 2 shows that we can still learn optimal representations for IDG without labels if we use right augmentations. But how realistic are those augmentations? Standard image augmentations like cropping and color jittering are generally Bayes-preserving, but not domain-covering for typical domains (e.g. sketches and photos), since outputs A are highly correlated with the domain D of the original input X, as seen in Fig. 2b. A practical choice of A that is nearly domain-covering, is a mapping from images to text descriptions, as with CLIP [43]. Image-text augmentations have many advantages: they (i) preserve label information for many downstream tasks; (ii) are close to being domain-covering, as images from different domains but similar semantics are often mapped to similar descriptions (Fig. 2a); (iii) are easy to access in practice given their abundance on the internet. This may explain the incredible robustness of CLIP compared to other SSL methods [11, 24, 21].

We now design practical objectives for optimizing Eq. (3) by using a Lagrangian relaxation and introducing a *domain bottleneck* $\mathrm{B}[Z, D]$ that enforces support match. Specifically, we convert Eq. (3) to a unconstrained objective $\arg\min_{p_{Z \mid X}} \mathrm{H}[A \mid Z] + \lambda\, \mathrm{B}[Z, D]$. where $\mathrm{H}[A \mid Z]$ replaces $\mathrm{I}[A; Z]$ as $\mathrm{H}[A]$ is a constant w.r.t. $p_{Z \mid X}$. A valid $\mathrm{B}[Z, D]$ ensures that minimizing $\mathrm{B}[Z, D]$ while maximizing $\mathrm{I}[A; Z]$ enforces the support constraint which we will introduce later. We discuss variational bounds for the objective that can be efficiently estimated from samples and optimized with SGD [9]. For simplicity, we use a deterministic encoder $e_\varphi : \mathcal{X} \to \mathcal{Z}$. Detailed derivations are in Appx. C.

For the first term, we use an upper bound on $\mathrm{H}[A \mid Z] \leq \mathbb{E}_{p_{A,Z}}[-\log q(A \mid Z)]$, where q is the contrastive variational distribution as InfoNCE [40] that is standard in SSL. Specifically, for a sample X, we obtain a collection $\mathbf{A} := \{A^+, A_1^-, \ldots, A_n^-\}$ of one positive augmentation A^+ sampled from $p_{A \mid X}$ and n negatives A_i^- sampled from p_A. InfoNCE then constructs $q(A \mid Z)$ using a critic s_ψ to score how likely each $A' \in \mathbf{A}$ is to be positive, see Line 6 of Algorithm 1.

Algorithm 1 CAD objective

Require: $e_\varphi, s_\psi, D, X, n$
1: $Z \leftarrow e_\varphi(X)$
2: $A^+ \leftarrow \mathrm{sample}(p_{A \mid X})$
3: $\{(D_i^-, X_i^-, A_i^-)\}_{i=1}^n \xleftarrow{\text{i.i.d.}} \mathrm{sample}(p_{D,X,A})$
4: $\mathbf{X}, \mathbf{A} \leftarrow \{X\} \cup \{X_i^-\}_{i=1}^n, \{A^+\} \cup \{A_i^-\}_{i=1}^n$
5: $\mathbf{X}_D \leftarrow \{X\} \cup \{X_i^- \mid D_i^- = D, i \in [n]\}$
6: $\mathcal{L}_{\text{aug}} \leftarrow -\log \frac{\exp s_\psi(A^+, Z)}{\sum_{A' \in \mathbf{A}} \exp s_\psi(A', Z)}$ ▷ $\mathrm{H}[A \mid Z]$
7: $\mathcal{L}_{\text{supp}} \leftarrow \log \frac{\sum_{X' \in \mathbf{X}_D} \exp e_\varphi(X')^T Z}{\sum_{X'' \in \mathbf{X}} \exp e_\varphi(X'')^T Z}$ ▷ $\mathrm{I}[Z; D]$
8: **return** $\mathcal{L}_{\text{CAD}} = \mathcal{L}_{\text{aug}} + \lambda \mathcal{L}_{\text{supp}}$

For the second term $\mathrm{B}[Z, D]$, we introduce a novel contrastive adversarial domain (CAD) bottleneck and discuss more choices in Appx. C. Our CAD bottleneck aims to minimize $\mathrm{I}[Z; D]$, which enforces support match using a KL divergence. Dropping constants w.r.t. Z we thus aim to maximize $\mathrm{H}[D \mid Z]$. Domain-adversarial neural network [DANN, 17] does so by ensuring that a domain classifier q_ϕ cannot predict D from Z, i.e., it max-

Table 1: As suggested by our theory, it is important to (i) enforce support match with bottlenecks; (ii) use domain-covering augmentations; (iii) get access to target domain data for IDG.

Setup		Log likelihood
$\mathrm{R}\,[Y \mid Z]$	Base	-5.1 ± 0.3
	CAD	-0.7 ± 0.1
$\mathrm{H}[A \mid Z]$	CAD	-0.8 ± 0.2
	with Std. Aug.	-7.5 ± 0.2
	with only Src.	-4.2 ± 0.2

Table 2: Finetuning CLIP with our CAD bottleneck achieves SOTA on DomainBed. Avg. acc. over PACS, OfficeHome, VLCS, DomainNet.

Algorithm	Avg. target acc.
ERM	68.0 ± 0.3
DomainBed SOTA	69.3 ± 0.2
CLIP S	72.4 ± 0.2
CLIP S + Base	72.5 ± 0.3
CLIP S + CAD	73.8 ± 0.3
CLIP L	76.8 ± 0.4
CLIP L + CAD	$\mathbf{77.6 \pm 0.4}$

imizes $\mathbb{E}_{p_{D,Z}}[-\log q_\phi(D \mid Z)] \geq \mathrm{H}[D \mid Z]$ w.r.t. encoder φ but minimizes it w.r.t. ϕ. However, it maximizes an *upper* bound on the desired term and suffers from unstable adversarial training.

To overcome these issues, we construct $q(D \mid Z)$ without introducing additional parameters and with a bound that is tight with enough samples. We include the full derivation in Appx. C.3 and briefly present the algorithm here. Given a sample X with domain D and n tuples $\{(D_i^-, X_i^-)\}_{i=1}^n$ i.i.d. sampled from $p_{D,X}$, we first obtain a variational distribution $q(X \mid Z)$ as InfoNCE using a non-parametric critic $e_\varphi(X)^T Z$ tied with e_φ. Then we collect those inputs $\mathbf{X}_D$ associated with the current domain D, and sum $q(X' \mid Z)$ over $X' \in \mathbf{X}_D$ to compute $q(D \mid Z)$. See Algorithm 1 for details. In Appx. C.4, we derive a conditional variation of CAD that minimizes $\mathrm{I}[Z; D \mid Y]$, which can be used when labels Y are available. In Appx. C.2, we also introduce an entropy bottleneck that minimizes $\mathrm{H}[Z]$, which does not require domain labels D that are rarely accessible in SSL.

5 Experiments

We aim to empirically verify our theoretical results and investigate our proposed SSL objectives in practical DG. See Appcs. E and F for experimental details and additional results.

Optimal representations for worst-case IDG We validated our theory in an *idealized* DG setup on PACS [34]. Specifically, we (i) trained encoders on *all* PACS data, i.e., all domains, labels, and both train/test data; (ii) set a source D_s and target D_t; (iii) selected *worst-case* predictors h by minimizing source risk but maximizing target risk (see [14]); (iv) repeated the last 2 steps over all source-target pairs and averaged each h's negative target risk (log likelihood). Results are in Table 1.

How important is support match? Representations trained with CAD (2nd row) significantly outperform those trained without bottleneck (Base, 1st row) thus supporting Thm. 1. *Can we learn optimal representations without labels?* For domain-covering augmentations, minimizing the SSL $\mathrm{H}[A \mid Z]$ (3rd row) performs similarly to the supervised $\mathrm{R}\,[Y \mid Z]$ thus supporting Prop. 2. *Are standard augmentations sufficient?* Representations trained with CAD and standard augmentations perform poorly (4th row). This shows the importance of domain-covering augmentations. *How important is target knowledge?* Excluding the target domain from the encoder's training significantly decreases performance (5th row compared to 3rd row), which supports Prop. 1.

Approximating optimal representations with pretrained SSL In practice, we can learn IDG optimal representations by performing SSL using a large source of unlabelled inputs X and domain-covering augmentations A. This is nearly how CLIP was pretrained (SSL with 400M image-text pairs) except it did not include a domain bottleneck. Here, we exploit CLIP to learn robust representations, by: (i) freezing the pretrained CLIP and adding an MLP on top of it; (ii) training the MLP with our CAD bottleneck and $\mathrm{R}\,[Y \mid Z]$ on available data. We used the DomainBed benchmark and its protocol [22]. Due to space limit, we only report average target accuracy over PACS [34], VLCS [16], OfficeHome [54], and DomainNet [41] datasets and include as baselines 'ERM' and 'DomainBed SOTA', which is the best baseline on *each* dataset. Results are in Table 2.

Can we approx. optimal representations using CLIP? Fine-tuning a large (ViT-B/32) CLIP with our CAD achieves SOTA on DomainBed (7th vs 2nd row). In particular, CLIP L's performance on PACS (94.7%) is close to the 96.7% estimated performance for optimal representations (see Appx. F.2). *Are gains due to architectural differences?* DomainBed's baseline use smaller ResNet50 models. Finetuning a smaller ResNet50 (CLIP S) still outperforms baselines (5th vs 2nd row). Our theory does not constrain the encoder and so we expect larger encoders to be better (6th vs 5th row). *How important is our bottleneck?* Finetuning CLIP S with our CAD bottleneck outperforms finetuning without bottlenecks and CLIP S without finetuning (5th vs 4th and 3rd row).

References

[1] Alexander A Alemi, Ian Fischer, Joshua V Dillon, and Kevin Murphy. Deep variational information bottleneck. *arXiv preprint arXiv:1612.00410*, 2016.

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

[3] Johannes Ballé, Valero Laparra, and Eero P Simoncelli. End-to-end optimized image compression. *arXiv preprint arXiv:1611.01704*, 2016.

[4] Johannes Ballé, David Minnen, Saurabh Singh, Sung Jin Hwang, and Nick Johnston. Variational image compression with a scale hyperprior. *arXiv preprint arXiv:1802.01436*, 2018.

[5] Andrei Barbu, David Mayo, Julian Alverio, William Luo, Christopher Wang, Danny Gutfreund, Joshua Tenenbaum, and Boris Katz. Objectnet: A large-scale bias-controlled dataset for pushing the limits of object recognition models. 2019.

[6] Sara Beery, Grant Van Horn, and Pietro Perona. Recognition in terra incognita. In *Proceedings of the European conference on computer vision (ECCV)*, pp. 456–473, 2018.

[7] Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. A theory of learning from different domains. *Machine Learning*, 79 (1):151–175, 2010.

[8] Shai Ben-David, Tyler Lu, Teresa Luu, and David Pal. Impossibility theorems for domain adaptation. In Yee Whye Teh and Mike Titterington (eds.), *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, volume 9 of *Proceedings of Machine Learning Research*, pp. 129–136, Chia Laguna Resort, Sardinia, Italy, 13–15 May 2010. PMLR. URL `https://proceedings.mlr.press/v9/david10a.html`.

[9] Léon Bottou. Large-scale machine learning with stochastic gradient descent. In *Proceedings of COMPSTAT'2010*, pp. 177–186. Springer, 2010.

[10] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. *arXiv preprint arXiv:2104.14294*, 2021.

[11] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pp. 1597–1607. PMLR, 2020.

[12] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. Ieee, 2009.

[13] Remi Tachet des Combes, Han Zhao, Yu-Xiang Wang, and Geoffrey J. Gordon. Domain adaptation with conditional distribution matching and generalized label shift. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin (eds.), *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. URL `https://proceedings.neurips.cc/paper/2020/hash/dfbfa7ddcfffeb581f50edcf9a0204bb-Abstract.html`.

[14] Yann Dubois, Douwe Kiela, David J Schwab, and Ramakrishna Vedantam. Learning optimal representations with the decodable information bottleneck. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 18674–18690. Curran Associates, Inc., 2020. URL `https://proceedings.neurips.cc/paper/2020/file/d8ea5f53c1b1eb087ac2e356253395d8-Paper.pdf`.

[15] Yann Dubois, Benjamin Bloem-Reddy, Karen Ullrich, and Chris J. Maddison. Lossy compression for lossless prediction. *arXiv preprint arXiv:2106.10800*, 2021.

[16] Chen Fang, Ye Xu, and Daniel N Rockmore. Unbiased metric learning: On the utilization of multiple datasets and web images for softening bias. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1657–1664, 2013.

[17] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *The journal of machine learning research*, 17(1):2096–2030, 2016.

[18] Tilmann Gneiting and Adrian E Raftery. Strictly proper scoring rules, prediction, and estimation. *Journal of the American statistical Association*, 102(477):359–378, 2007.

[19] Mingming Gong, Kun Zhang, Tongliang Liu, Dacheng Tao, Clark Glymour, and Bernhard Schölkopf. Domain adaptation with conditional transferable components. In *International conference on machine learning*, pp. 2839–2848. PMLR, 2016.

[20] Ian Goodfellow. Nips 2016 tutorial: Generative adversarial networks. *arXiv preprint arXiv:1701.00160*, 2016.

[21] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent: A new approach to self-supervised learning. *arXiv preprint arXiv:2006.07733*, 2020.

[22] Ishaan Gulrajani and David Lopez-Paz. In search of lost domain generalization. In *International Conference on Learning Representations*, 2021. URL `https://openreview.net/forum?id=lQdXeXDoWtI`.

[23] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

[24] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9729–9738, 2020.

[25] Dan Hendrycks, Steven Basart, Norman Mu, Saurav Kadavath, Frank Wang, Evan Dorundo, Rahul Desai, Tyler Zhu, Samyak Parajuli, Mike Guo, et al. The many faces of robustness: A critical analysis of out-of-distribution generalization. *arXiv preprint arXiv:2006.16241*, 2020.

[26] Dan Hendrycks, Kevin Zhao, Steven Basart, Jacob Steinhardt, and Dawn Song. Natural adversarial examples. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15262–15271, 2021.

[27] Fredrik D Johansson, David Sontag, and Rajesh Ranganath. Support and invertibility in domain-invariant representations. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pp. 527–536. PMLR, 2019.

[28] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. *arXiv preprint arXiv:2004.11362*, 2020.

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

[30] Naveen Kodali, Jacob Abernethy, James Hays, and Zsolt Kira. On convergence and stability of gans. *arXiv preprint arXiv:1705.07215*, 2017.

[31] David Krueger, Ethan Caballero, Joern-Henrik Jacobsen, Amy Zhang, Jonathan Binas, Dinghuai Zhang, Remi Le Priol, and Aaron Courville. Out-of-distribution generalization via risk extrapolation (rex). In *International Conference on Machine Learning*, pp. 5815–5826. PMLR, 2021.

[32] LAION. Laion-400m open dataset. `https://laion.ai/laion-400-open-dataset`, 2021. Accessed: 2021-09-14.

[33] Bo Li, Yifei Shen, Yezhen Wang, Wenzhen Zhu, Colorado J Reed, Jun Zhang, Dongsheng Li, Kurt Keutzer, and Han Zhao. Invariant information bottleneck for domain generalization. *arXiv preprint arXiv:2106.06333*, 2021.

[34] 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.

[35] Haoliang Li, Sinno Jialin Pan, Shiqi Wang, and Alex C Kot. Domain generalization with adversarial feature learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5400–5409, 2018.

[36] Ya Li, Xinmei Tian, Mingming Gong, Yajing Liu, Tongliang Liu, Kun Zhang, and Dacheng Tao. Deep domain generalization via conditional invariant adversarial networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 624–639, 2018.

[37] Mingsheng Long, Yue Cao, Jianmin Wang, and Michael Jordan. Learning transferable features with deep adaptation networks. In *International conference on machine learning*, pp. 97–105. PMLR, 2015.

[38] Mingsheng Long, Han Zhu, Jianmin Wang, and Michael I Jordan. Deep transfer learning with joint adaptation networks. In *International conference on machine learning*, pp. 2208–2217. PMLR, 2017.

[39] A Tuan Nguyen, Toan Tran, Yarin Gal, Philip HS Torr, and Atılım Güneş Baydin. Kl guided domain adaptation. *arXiv preprint arXiv:2106.07780*, 2021.

[40] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

[41] Xingchao Peng, Qinxun Bai, Xide Xia, Zijun Huang, Kate Saenko, and Bo Wang. Moment matching for multi-source domain adaptation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 1406–1415, 2019.

[42] Ben Poole, Sherjil Ozair, Aaron Van Den Oord, Alex Alemi, and George Tucker. On variational bounds of mutual information. In *International Conference on Machine Learning*, pp. 5171–5180. PMLR, 2019.

[43] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 8748–8763. PMLR, 18–24 Jul 2021. URL `https://proceedings.mlr.press/v139/radford21a.html`.

[44] Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do imagenet classifiers generalize to imagenet? In *International Conference on Machine Learning*, pp. 5389–5400. PMLR, 2019.

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

[46] Vaishaal Shankar, Achal Dave, Rebecca Roelofs, Deva Ramanan, Benjamin Recht, and Ludwig Schmidt. Do image classifiers generalize across time? *arXiv preprint arXiv:1906.02168*, 2019.

[47] Jian Shen, Yanru Qu, Weinan Zhang, and Yong Yu. Wasserstein distance guided representation learning for domain adaptation. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[48] Ravid Shwartz-Ziv and Naftali Tishby. Opening the black box of deep neural networks via information. *CoRR*, abs/1703.00810, 2017. URL `http://arxiv.org/abs/1703.00810`.

[49] Baochen Sun and Kate Saenko. Deep coral: Correlation alignment for deep domain adaptation. In *European conference on computer vision*, pp. 443–450. Springer, 2016.

[50] Remi Tachet des Combes, Han Zhao, Yu-Xiang Wang, and Geoffrey J Gordon. Domain adaptation with conditional distribution matching and generalized label shift. *Advances in Neural Information Processing Systems*, 33, 2020.

[51] Rohan Taori, Achal Dave, Vaishaal Shankar, Nicholas Carlini, Benjamin Recht, and Ludwig Schmidt. Measuring robustness to natural distribution shifts in image classification. *arXiv preprint arXiv:2007.00644*, 2020.

[52] Lucas Theis, Wenzhe Shi, Andrew Cunningham, and Ferenc Huszár. Lossy image compression with compressive autoencoders. *arXiv preprint arXiv:1703.00395*, 2017.

[53] Naftali Tishby, Fernando C Pereira, and William Bialek. The information bottleneck method. *arXiv preprint physics/0004057*, 2000.

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

[55] Haohan Wang, Songwei Ge, Eric P Xing, and Zachary C Lipton. Learning robust global representations by penalizing local predictive power. *arXiv preprint arXiv:1905.13549*, 2019.

[56] Aolin Xu and Maxim Raginsky. Minimum excess risk in bayesian learning. *arXiv preprint arXiv:2012.14868*, 2020.

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

[58] Han Zhao, Remi Tachet Des Combes, Kun Zhang, and Geoffrey Gordon. On learning invariant representations for domain adaptation. In *International Conference on Machine Learning*, pp. 7523–7532. PMLR, 2019.

A Preliminaries

A.1 Notation

For the most part, we will assume that all spaces are discrete probability spaces. A full list of assumptions is found at Appx. A.3.

General The image of a set $A \subseteq \mathcal{X}$ under a function $f : \mathcal{X} \to \mathcal{Y}$ is denoted $f^{\to}(A) = \{f(x) \,|\, x \in A\}$. The pre-image is denoted $f^{\leftarrow}(B) = \{x \in \mathcal{X} \,|\, f(x) \in B\}$ for $B \subseteq \mathcal{Y}$.

Probability Random variables (r.v.) are denoted by uppercase letters (e.g., X), and their sample space and realizations are denoted by the corresponding calligraphic (e.g., $\mathcal{X}$) and lowercase letters (e.g., x) respectively. The probability mass function (pmf) of a random variable X is denoted as p_X. We use capital P instead of p to denote the measure under p. The support $\mathrm{supp}(p_X)$ of a discrete distribution is the set of all points $x \in \mathcal{X}$ with positive probability, i.e., $\mathrm{supp}(p_X) = \{x \in \mathcal{X} \,|\, p_X(x) > 0\}$. The space of all probability distributions on $\mathcal{X}$ is denoted $\mathcal{P}(\mathcal{X}) = \{p_X \,|\, p_X(x) \geq 0 \text{ and } \sum_{x \in \mathcal{X}} p_X(x) = 1\}$.

When it is necessary to be explicit, we will denote 'X is distributed as p_X' using the notation $X \overset{\mathrm{d}}{\sim} p_X$. Expectations are written as: $\mathbb{E}_{p_X}[f(X)]$, independence of two r.v. as $\cdot \perp\!\!\!\perp \cdot$, conditional independence as $\cdot \perp\!\!\!\perp \cdot \,|\, \cdot$.

For jointly distributed random variables (X, Y) taking value in (t.v.i.) $\mathcal{X} \times \mathcal{Y}$, the conditional distribution is denoted as $p_{Y \,|\, X} : \mathcal{Y} \times \mathcal{X} \to [0, 1]$. For convenience, let $p_{Y \,|\, x} = p_{Y \,|\, X}(\,\cdot \,|\, x)$ be the conditional distribution of Y given x. All random variables are independently distributed, unless an explicit joint distribution or coupling is given.

A.2 Definitions

We are interested in prediction problems with domain shift. There are three random variables: the target domain D_t, the input X, the label Y. They have the following joint distribution:

$$(D_t, X, Y) \overset{\mathrm{d}}{\sim} p_{D_t} \cdot p_{X,Y \,|\, D_t} \tag{4}$$

where we drop the arguments of the probability densities for clarity. We make a variety of convenience assumptions on these random variables (Assumption 6). Crucially, we will be making the Bayes invariance assumption on $p_{D_t,X,Y}$ that can be thought of as a generalized covariate shift assumption (Assumption 4).

We will be studying the effect of changing the representation of the data. This is done by "encoding" X into a representation Z using a conditional distribution $p_{Z \,|\, X}$.

Definition 1 (Encoder)**.** An *encoder* is a conditional distribution $p_{Z \,|\, X} : \mathcal{Z} \times \mathcal{X} \to [0, 1]$ from the input space $\mathcal{X}$ to the representation space $\mathcal{Z}$.

The data together with the representation has the following joint:

$$(D_t, X, Y, Z) \overset{\mathrm{d}}{\sim} p_{D_t} \cdot p_{X,Y \,|\, D_t} \cdot p_{Z \,|\, X} \tag{5}$$

The key thing to notice here is that Z is conditionally independent of Y, D_t given X. In particular, the same encoder is used across all domains.

A.2.1 Risk minimization

Our ultimate goal is to predict Y from the representation Z of X in a manner that is robust to changes in the domain.

We formalize this in the standard way by making predictions $\gamma \in \Gamma$ in a space of predictions or actions. For example the prediction space may be the set of all possible labels $\Gamma = \mathcal{Y}$, in which case we would be predicting deterministic labels. Or we may predict a distribution over labels, in which case the prediction space would be the set of all probability distributions on $\mathcal{Y}$, i.e. $\Gamma = \mathcal{P}(\mathcal{Y})$.

A *predictor* is a function mapping inputs to predictions, i.e., $f : \mathcal{X} \to \Gamma$, or representations to predictions, i.e., $h : \mathcal{Z} \to \Gamma$. For example, f may be a neural network that takes as input a sample x and outputs a vector of logits that parameterize a softmax distribution over finitely many labels.

We select predictors according to the *risk* defined via a loss function $\ell : \mathcal{Y} \times \Gamma \to \mathbb{R}_{\geq 0} \cup \{\infty\}$:

$$\mathrm{R}_f\left[Y \mid X\right] := \mathbb{E}_{p_{X,Y}}\left[\ell(Y, f(X))\right]. \tag{6}$$

In particular, we are interested in the Bayes (minimum) risk over all predictors:

$$\mathrm{R}\left[Y \mid X\right] := \inf_f \mathrm{R}_f\left[Y \mid X\right], \tag{7}$$

We denote the set of all optimal predictors from X as

$$\mathcal{F}^* := \{f \mid \mathrm{R}_f\left[Y \mid X\right] = \mathrm{R}\left[Y \mid X\right]\} \tag{8}$$

Similarly, we define the risk $\mathrm{R}_h\left[Y \mid Z\right]$, the Bayes risk $\mathrm{R}\left[Y \mid Z\right]$, and the set of optimal predictors

$$\mathcal{H}_Z^* := \{h \mid \mathrm{R}_h\left[Y \mid Z\right] = \mathrm{R}\left[Y \mid Z\right]\} \tag{9}$$

from Z, all of which vary as a function of the encoder $p_{Z \mid X}$. Note, in the main body of the paper, we omitted the subscript Z from $\mathcal{H}_Z^*$ for clarity, but we will keep it in the Appendices. We assume that together our loss and prediction space always admit optima (Item 2 of Assumption 2), and thus $\mathcal{F}^*, \mathcal{H}_Z^*$ are always non-empty.

We will be assuming that the risk admits unique optimal prediction when predicting from X (Item 3 of Assumption 2). Thus it makes sense to define the following:

Definition 2 (The Bayes predictor)**.** *The Bayes predictor* $f^* : \mathcal{X} \to \Gamma$ is the unique predictor that is optimal for all $x \in \mathcal{X}$:

$$f^*(x) = \arg\min_{\gamma \in \Gamma} \mathbb{E}_{p_{Y \mid x}}[\ell(Y, \gamma)] \tag{10}$$

Definition 3 (The Bayes image)**.** The image of all the inputs under the Bayes predictor will be denoted as $\Gamma^* = f^{*\rightarrow}(\mathcal{X})$ and called *the Bayes image*.

Note that $\mathcal{F}^*$ becomes a singleton $\{f^*\}$, but it is not necessarily the case for $\mathcal{H}_Z^*$ since we will not be making any uniqueness assumption on optimal prediction from Z.

A.2.2 Domain generalization

We are interested in controlling the risk in a domain generalization setting, and so we define the *domain-conditional risk*,

$$\mathrm{R}_f^d\left[Y \mid X\right] := \mathbb{E}_{p_{X,Y \mid d}}[\ell(Y, f(X))]. \tag{11}$$

$\mathrm{R}^d\left[Y \mid X\right], \mathcal{F}_d^*$ are defined as Eqs. (7) and (8), respectively, but with respect to R_f^d. Similarly, define the Bayes image for domain d as

$$\Gamma_d^* := f^{*\rightarrow}\big(\mathrm{supp}(p_{X \mid d})\big). \tag{12}$$

We also define domain-conditional quantities for prediction from a representation Z. The most important term which we will be investigating is an idealization of the domain generalization worst-case risk.

Definition 4 (IDG risk)**.** Given an encoder $p_{Z \mid X}$ and a distribution p_{D_t, D_s} over a target domain D_t and source domain D_s, the idealized domain generalization worst-case risk, *IDG risk* for short, is the expected worst-case target risk taken over source minimizers, i.e.,

$$\mathrm{R}_{\mathrm{IDG}}\left[Y \mid Z\right] := \mathbb{E}_{p_{D_t, D_s}}\left[\sup_{h \in \mathcal{H}_{Z,D_s}^*} \mathrm{R}_h^{D_t}\left[Y \mid Z\right]\right] \tag{13}$$

Note that the IDG risk is well-defined because $\mathcal{H}_{Z,D_s}^*$ is non-empty by Assumption 2. The desired optimal representations, are then those that minimize the IDG risk.

Definition 5 (Optimal representations for IDG)**.** An encoder $p_{Z^* \mid X}$ is *optimal* for idealized domain generalization if and only if it minimizes the IDG risk, i.e.,

$$\mathrm{R}_{\mathrm{IDG}}\left[Y \mid Z^*\right] = \inf_{p_{Z \mid X}} \mathrm{R}_{\mathrm{IDG}}\left[Y \mid Z\right] \tag{14}$$

A.3 Assumptions

We make a the following assumptions throughout the paper. **All these assumptions should hold for practical settings.**

Assumption 1 (Convenience: discrete probability spaces)**.** All data spaces $(\mathcal{D}, \mathcal{X}, \mathcal{Y}, \mathcal{Z}, \mathcal{A})$ are discrete spaces. Because the distributions of X, Y, D are fixed, we assume for convenience that $\operatorname{supp}(p_X) = \mathcal{X}$, $\operatorname{supp}(p_Y) = \mathcal{Y}$, and $\operatorname{supp}(p_{D_t}) = \mathcal{D}$.

Assumption 1 is a convenience assumption to avoid measure theory for the sake of clarity. It always holds in practice due to finiteness of computers, i.e., all spaces will be finite but arbitrarily large. We believe that our claims can nevertheless be generalized to typical continuous spaces with some minor technical assumptions.

Assumption 2 (Properness of our losses)**.** We assume that our risk always admits optimal predictions:

1. $|\Gamma| > 1$.
2. For all $p_\Upsilon \in \mathcal{P}(\mathcal{Y})$, there exists $\gamma^* \in \Gamma$, such that
$$\mathbb{E}_{p_\Upsilon}[\ell(\Upsilon, \gamma^*)] \leq \mathbb{E}_{p_\Upsilon}[\ell(\Upsilon, \gamma)] \qquad \forall\, \gamma \in \Gamma. \tag{15}$$
3. For all $x \in \mathcal{X}$, there exist $\gamma^* \in \Gamma$, such that
$$\mathbb{E}_{p_{Y \mid x}}[\ell(Y, \gamma^*)] < \mathbb{E}_{p_{Y \mid x}}[\ell(Y, \gamma)] \qquad \forall\, \gamma \neq \gamma^*. \tag{16}$$

Note that for log-loss $\ell(y, \gamma) = -\log \gamma(y)$ and finite $\mathcal{Y}$, these assumptions are satisfied if $\Gamma = \mathcal{P}(\mathcal{Y})$ where the optimal prediction for Item 3 is $\gamma^* = p_{Y \mid x}$ by strict properness [18]. If we consider the 0-1 loss (reverse accuracy) $\ell(y, \gamma) = 1 - \mathbb{1}[y = \gamma]$ with $\Gamma = \mathcal{Y}$ and a finite label space where the optimal prediction for Item 3 is $\gamma^* = \arg\max_{y \in \mathcal{Y}} p_{Y \mid x}(y)$, this assumption is mostly satisfied, except we assume that $p_{Y \mid x}$ has a unique mode.

Assumption 2 serves two purposes: Item 2 ensures that for any representation the optimal predictors from Z exists such that the IDG risk is well-defined as in Def. 5; Item 3 ensures a unique Bayes predictor from X, which simplifies the analysis and is satisfied by common losses as described above.

Assumption 3 (Cardinalities)**.** We assume that
$$|\mathcal{Z}| \geq |\Gamma^*| \geq 2 \tag{17}$$

Assumption 3 is very weak and ensures that optimal representations always exists (Prop. 3).

Assumption 4 (Generalized covariate shift)**.** The Bayes predictor is optimal for all domains. I.e., for all $(x, d) \in \operatorname{supp}(p_{X, D_t}), \gamma \in \Gamma$ such that $\gamma \neq f^*(x)$, we have
$$\mathbb{E}_{p_{Y \mid x, d}}[\ell(Y, f^*(x))] < \mathbb{E}_{p_{Y \mid x, d}}[\ell(Y, \gamma)]\,. \tag{18}$$

For example, in the case of strictly proper scoring rules, e.g. log loss, covariate shift $p_{Y \mid X, D} = p_{Y \mid X}$ is equivalent to the invariance of the Bayes predictor. For the 0-1 loss, this is guaranteed by invariance of the most likely label. For MSE it is guaranteed by the invariance of the expected label. In the latter two cases, Assumption 4 is less stringent than the typical covariate shift assumption.

Assumption 4 is the core assumption for our theoretical results. It ensures that source and target domains are related in a useful way that can be utilized by the representation.

Assumption 5 (Constant Bayes image)**.** The Bayes image is invariant across domains, i.e., for all $d \in \mathcal{D}$,
$$\Gamma^*_d = \Gamma^*. \tag{19}$$

For the case of 0-1 loss, this simply means that the label set for all domains is the same, which is trivial. For log-loss, this means that the set of possible conditional distributions $\Gamma^*_d = \{p_{Y \mid x} \mid x \in \operatorname{supp}(p_{X \mid d})\}$ is the same across domains.

Assumption 5 is crucial to be able to learn. Without it, in the extreme case, one could set each domain to be all examples associated with a single element from the label set (or the Bayes image set) in which case it is impossible to generalize across different domains. Assumption 5 is also necessary to guarantee the existence of optimal representations as in Prop. 3.

Assumption 6 (Domain joint)**.** p_{D_t,D_s} is any distribution such that $\mathrm{supp}(p_{D_t,D_s}) = \mathcal{D} \times \mathcal{D}$.

In a simplified scenario, one could define the source D_s and target D_t as i.i.d. r.v. from p_{D_t}, where $p_{D_t,D_s} = p_{D_t} \cdot p_{D_s} = p_{D_t} \cdot p_{D_t}$ and Assumption 6 is trivially satisfied.

B Proofs

B.1 Lemmas for general losses

An important result that we will be using is the generalized data processing inequality of Bayes risk [56, 15]. We include it here for completeness.

Lemma 1 (Generalized DPI [56, 15])**.** *Let $Z - X - Y$ be a Markov chain of random variables. For any loss function ℓ,*

$$\mathrm{R}\left[Y \mid X\right] \leq \mathrm{R}\left[Y \mid Z\right]. \tag{20}$$

For the case of strictly proper losses (Assumption 2) we can go one step further.

Lemma 2. *Let $Z - X - Y$ be a Markov chain of random variables. Then, under Assumptions 1 and 2 we have that*

$$\mathrm{R}\left[Y \mid Z\right] = \mathrm{R}\left[Y \mid X\right] \quad \Longleftrightarrow \quad \forall h^* \in \mathcal{H}_Z^*, \forall (x,z) \in \mathrm{supp}(p_{X,Z}),\ h^*(z) = f^*(x). \tag{21}$$

Proof. Suppose that for all $h^* \in \mathcal{H}_Z^*$ we have $h^*(z) = f^*(x)$ on the support of $p_{X,Z}$. Then,

$$\begin{aligned}
\mathrm{R}\left[Y \mid X\right] &= \mathbb{E}_{p_{X,Y}}[\ell(Y, f^*(X))] && (22)\\
&= \mathbb{E}_{p_{X,Y} p_{Z \mid X}}[\ell(Y, f^*(X))] && (23)\\
&= \mathbb{E}_{p_{X,Y} p_{Z \mid X}}[\ell(Y, h^*(Z))] && (24)\\
&= \mathbb{E}_{p_{Z,Y}}[\ell(Y, h^*(Z))] && (25)\\
&= \mathrm{R}\left[Y \mid Z\right]. && (26)
\end{aligned}$$

Now suppose there exists a $h^* \in \mathcal{H}_Z^*$ and a pair $(x', z') \in \mathrm{supp}(p_{X,Z})$ such that $h^*(z') \neq f^*(x')$. Then

$$\begin{aligned}
&\mathrm{R}\left[Y \mid Z\right] && (27)\\
&= \mathbb{E}_{p_{X,Z} p_{Y \mid X}}[\ell(Y, h^*(Z))] && (28)\\
&= p_{X,Z}(x', z')\, \mathbb{E}_{p_{Y \mid x'}}[\ell(Y, h^*(z'))] + \sum_{(x,z) \neq (x',z')} p_{X,Z}(x,z)\, \mathbb{E}_{p_{Y \mid x}}[\ell(Y, h^*(z))] && (29)\\
&\geq p_{X,Z}(x', z')\, \mathbb{E}_{p_{Y \mid x'}}[\ell(Y, h^*(z'))] + \sum_{(x,z) \neq (x',z')} p_{X,Z}(x,z)\, \mathbb{E}_{p_{Y \mid x}}[\ell(Y, f^*(x))] && (30)\\
&> p_{X,Z}(x', z')\, \mathbb{E}_{p_{Y \mid x'}}[\ell(Y, f^*(x'))] + \sum_{(x,z) \neq (x',z')} p_{X,Z}(x,z)\, \mathbb{E}_{p_{Y \mid x}}[\ell(Y, f^*(x))] && (31)\\
&= \mathrm{R}\left[Y \mid X\right] && (32)
\end{aligned}$$

Eq. (30) follows by Item 3 of Assumption 2 along with the definition of f^*. Eq. (31) follows by Item 3 of Assumption 2 and the fact that $h^*(z') \neq f^*(x')$. This completes the proof, because Lemma 1 prevents $\mathrm{R}\left[Y \mid Z\right] < \mathrm{R}\left[Y \mid X\right]$. □

B.2 Proof of Theorem 2

First we will show that the desired representation exists by taking all inputs for which the Bayes predictor predicts similarly and "bucketing" them to the same representation. This is a direct extension of the example from Dubois et al.'s (2020) Proposition 6, to the case of proper losses.

Proposition 3 (Existence of optimal representations)**.** *Under Assumptions 1 to 5, there exists an encoder $p_{Z^* \mid X}$ that is optimal for IDG, i.e.,*

$$p_{Z^* \mid X} \in \arg\min_{p_{Z \mid X}} \mathrm{R}\left[Y \mid Z\right] \quad \text{s.t.} \quad \forall\, d \in \mathcal{D},\ \mathrm{supp}(p_{Z \mid d}) = \mathrm{supp}(p_Z). \tag{33}$$

Moreover, we have that

$$\mathrm{R}\left[Y \mid X\right] = \mathrm{R}\left[Y \mid Z^*\right]. \tag{34}$$

Proof. Because we assume arbitrary encoders $p_{Z \mid X}$, the essence of this construction is simple: we embed the Bayes image into $\mathcal{Z}$. Indeed, let $\phi : \Gamma^* \to \mathcal{Z}$ be any one-to-one function, which exists due to Assumption 3 (here we use deterministic one-to-one function for simplicity, the construction can be easily extended to stochastic case). Then let $Z^* = \phi(f^*(X))$. We now verify the properties of $p_{Z^* \mid X}$.

1. Z^* satisfies $\mathrm{R}\left[Y \mid X\right] = \mathrm{R}\left[Y \mid Z^*\right]$. Indeed,

$$\begin{aligned}
\mathrm{R}\left[Y \mid X\right] &= \mathbb{E}_{p_{X,Y}}[\ell(Y, f^*(X))] && (35)\\
&= \mathbb{E}_{p_{X,Y} p_{Z^* \mid X}}[\ell(Y, f^*(X))] && (36)\\
&= \mathbb{E}_{p_{Z^*,Y}}\left[\ell(Y, \phi^{-1}(Z^*))\right] && (37)\\
&\geq \mathrm{R}\left[Y \mid Z^*\right]. && (38)
\end{aligned}$$

   Eq. (37) is by our construction of Z^* and Eq. (38) is by the definition of the Bayes risk. Due to the data processing inequality of Bayes risk (Lemma 1) we also have $\mathrm{R}\left[Y \mid X\right] \leq \mathrm{R}\left[Y \mid Z^*\right]$, from which we conclude that $\mathrm{R}\left[Y \mid X\right] = \mathrm{R}\left[Y \mid Z^*\right]$ and that Eq. (34) holds.

2. Recall that $\Gamma^* = f^{*\rightarrow}(\mathcal{X})$ and $\Gamma^*_d = f^{*\rightarrow}\big(\mathrm{supp}(p_{X \mid d})\big)$. Now let us compute the desired support for all $d \in \mathcal{D}$:

$$\begin{aligned}
\mathrm{supp}(p_{Z^* \mid d}) &= \phi^{\rightarrow}(\Gamma^*_d) && (39)\\
&= \phi^{\rightarrow}(\Gamma^*) && (40)\\
&= \mathrm{supp}(p_{Z^*}). && (41)
\end{aligned}$$

   Eq. (40) is by Assumption 5.

Because $\mathrm{R}\left[Y \mid X\right]$ is the minimum achievable risk by any encoder regardless of constraint (this is by Lemma 1), this implies that $p_{Z^* \mid X}$ is an optimal encoder for IDG. □

The following lemma essentially says that when $\mathrm{R}\left[Y \mid Z\right]$ is minimized, then the optimal predictors for each domain all agree on the intersection of their support.

Lemma 3. *Let $p_{Z \mid X}$ be an encoder such that $\mathrm{R}\left[Y \mid Z\right] = \mathrm{R}\left[Y \mid X\right]$. Under Assumptions 1 and 2, we have that for all $z \in \mathrm{supp}(p_Z)$, there exists $\gamma^* \in \Gamma$ such that*

$$\mathbb{E}_{p_{Y \mid z}}[\ell(Y, \gamma^*)] < \mathbb{E}_{p_{Y \mid z}}[\ell(Y, \gamma)] \qquad \forall\, \gamma \neq \gamma^*. \tag{42}$$

In other words, the restriction of any $h^* \in \mathcal{H}^*_Z$ to $\mathrm{supp}(p_Z)$ is unique. If, in addition, Assumption 4 holds, then for all $(z, d) \in \mathrm{supp}(p_{Z,D_t}), \gamma \in \Gamma$ such that $\gamma \neq h^*(z)$,

$$\mathbb{E}_{p_{Y \mid z,d}}[\ell(Y, h^*(z)] < \mathbb{E}_{p_{Y \mid z,d}}[\ell(Y, \gamma)]. \tag{43}$$

In other words, the restriction of any $h \in \mathcal{H}^*_{Z,d}$ to $\mathrm{supp}(p_{Z \mid d})$ is unique and equal to h^*.

Proof. For the first result, let $z \in \mathrm{supp}(p_Z)$ and consider $x \in \mathrm{supp}(p_{X \mid z})$. By Lemma 2, it must be the case that f^* is constant on $\mathrm{supp}(p_{X \mid z})$. Thus, we can pick $\gamma^* = f^*(x)$. Now, let $\gamma \neq \gamma^*$. We have that,

$$\begin{aligned}
\mathbb{E}_{p_{Y \mid z}}[\ell(Y, \gamma^*)] &= \mathbb{E}_{p_{X \mid z} p_{Y \mid X}}[\ell(Y, \gamma^*)] && (44)\\
&= \mathbb{E}_{p_{X \mid z} p_{Y \mid X}}[\ell(Y, f^*(X))] && (45)\\
&< \mathbb{E}_{p_{X \mid z} p_{Y \mid X}}[\ell(Y, \gamma)] && (46)\\
&= \mathbb{E}_{p_{Y \mid z}}[\ell(Y, \gamma)]. && (47)
\end{aligned}$$

Eq. (44) is due to the conditional independence of Y and Z given X. Eq. (46) is due to Assumption 2 and the definition of the Bayes predictor. Let $h^* : \mathrm{supp}(p_Z) \to \Gamma$ be the unique Bayes predictor from Z.

Now, for the second result, note that

$$\begin{aligned}
\mathrm{R}\left[Y \mid X\right] &= \mathrm{R}_{f^*}\left[Y \mid X\right] && (48)\\
&= \sum_{d \in \mathcal{D}} p_{D_t}(d)\, \mathrm{R}^d_{f^*}\left[Y \mid X\right] && (49)\\
&= \sum_{d \in \mathcal{D}} p_{D_t}(d)\, \mathrm{R}^d\left[Y \mid X\right], \qquad \text{Assumption 4} && (50)
\end{aligned}$$

and

$$\begin{aligned}
\mathrm{R}\left[Y \mid Z\right] &= \mathrm{R}_{h^*}\left[Y \mid Z\right] && (51)\\
&= \sum_{d \in \mathcal{D}} p_{D_t}(d)\, \mathrm{R}^d_{h^*}\left[Y \mid Z\right] && (52)\\
&\geq \sum_{d \in \mathcal{D}} p_{D_t}(d)\, \mathrm{R}^d\left[Y \mid Z\right], && (53)
\end{aligned}$$

where Eq. (53) is due to the definition of (domain-conditional) Bayes risk. Then

$$\begin{aligned}
\mathrm{R}\left[Y \mid Z\right] - \mathrm{R}\left[Y \mid X\right] &\geq \sum_{d \in \mathcal{D}} p_{D_t}(d) \left(\mathrm{R}^d\left[Y \mid Z\right] - \mathrm{R}^d\left[Y \mid X\right]\right) && (54)\\
&\geq 0. \qquad \text{Lemma 1 conditioned on } d && (55)
\end{aligned}$$

Thus, any encoder that achieves $\mathrm{R}\left[Y \mid Z\right] = \mathrm{R}\left[Y \mid X\right]$ also satisfies $\mathrm{R}^d\left[Y \mid Z\right] = \mathrm{R}^d\left[Y \mid X\right]$ for all $d \in \mathcal{D}$ since we assume that $\mathrm{supp}(p_{D_t}) = \mathcal{D}$ in Assumption 1. Now, let $d \in \mathcal{D}$. An argument analogous to Lemma 2 gives us,

$$\forall h \in \mathcal{H}^*_{Z,d}, \forall (x,z) \in \mathrm{supp}(p_{X,Z \mid d}),\ h(z) = f^*(x) = h^*(z). \tag{56}$$

Eq. (56) is derived from $\mathrm{R}^d\left[Y \mid Z\right] = \mathrm{R}^d\left[Y \mid X\right]$ using Assumption 4 in place of Item 3 of Assumption 2 for a specific domain d. Let $z \in \mathrm{supp}(p_{Z \mid d})$ and $\gamma \in \Gamma$ such that $\gamma \neq h^*(z)$. Since $\mathrm{supp}(p_{X \mid z,d}) \subseteq \mathrm{supp}(p_{X \mid z})$, f^* is a constant on $\mathrm{supp}(p_{X \mid z,d})$ and equal to h^*. Now, as above, we have that

$$\begin{aligned}
\mathbb{E}_{p_{Y \mid z,d}}[\ell(Y, h^*(z))] &= \mathbb{E}_{p_{X \mid z,d} p_{Y \mid X,d}}[\ell(Y, h^*(z))] && (57)\\
&= \mathbb{E}_{p_{X \mid z,d} p_{Y \mid X,d}}[\ell(Y, f^*(X))] && (58)\\
&< \mathbb{E}_{p_{X \mid z,d} p_{Y \mid X,d}}[\ell(Y, \gamma)] && (59)\\
&= \mathbb{E}_{p_{Y \mid z,d}}[\ell(Y, \gamma)]. && (60)
\end{aligned}$$

Eq. (59) is due to Assumption 4. □

Corollary 1. *Let $p_{Z \mid X}$ be an encoder such that $\mathrm{R}\left[Y \mid Z\right] = \mathrm{R}\left[Y \mid X\right]$. Under Assumptions 1, 2 and 4 we have that $\mathcal{H}^*_Z \subseteq \mathcal{H}^*_{Z,d}$ for all $d \in \mathcal{D}$ and that for all $d_s, d_t \in \mathcal{D}$*

$$\inf_{h \in \mathcal{H}^*_{Z,d_s}} \mathrm{R}^{d_t}_h\left[Y \mid Z\right] = \mathrm{R}^{d_t}\left[Y \mid Z\right] \tag{61}$$

Proof. $\mathcal{H}^*_Z \subseteq \mathcal{H}^*_{Z,d}$ is immediate from Lemma 3. Now, we have that $\mathrm{R}^{d_t}_h\left[Y \mid Z\right] \geq \mathrm{R}^{d_t}\left[Y \mid Z\right]$. So, the result follows by taking any $h \in \mathcal{H}^*_Z \subseteq \mathcal{H}^*_{Z,d_s}$ in the $\inf$ of Eq. (61). □

Theorem 2 (Characterizing optimal representations for IDG)**.** *Under Assumptions 1 to 6, an encoder $p_{Z \mid X}$ is optimal for idealized domain generalization if and only if it minimizes the Bayes risk while matching the support of $p_{Z \mid d}$ and p_Z for all $d \in \mathcal{D}$, i.e.,*

$$\begin{aligned}
p_{Z \mid X} &\in \arg\min_{p_{Z \mid X}} \mathrm{R}\left[Y \mid Z\right] && (62)\\
&\text{s.t.} \quad \forall\, d \in \mathcal{D},\ \mathrm{supp}(p_{Z \mid d}) = \mathrm{supp}(p_Z) && (63)
\end{aligned}$$

Proof. The IDG risk is lower bounded by $\mathrm{R}[Y \mid X]$:

$$\mathrm{R}_{\mathrm{IDG}}[Y \mid Z] \geq \mathbb{E}_{p_{D_s,D_t}}\left[\inf_{h \in \mathcal{H}^*_{Z,D_s}} \mathrm{R}^{D_t}_h[Y \mid Z]\right] \tag{64}$$

$$\geq \mathbb{E}_{p_{D_s,D_t}}\left[\mathrm{R}^{D_t}[Y \mid Z]\right] \tag{65}$$

$$\geq \mathbb{E}_{p_{D_s,D_t}}\left[\mathrm{R}^{D_t}[Y \mid X]\right] \qquad \text{Lemma 1} \tag{66}$$

$$= \mathrm{R}[Y \mid X] \qquad \text{Assumption 4} \tag{67}$$

We will now show that this lower bound is achieved by an encoder if and only if it satisfies Eqs. (62) and (63), which exist by Prop. 3.

Sufficiency ($\Longleftarrow$): Let $p_{Z \mid X}$ be an encoder that satisfies Eqs. (62) and (63). Note that $\mathrm{R}[Y \mid Z] = \mathrm{R}[Y \mid X]$ by Prop. 3. Let $h^* \in \mathcal{H}^*_Z$, then we have the following IDG risk

$$\mathrm{R}_{\mathrm{IDG}}[Y \mid Z] \tag{68}$$

$$= \mathbb{E}_{p_{D_s,D_t}}\left[\sup_{h \in \mathcal{H}^*_{Z,D_s}} \mathbb{E}_{p_{Z,Y \mid D_t}}[\ell(Y, h(Z))]\right] \tag{69}$$

$$= \mathbb{E}_{p_{D_s,D_t}}\left[\sup_{h \in \mathcal{H}^*_{Z,D_s}} \mathbb{E}_{p_{Z,Y \mid D_t}}[\ell(Y, h^*(Z))]\right] \qquad \text{Lemma 3 under matching support} \tag{70}$$

$$= \mathbb{E}_{p_{D_t}}\left[\mathbb{E}_{p_{Z,Y \mid D_t}}[\ell(Y, h^*(Z))]\right] \qquad \text{constant w.r.t } D_s \tag{71}$$

$$= \mathrm{R}[Y \mid Z] = \mathrm{R}[Y \mid X] \tag{72}$$

Necessity ($\Longrightarrow$): If the IDG risk is $\mathrm{R}[Y \mid X]$, then it must be the case that

$$\mathrm{R}[Y \mid Z] = \mathrm{R}[Y \mid X] \tag{73}$$

$$\sup_{h \in \mathcal{H}^*_{Z,d_s}} \mathrm{R}^{d_t}_h[Y \mid Z] = \mathrm{R}^{d_t}[Y \mid Z] \quad \forall (d_s, d_t) \in \mathrm{supp}(p_{D_s,D_t}) \tag{74}$$

We will prove by contrapositive that Eq. (74) implies support match (Eq. (63)). Suppose that the support match does not hold. Since $\mathrm{supp}(p_Z) = \cup_{d \in \mathcal{D}} \mathrm{supp}(p_{Z \mid d})$ and $\mathrm{supp}(p_{D_s,D_t}) = \mathcal{D} \times \mathcal{D}$ (Assumption 6), there must exist $(d_s, d_t) \in \mathrm{supp}(p_{D_s,D_t})$ such that $\mathrm{supp}(p_{Z \mid d_s}) \neq \mathrm{supp}(p_{Z \mid d_s})$.

Define the set $S = \mathrm{supp}(p_{Z \mid d_s}) \cap \mathrm{supp}(p_{Z \mid d_t})$ and $\bar{S} = \mathrm{supp}(p_{Z \mid d_t}) \setminus \mathrm{supp}(p_{Z \mid d_s})$, let $\rho = P_{Z \mid d_t}(S)$, and let $h^* \in \mathcal{H}^*_Z$. Then,

$$\sup_{h \in \mathcal{H}^*_{Z,d_s}} \mathrm{R}^{d_t}_h[Y \mid Z] \tag{75}$$

$$= \sup_{h \in \mathcal{H}^*_{Z,d_s}} \rho\, \mathbb{E}_{p_{Y,Z \mid S,d_t}}[\ell(Y, h(Z))] + (1-\rho)\, \mathbb{E}_{p_{Y,Z \mid \bar{S},d_t}}[\ell(Y, h(Z))] \tag{76}$$

$$= \sup_{h \in \mathcal{H}^*_{Z,d_s}} \rho\, \mathbb{E}_{p_{Y,Z \mid S,d_t}}[\ell(Y, h^*(Z))] + (1-\rho)\, \mathbb{E}_{p_{Y,Z \mid \bar{S},d_t}}[\ell(Y, h(Z))] \qquad \text{Lem. 3} \tag{77}$$

$$= \rho\, \mathbb{E}_{p_{Y,Z \mid S,d_t}}[\ell(Y, h^*(Z))] + (1-\rho) \sup_{h \in \mathcal{H}^*_{Z,d_s}} \mathbb{E}_{p_{Y,Z \mid \bar{S},d_t}}[\ell(Y, h(Z))] \tag{78}$$

$$= \mathrm{R}^{d_t}[Y \mid Z] + (1-\rho) \sup_{h \in \mathcal{H}^*_{Z,d_s}} \mathbb{E}_{p_{Y,Z \mid \bar{S},d_t}}[\ell(Y, h(Z)) - \ell(Y, h^*(Z))] \tag{79}$$

$$> \mathrm{R}^{d_t}[Y \mid Z] \qquad \text{Lem. 3} \tag{80}$$

Eq. (80) uses the following reasoning. $1 - \rho > 0$ due to support mismatch. For any $h \in \mathcal{H}^*_{Z,d_s}$ such that $h \neq h^*$ on $\bar{S}$ (such an h exists by Item 1 of Assumption 2), we have that

$$\mathbb{E}_{p_{Y,Z \mid \bar{S},d_t}}[\ell(Y, h(Z)) - \ell(Y, h^*(Z))] > 0 \tag{81}$$

by Lemma 3. □

As a corollary from the proof strategy we directly have that the optimal DG risk is simply $\mathrm{R}[Y \mid X]$. This means that using the optimal encoder one can actually perform just as well by training on the source as if you were to directly train on the target using the raw data.

Corollary 2 (Optimal IDG Risk)**.** *Under Assumptions 1 to 6,* $\inf_{p_{Z \mid X}} \mathrm{R}_{\mathrm{IDG}}[Y \mid Z] = \mathrm{R}[Y \mid X]$.

B.3 Impossibility results

As a direct corollary of Thm. 2 we know that it is impossible to learn an optimal representation without knowledge or assumptions on the target domain. We can actually prove the following much stronger negative result, which essentially states that it is impossible to find a useful representation without having some information about the target domain. Specifically, we prove that if there exists a non-trivial target domain on which the representation is advantageous then there exists an infinite amount of target domains on which it is disadvantageous compared to predicting from a constant.

For clarity, we will focus on the proof for the standard accuracy (0-1 loss) which is much shorter and simpler to understand, but note that we can generalize the proof to all losses with the right assumptions.

The key is that outside of the source domain, the label distribution is unconstrained because generalized covariate shift has no effect. In other words, for any domain which gives some probability mass on an example that has not been seen during training, then all possible labels for that example gives a valid domain. Furthermore, if there exists one domain on which the representation is good, then one can construct a domain on which the representation is bad simply by labelling this point as the constant prediction.

Proposition 4 (No free lunch learning representations for DG)**.** *Let ℓ be the 0-1 loss with prediction space $\Gamma = \mathcal{Y}$. Let $\mathrm{Rep} : \mathcal{P}(\mathcal{X}, \mathcal{Y}) \to \mathcal{P}(\mathcal{Z}|\mathcal{X})$ be any algorithm for choosing an encoder $p_{Z \mid X}$ from the data distribution $p_{X,Y}$, C be any constant r.v. that t.v.i. $\mathcal{Z}$, and $p_{X,Y \mid d_s}$ be any desired source distribution such that*

- *there is a unique constant prediction $\gamma_C = \arg\min_{y \in \mathcal{Y}} \mathbb{E}_{p_{Y \mid d_s}}[\ell(Y, y)]$,*
- *and $|\mathcal{X} \setminus \mathrm{supp}(p_{X \mid d_s})| > 1$.*

Let $p_{Z_{d_s} \mid X} := \mathrm{Rep}(p_{X,Y \mid d_s})$ be the chosen source encoder. If there exists a target domain $p_{X,Y \mid d_t^g}$ such that

- ***(Non-trivial support)*** *$\emptyset \neq \mathrm{supp}(p_{X \mid d_t^g}) \subseteq \mathcal{X} \setminus \mathrm{supp}(p_{X \mid d_s})$;*
- ***(Satisfies Bayes image invariance)*** *$\Gamma^*_{d_t^g} = \mathcal{Y}$, i.e., there is at least one example for every possible label;*
- ***(Source encoder is useful)*** *$p_{Z_{d_s} \mid X}$ performs better than a constant representation,*

$$\sup_{h \in \mathcal{H}^*_{Z_{d_s}, d_s}} \mathrm{R}_h^{d_t^g}\left[Y \mid Z_{d_s}\right] < \sup_{h \in \mathcal{H}^*_{C, d_s}} \mathrm{R}_h^{d_t^g}\left[Y \mid C\right], \tag{82}$$

Then there exist multiple target domains d_t^b such that $p_{Z_{d_s} \mid X}$ underperforms a constant encoder,

$$\sup_{h \in \mathcal{H}^*_{Z_{d_s}, d_s}} \mathrm{R}_h^{d_t^b}\left[Y \mid Z_{d_s}\right] > \sup_{h \in \mathcal{H}^*_{C, d_s}} \mathrm{R}_h^{d_t^b}\left[Y \mid C\right]. \tag{83}$$

Proof. Let $h^* \in \mathcal{H}^*_{Z_{d_s}, d_s}$ be any source Bayes predictor corresponding to our encoder. Partition $\mathcal{Z}$ according to whether h^* predicts like the constant or not:

$$\mathcal{Z}_C := \{z \in \mathcal{Z} \mid h^*(z) = \gamma_C\} \qquad \mathcal{Z}_{\neq C} := \mathcal{Z} \setminus \mathcal{Z}_C. \tag{84}$$

We know by assumption that d_t^g is s.t.

$$\sup_{h \in \mathcal{H}^*_{Z_{d_s}, d_s}} \mathrm{R}_h^{d_t^g}\left[Y \mid Z_{d_s}\right] < \sup_{h \in \mathcal{H}^*_{C, d_s}} \mathrm{R}_h^{d_t^g}\left[Y \mid C\right], \tag{85}$$

which is clearly only possible if

$$P_{Z_{d_s} \mid d_t^g}(\mathcal{Z}_{\neq C}) > 0. \tag{86}$$

In other words, there exists some input $x_{\neq C} \in \mathcal{X} \setminus \mathrm{supp}(p_{X \mid d_s})$ that will get represented outside of the constant region, i.e.,

$$P_{Z_{d_s} \mid x_{\neq C}}(\mathcal{Z}_{\neq C}) > 0. \tag{87}$$

We will now construct the desired bad domain d_t^b by giving nearly all mass to this $x_{\neq C}$, specifically, let $p_{X \mid d_t^b}(x_{\neq C}) = 1 - \delta$ for some $0 < \delta < 1$. We assign this example to the constant label, i.e., $p_{Y \mid x_{\neq C}, d_t^b}(\gamma_C) = 1$. The rest of the target domain mass δ is distributed as with the source domain, i.e., $p_{X,Y \mid d_t^b}(x, y) = \delta \cdot p_{X,Y \mid d_s}(x, y)$ for all $x, y \in \operatorname{supp}(p_{X,Y \mid d_s})$. Importantly, the constructed domain d_t^b is valid. Indeed, the Bayes image is the same as the source's (Assumption 5), because we removed no prediction γ from the source's Bayes image ($\delta > 0$). We added no new prediction γ, because $f^*(x_{\neq C}) = \gamma_C \in \mathcal{Y}$ which must already have been in Γ^* due to the validity of d_t^g.

Now let us compute the desired risk for that "bad" domain and show that the desired encoder performs worse than a constant encoder.

$$\sup_{h \in \mathcal{H}^*_{Z_{d_s}, d_s}} \mathrm{R}_h^{d_t^b}[Y \mid Z_{d_s}] \tag{88}$$

$$= \sup_{h \in \mathcal{H}^*_{Z_{d_s}, d_s}} (1-\delta)\, \mathbb{E}_{p_{Z_{d_s} \mid x_{\neq C}}}[1 - \mathbb{1}[\gamma_C = h(Z_{d_s})]] + \delta\, \mathrm{R}_h^{d_s}[Y \mid Z_{d_s}] \tag{89}$$

$$\geq (1-\delta)(1 - P_{Z_{d_s} \mid x_{\neq C}}(\mathcal{Z}_C)) \tag{90}$$

$$= (1-\delta) P_{Z_{d_s} \mid x_{\neq C}}(\mathcal{Z}_{\neq C}) \tag{91}$$

In contrast, it is easy to show that $\sup_{h \in \mathcal{H}^*_{C, d_s}} \mathrm{R}_h^{d_t^b}[Y \mid C] \leq \delta$ because the constant predictor would be perfect for $x_{\neq C}$. So any choice of $0 < \delta < \frac{P_{Z_{d_s} \mid x_{\neq C}}(\mathcal{Z}_{\neq C})}{1 + P_{Z_{d_s} \mid x_{\neq C}}(\mathcal{Z}_{\neq C})}$, would satisfy Eq. (83). We conclude the proof by noting that there are infinitely many such choices of δ, and any choice of those would result in a different valid bad domain d_t^b.

□

Note that representations can often be much worse than using a constant r.v. Specifically, if an encoder $p_{Z \mid X}$ maps an x outside of the source support then there exists an infinite number of target domains where that representation is the worst possible representation.

Proposition 5 (Worst representation)**.** *Let* $\mathrm{Rep}, p_{Y,X \mid d_s}, p_{Z_{d_s} \mid X}, \ell$ *be as in Prop. 1, and* $\epsilon > 0$*. If there exists an example* $x_b \in \mathcal{X} \setminus \operatorname{supp}(p_{X \mid d_s})$ *that is mapped outside of the source support, i.e.,* $\operatorname{supp}(p_{Z_{d_s} \mid x_b}) \cap \operatorname{supp}(p_{Z \mid d_s}) = \emptyset$*, then there exist many target domains* $p_{X,Y \mid d_t}$ *s.t.* $p_{Z_{d_s} \mid X}$ *is* ϵ *close to the worst possible loss, i.e.,*

$$\sup_{h \in \mathcal{H}^*_{Z_{d_s}, d_s}} \mathrm{R}_h^{d_t}[Y \mid Z_{d_s}] \geq 1 - \epsilon. \tag{92}$$

Proof. By assumption there exists an x_b whose support is outside the source support. Then similarly to Prop. 1 we construct a bad target domain d_t by giving nearly all mass to that example $p_{X \mid d_t}(x_b) = 1 - \delta$ where $\delta > 0$ and assign with probability 1 to some label that is in the source Bayes image, i.e., $p_{Y \mid x_b, d_t}(\gamma_b) = 1$ for some $\gamma_b \in \Gamma^*_{d_s}$. The rest of the target domain mass δ is distributed as in Prop. 1 to the source inputs. As in Prop. 1, such a target domain d_t satisfies our assumptions. Now let us compute the risk for that d_t and show that the desired encoder performs arbitrarily bad.

$$\sup_{h \in \mathcal{H}^*_{Z_{d_s}, d_s}} \mathrm{R}_h^{d_t}[Y \mid Z_{d_s}] \tag{93}$$

$$= \sup_{h \in \mathcal{H}^*_{Z_{d_s}, d_s}} (1-\delta)\, \mathbb{E}_{p_{Z_{d_s} \mid x_b}}[1 - \mathbb{1}[\gamma_b = h(Z_{d_s})]] + \delta\, \mathrm{R}_h^{d_s}[Y \mid Z_{d_s}] \qquad \text{Eq. (89)} \tag{94}$$

$$\geq \sup_{h \in \mathcal{H}^*_{Z_{d_s}, d_s}} (1-\delta)\, \mathbb{E}_{p_{Z_{d_s} \mid x_b}}[1 - \mathbb{1}[\gamma_b = h(Z_{d_s})]] \tag{95}$$

$$= 1 - \delta \tag{96}$$

Eq. (96) uses the fact that $\mathcal{H}^*_{Z_{d_s}, d_s}$ is unconstrained outside of the source support and that by assumption $\operatorname{supp}(p_{Z_{d_s} \mid x_b}) \cap \operatorname{supp}(p_{Z_{d_s} \mid d_s}) = \emptyset$. To achieve the sup $1 - \delta$ it then suffices to predict an $\gamma \neq \gamma_b \in \Gamma$. We thus see that Eq. (92) holds for d_t as long as $0 < \delta < \epsilon$. We conclude the proof by noting that there is an infinite possible choices of δ each of which give rise to a bad target domain. □

B.4 Augmentations

Prop. 2 shows that the optimal representations for IDG can be learned with augmentations in a self-supervised fashion. Here, we provide formal definitions, assumptions, and proofs.

Definition 6 (Augmenter)**.** An *augmenter* is a conditional distribution $p_{A \mid X} : \mathcal{A} \times \mathcal{X} \to [0, 1]$ from the input space $\mathcal{X}$ to an augmentation space $\mathcal{A}$. For example, in CLIP $\mathcal{X}$ is the space of images and $\mathcal{A}$ is the space of text. In standard SSL, $\mathcal{A}$ is typically the same as $\mathcal{X}$ (e.g., both $\mathcal{X}$ and $\mathcal{A}$ are the space of images).

Definition 7 (Augmentation conditional set)**.** Given an augmenter $p_{A \mid X}$, define the augmentation conditional set as the set of conditionals of A given X:

$$\mathcal{P}^*(A \mid X) := \{p_{A \mid x} \mid x \in \mathcal{X}\} \tag{97}$$

Similarly, we can define the augmentation conditional set for domain d:

$$\mathcal{P}^*_d(A \mid X) := \{p_{A \mid x} \mid x \in \operatorname{supp}(p_{X \mid d})\} \tag{98}$$

These sets are clearly countable. Note that the augmentation conditional set can be seen as a special case of the Bayes image (Def. 3) if we view the augmentation A as the label and consider the log-loss where the conditional distribution is the Bayes optimal predictor due to its strict properness [18].

Assumption 7 (Finite augmentation entropy)**.** We consider the augmenter $p_{A \mid X}$ such that the entropy of the augmentation A is finite, i.e., $\mathrm{H}[A] < \infty$.

Assumption 8 (Cardinalities)**.** We assume that

$$|\mathcal{Z}| \geq |\mathcal{P}^*(A \mid X)| \tag{99}$$

This is a similar assumption as Assumption 3, which ensures the existence of optimal representations.

Assumption 9 (Domain-covering augmentation)**.** We assume that the augmentation A is *domain-covering*, i.e., the augmentation conditional set is invariant across domains,

$$\mathcal{P}^*_d(A \mid X) = \mathcal{P}^*(A \mid X), \quad \forall d \in \mathcal{D} \tag{100}$$

This assumption is generalized from the constant Bayes image assumption (Assumption 5), which guarantees the existence of optimal representations.

Domain-covering augmentations essentially ensures that each augmentation conditional $p_{A \mid x} \in \mathcal{P}^*(A \mid X)$ is seen at least once in all domains. If we introduce an equivalence relation $\sim$ as $x \sim x'$ iff $p_{A \mid x} = p_{A \mid x'}$ and the equivalence class $[x] := \{x' \in \mathcal{X} \mid x' \sim x\}$. Under this relation, it is easy to see that the above assumption is satisfied if and only if, for all possible equivalence classes $[x] \in \{[x'] \mid x' \in \mathcal{X}\}$, we have that $[x]$ has intersections with all domains:

$$[x] \cap \operatorname{supp}(p_{X \mid d}) \neq \emptyset, \quad \forall d \in \mathcal{D} \tag{101}$$

Not all augmentations are domain-covering. In particular, the standard image augmentations used by typical SSL models like SimCLR are not domain-covering, but the text-image augmentations of CLIP nearly are, as discussed in the main body (Sec. 4).

Assumption 10 (Bayes-preserving augmentation)**.** We assume that the augmentation A is *Bayes-preserving*, i.e., $\forall x, x' \in \mathcal{X}$,

$$p_{A \mid x} = p_{A \mid x'} \implies f^*(x) = f^*(x'). \tag{102}$$

Under the notion of equivalence relation in Assumption 9, this means that for each equivalence class $[x]$, all $x' \in [x]$ have the same Bayes prediction. Note that most augmentations used in practice like standard image augmentations are Bayes-preserving.

Next, we show that under the above assumptions, we can learn optimal representations by maximizing the mutual information $\mathrm{I}[A; Z]$ (in the case of log-loss ℓ) under the support match constraint. We use log-loss simply because it is typically the loss used for training in practice. Note that the learned representations are optimal for any strict proper losses.

Proposition 6 (Optimal encoders without labels)**.** *Let $p_{A \mid X}$ be an augmenter. Under Assumptions 1 to 10, any encoder $p_{Z \mid X}$ such that*

$$p_{Z \mid X} \in \arg\max_{p_{Z \mid X}} \; \mathrm{I}[A; Z] \tag{103}$$

$$\text{s.t.} \quad \forall\, d \in \mathcal{D},\; \mathrm{supp}(p_{Z \mid d}) = \mathrm{supp}(p_Z) \tag{104}$$

is optimal for idealized domain generalization.

Proof. The support match constraint Eq. (104) is equivalent to the support match constraint Eq. (63). Thus, Prop. 3 and Thm. 2 state that we only need to prove that maximizing the mutual information of A and Z under the support constraint implies that

$$\mathrm{R}\left[Y \mid Z\right] = \mathrm{R}\left[Y \mid X\right]. \tag{105}$$

We will prove this by constructing an optimal predictor h^*.

Since $\mathrm{H}[A] < \infty$ (Assumption 7) we have that

$$\arg\max_{p_{Z \mid X}} \mathrm{I}[A; Z] = \arg\min_{p_{Z \mid X}} \mathrm{H}[A \mid Z]. \tag{106}$$

Note the fact that the conditional entropy is the Bayes risk under the log-loss [18], i.e., $\mathrm{H}[A \mid Z] = \mathrm{R}\left[A \mid Z\right]$. By construction, A satisfies covariate shift w.r.t. X (thus Bayes invariant) since $A - X - D$ forms a Markov chain. Together with Assumptions 1 and 7 to 9, it means that the optimization problem in Eqs. (103) and (104) satisfies the assumptions of Prop. 3, with A in place of Y. Thus, an optimal encoder satisfies $\mathrm{R}\left[A \mid Z\right] = \mathrm{R}\left[A \mid X\right]$, which leads to

$$\mathrm{H}[A \mid Z] = \mathrm{H}[A \mid X]. \tag{107}$$

By Assumption 7, we can invoke Lemma 2 with the fact that $A - X - Z$ forms a Markove chain to show that for all $(x, z) \in \mathrm{supp}(p_{X,Z})$

$$p_{A \mid z} = p_{A \mid x}, \tag{108}$$

as the conditional distributions are the Bayes optimal predictors due to strict properness of log-loss.

Now, define the following equivalence relation on $\mathcal{X}$,

$$x \sim x' \quad \Longleftrightarrow \quad p_{A \mid x} = p_{A \mid x'}. \tag{109}$$

Because the number of equivalence classes under $\sim$ is countable, there exists a maximal invariant $M : \mathcal{X} \to \mathbb{N}$ from $\mathcal{X}$ to the natural numbers [for our definition of a maximal invariant see Definition 2, 15]. By Assumption 10, f^* is invariant on the equivalence classes $[x] := \{x' \in \mathcal{X} \mid x' \sim x\}$ for all $x \in \mathcal{X}$. Thus, there exists a function $g : \mathbb{N} \to \mathcal{A}$ such that $f^* = g \circ M$ [Lemma 5, 15]. Given $z \in \mathrm{supp}(p_Z)$, we construct h^* in the following way. Let $x_z \in \mathrm{supp}(p_{X \mid z})$ be any input point that could have led to this representation z and define

$$h^*(z) = g(M(x_z)). \tag{110}$$

By Eq. (108) we are guaranteed that all $x \in \mathrm{supp}(p_{X \mid z})$ share the same value for f^* since they are in the same equivalence class. Thus, by the definition of M we have that

$$M(x_Z) = M(X) \quad \text{for} \quad (X, Z) \sim p_{X,Z}. \tag{111}$$

Therefore,

$$\begin{aligned}
\mathrm{R}_{h^*}\left[Y \mid Z\right] &= \mathbb{E}_{p_{Y,Z}}[\ell(Y, h^*(Z)] && (112)\\
&= \mathbb{E}_{p_{Y \mid X} p_{X,Z}}[\ell(Y, h^*(Z)] && (113)\\
&= \mathbb{E}_{p_{Y \mid X} p_{X,Z}}[\ell(Y, g(M(x_Z)))] && \text{Eq. (110)} \quad (114)\\
&= \mathbb{E}_{p_{Y \mid X} p_{X,Z}}[\ell(Y, g(M(X)))] && \text{Eq. (111)} \quad (115)\\
&= \mathbb{E}_{p_{Y \mid X} p_{X,Z}}[\ell(Y, f^*(X))] = \mathrm{R}\left[Y \mid X\right]. && (116)
\end{aligned}$$

□

C Practical objectives

Proposition 6 provides an objective to obtain the desired optimal representations, compared to Thm. 2 it is more practical in that it does not require access to the labels but right augmentations. There are nevertheless multiple remaining issues for deriving objectives that can be trained with in practice. Specifically, (i) the support constraint is hard to satisfy in practice; (ii) mutual information $\mathrm{I}[A; Z]$ is hard to estimate from samples [42]; (iii) the objective is constrained which is harder to optimize. We will now show different objectives and variational bounds of them that do not suffer from these issues, and could still recover the desired encoders in their optima. In contrast to the proofs of main theoretical results (previous section), here the derivations will be less formal.

As we have seen in Proposition 6, the optimal representation achieves $\mathrm{I}[A; Z] = \mathrm{I}[A; X]$, and thus $\mathrm{H}[A \mid Z] = \mathrm{H}[A \mid X]$ (Eq. (107)). In the following, we will rewrite the objective as the constrained optimization:

$$p_{Z \mid X} \in \arg\min_{p_{Z \mid X}} \mathrm{B}[Z, X, Y, D] \tag{117}$$

$$\text{s.t.} \quad \mathrm{H}[A \mid Z] = \mathrm{H}[A \mid X] \tag{118}$$

where we introduce the *domain bottleneck* $\mathrm{B}[Z, X, Y, D]$ as the objective for enforcing support match (which we denote as $\mathrm{B}[Z, D]$ in the main body for simplicity). The requirement on the domain bottleneck objective is that minimizing Eq. (117) under Eq. (118) implies that the support match constraint holds (and can be achieved by some encoder), which leads to optimal representations for idealized domain generalization. Different domain bottlenecks will be derived later this section. We can then use Lagrangian relaxation to get the following unconstrained objectives.

$$\arg\min_{p_{Z \mid X}} \quad \mathrm{H}[A \mid Z] + \lambda\, \mathrm{B}[Z, X, Y, D] \tag{119}$$

The first term can be easily optimized using variational objectives. Throughout the paper, we will use a contrastive variational upper bound which is based on InfoNCE [40]. Namely, let $\mathbf{X} := \left\{X, X_1^-, \ldots, X_n^-\right\}$ be a sequence of samples where each negative X_i^- is independently sampled from the marginal p_X. Now let $\mathbf{A} := \left\{A^+, A_1^-, \ldots, A_n^-\right\}$ be the sequence of augmented examples that come from independently augmenting each $X \in \mathbf{X}$ using the augmenter $p_{A \mid X}$. In particular, the augmented positive A^+ has the marginal $p_{A \mid X}$ while the negatives A_i^- follow the marginal p_A. Let Z be the representation of X by passing it through the encoder $p_\varphi := p_{Z \mid X}$ parameterized by φ and s_ψ the critic function parametrized by ψ used to score which $A' \in \mathbf{A}$ is the positive augmentation. Then we have a variational distribution of $p_{A \mid Z}$:

$$q_{\psi, \mathbf{A}}(A \mid Z) := \frac{\exp s_\psi(A, Z)}{\sum_{A' \in \mathbf{A}} \exp s_\psi(A', Z)} \tag{120}$$

which leads to the following variational bound:

$$\mathrm{H}[A \mid Z] = \mathbb{E}_{p_{X,Z}}\left[-\log p_{A \mid Z}(A \mid Z)\right] \leq \mathbb{E}_{p_{\mathbf{A},X,Z}}\left[-\log q_{\psi,\mathbf{A}}(A^+ \mid Z)\right] \tag{121}$$

with equality if the variational families s_ψ, p_φ are unconstrained and and we use infinite samples ($n \to \infty$). Typically the critic is separable, i.e., $s_\psi(A, Z) := g_\psi(A)^T h_\psi(Z)$. As discussed in the main body, it can be tied with p_φ when $\mathcal{A} = \mathcal{X}$.

In the following we focus on the second term $\mathrm{B}[Z, X, Y, D]$ and discuss several choices.

C.1 Mutual information bottleneck $\mathrm{B}[Z, X, Y, D] = \mathrm{I}[Z; X]$

The first bottleneck we consider is so called mutual information (MI) bottleneck $\mathrm{B}[Z, X, Y, D] = \mathrm{I}[Z; X]$, which was introduced by Tishby et al. [53] to achieve a tradeoff between the predictive power and the complexity of representations. Intuitively, it tries to remove all information of Z that is not needed for maximizing $\mathrm{I}[Z; A]$. In particular, using the fact that $Z - X - D$ forms a Markov chain and the chain rule of MI, we have $\mathrm{I}[Z; X] = \mathrm{I}[Z; X, D] = \mathrm{I}[Z; D] + \mathrm{I}[Z; X \mid D]$. Thus, it not only minimizes $\mathrm{I}[Z; D]$, i.e., matches the representations' distribution *across* domains, but also minimizes $\mathrm{I}[Z; X \mid D]$, i.e., matches the representations' distribution *inside* domains.

Why The key to show is that minimizing Eq. (117), i.e., $\arg\min_{p_{Z\,|\,X}} \mathrm{I}[Z;X]$ under $\mathrm{I}[A;Z] = \mathrm{I}[A;X]$, implies the support match constraint. This can be seen as a specific subcase of Dubois et al.'s (2021) Corollary 15 with A in place of Y and $M(X)$ induced by $p_{A\,|\,X}$ as in the proof of Prop. 6. From the corollary, we know that $\min_{p_{Z\,|\,X}} \mathrm{I}[Z;X] = \mathrm{H}[M(X)]$ which can be achieved by any Z s.t. $p_{Z\,|\,x} = p_{Z\,|\,x'} \iff M(x) = M(x')$. With the assumption of domain-covering augmentations (Assumption 9), we have that the set of maximal invariant $\{M(x) \,|\, x \in \mathrm{supp}(p_{X\,|\,d})\}$ is invariant across domains. Then we directly have $\mathrm{supp}(p_{Z\,|\,d}) = \cup_{x\in\mathrm{supp}(p_{X\,|\,d})}\mathrm{supp}(p_{Z\,|\,x}) = \cup_{x\in\mathrm{supp}(p_X)}\mathrm{supp}(p_{Z\,|\,x}) = \mathrm{supp}(p_Z)$, where we use the fact that x within the same equivalence class has the the same $p_{Z\,|\,x}$.

How Essentially, we can use any variational upper bound of mutual information. We consider the one used by Variational Information Bottelenck [1], i.e.,

$$\begin{aligned}
\mathrm{I}[Z;X] &= \mathbb{E}_{p_{X,Z}}\left[\log \frac{p_\varphi(X\,|\,Z)}{p_Z(Z)}\right] && (122)\\
&= \mathbb{E}_{p_{X,Z}}\left[\log \frac{p_\varphi(X\,|\,Z)}{q_\theta(Z)}\right] - \mathrm{D}_{\mathrm{KL}}[p_Z(Z)\|q_\theta(Z)] && (123)\\
&\le \mathbb{E}_{p_{X,Z}}\left[\log \frac{p_\varphi(X\,|\,Z)}{q_\theta(Z)}\right] && (124)\\
&= \mathbb{E}_{p_X}[\mathrm{D}_{\mathrm{KL}}[p_\varphi(Z\,|\,X)\|q_\theta(Z)]] && (125)
\end{aligned}$$

where a variational distribution q_θ is used to approximate p_Z and is jointly optimized with p_φ to minimize the bound. The approximation gap of the bound is $\mathrm{D}_{\mathrm{KL}}[p_Z(Z)\|q_\theta(Z)]$. Then the final loss becomes

$$\mathcal{L}_{\mathrm{MI}}(\psi,\varphi,\theta) := \mathbb{E}_{p_{X,\mathbf{A},Z}}\left[-\log \frac{\exp s_\psi(A^+,Z)}{\sum_{A'\in\mathbf{A}} \exp s_\psi(A',Z)} + \lambda\,\mathrm{D}_{\mathrm{KL}}[p_\varphi(Z\,|\,X)\|q_\theta(Z)]\right] \tag{126}$$

which recovers the optimal encoder in the case of unconstrained variational families for $p_\varphi, q_\theta, s_\psi$, infinite samples $|\mathbf{A}|$, and any $\lambda > 1$ [15].

C.2 Entropy bottleneck $\mathrm{B}[Z,X,Y,D] = \mathrm{H}[Z]$

The entropy (Ent) bottleneck introduced in the main body is a special case of the MI bottleneck, where the encoder is a deterministic mapping, i.e., $p_\varphi(Z\,|\,x)$ is a dirac delta function for all $x \in \mathcal{X}$ and we denote by $e_\varphi(x)$ the deterministic encoder s.t. $p_\varphi(e_\varphi(x)\,|\,x) = 1$.

Why In the deterministic case, the MI bottleneck becomes the entropy bottleneck because $\mathrm{I}[X;Z] = \mathrm{H}[Z] - \mathrm{H}[Z\,|\,X] = \mathrm{H}[Z]$, where we use the fact that $\mathrm{H}[Z\,|\,X] = 0$. Importantly, considering only deterministic encoders does not constrain our ability to learning optimal encoders. Indeed, just as with the MI bottleneck optimizing the objective with the entropy bottleneck under $\mathrm{I}[A;Z] = \mathrm{I}[A;X]$ will recover encoders s.t. $e_\varphi(x') = e_\varphi(x) \iff M(x) = M(x')$, which also satisfies the support match constraint as discussed before.

How Using the same derivation as the MI bottleneck, we can derive the variational upper bound on entropy

$$\mathrm{H}[Z] \le \mathbb{E}_{p_Z}[-\log q_\theta(Z)] \tag{127}$$

which is the standard variational bound used in neural compression [3, 52]. Putting all together, we have

$$\mathcal{L}_{\text{Ent}}(\psi,\theta,\varphi) := \mathbb{E}_{p_{X,\mathbf{A},Z}}\left[-\log \frac{\exp s_\psi(A^+,Z)}{\sum_{A'\in\mathbf{A}} \exp s_\psi(A',Z)} - \lambda \log q_\theta(Z)\right] \tag{128}$$

which also recovers the optimal encoder with unconstrained variational families, infinite samples, and $\lambda > 1$ as with the MI bottleneck. The detialed algorithm is provided in Algorithm 2. Note that the discreteness of Z could lead to difficulty of gradient-based optimization, and we follow Ballé et al. [3] to add uniform noise to Z as a differentiable substitute for rounding during training. In our experiments, we will mostly use the Ent bottleneck instead of the MI bottleneck to avoid introducing stochastic encoders.

Algorithm 2 Ent objective

Require: $e_\varphi, s_\psi, q_\theta, X, n$

$Z \leftarrow e_\varphi(X)$

$A^+ \leftarrow \text{sample}(p_{A \mid X})$

$\{(X_i^-, A_i^-)\}_{i=1}^n \xleftarrow{\text{i.i.d.}} \text{sample}(p_{X,A})$

$\mathbf{A} \leftarrow \{A^+\} \cup \{A_i^-\}_{i=1}^n$

$\mathcal{L}_{\text{aug}} \leftarrow -\log \frac{\exp s_\psi(A^+, Z)}{\sum_{A' \in \mathbf{A}} \exp s_\psi(A', Z)}$ $\quad \triangleright \mathrm{H}[A \mid Z]$

$\mathcal{L}_{\text{supp}} \leftarrow -\log q_\theta(Z)$ $\quad \triangleright \mathrm{H}[Z]$

return $\mathcal{L}_{\text{Ent}} = \mathcal{L}_{\text{aug}} + \lambda \mathcal{L}_{\text{supp}}$

C.3 Contrastive adversarial domain bottleneck $\mathrm{B}[Z, X, Y, D] = \mathrm{I}[Z; D]$

The previous two bottlenecks require to remove the information of Z (about X) as much as possible, which seems to be unnecessary since our ultimate goal is to match the support of Z across domains. Now we introduce a bottleneck $\mathrm{B}[Z, X, Y, D] = \mathrm{I}[Z; D]$ which we only seek to remove the information of Z about the domain D. This is very related to the work on invariant representation learning for domain generalization/adaptation [e.g., 17, 35]. We derive a new variational bound called the contrastive adversarial domain (CAD) bottleneck that is more stable to train and leads to better empirical performance. For simplicity we consider the deterministic encoder $e_\varphi(x)$ as with the main body.

Why Similar to the previous analysis, we aim to show that $\arg\min_{p_{Z \mid X}} \mathrm{I}[Z; D]$ under $\mathrm{I}[A; Z] = \mathrm{I}[A; X]$ leads to the support match constraint. Using Eq. (111) we have $\mathrm{I}[Z; D] = \mathrm{I}[Z, M(X_Z); D] = \mathrm{I}[Z, M(X); D] = \mathrm{I}[M(X); D] + \mathrm{I}[Z; D \mid M(X)]$ where the last equality uses the chain rule of mutual information. Due to the non-negativity of (conditional) mutual information, we have that the minimum of $\mathrm{I}[Z; D]$ under $\mathrm{I}[A; Z] = \mathrm{I}[A; X]$ is $\mathrm{I}[M(X); D]$. Then we show the minimum is achievable by constructing the same optimal encoder $e_\varphi(X)$ as the Ent bottleneck which clearly satisfies $\mathrm{I}[Z; D \mid M(X)] = 0$. It is then easy to show that the support match constraint has to hold when $\mathrm{I}[Z; D \mid M(X)] = 0$ by contrapositive. Indeed, suppose that the support constraint does not hold then it must be true that $\mathrm{I}[Z; D \mid M(X)] > 0$ and so the encoder cannot be optimal.

How The typical way of minimizing $\mathrm{I}[Z; D]$ is to derive the variational bound as

$$\begin{aligned}
\mathrm{I}[Z; D] &= \mathrm{H}[D] - \mathrm{H}[D \mid Z] && (129)\\
&= (const) - \mathbb{E}_{p_{D,Z}}\big[-\log p_{D \mid Z}(D \mid Z)\big] && (130)\\
&\geq (const) - \mathbb{E}_{p_{D,Z}}[-\log q_\phi(D \mid Z)] && (131)
\end{aligned}$$

where a variational distribution (or domain classifier) q_ϕ is used to approximate $p_{D \mid Z}$ and jointly trained to *maximize* the bound. This recovers the domain-adversarial training method as introduced in Ganin et al. [17]. However, this has two potential issues: 1) it gives a *lower* bound instead of the desired upper bound on $\mathrm{I}[Z; D]$; 2) it requires adversarial training which is not stable [20, 30].

We propose the contrastive adversarial domain (CAD) bottleneck, which is based on the above explicit version but uses a variational distribution $q_\phi(D \mid Z)$ that is *tied with other parts of the model*, thus no need to learn a domain classifier. Specifically, as with Eq. (120), we first introduce a contrastive variational distribution $q_{\varphi,\mathbf{X}}(X \mid Z)$ of $p_{X \mid Z}$ as:

$$q_{\varphi,\mathbf{X}}(X \mid Z) := \frac{\exp s_\varphi(X, Z)}{\sum_{X' \in \mathbf{X}} \exp s_\varphi(X', Z)} \tag{132}$$

where $s_\varphi(X, Z) := e_\varphi(X)^T Z$ is tied with the encoder e_φ. Since $p_{D \mid Z}$ can be rewritten as $\mathbb{E}_{p_{X \mid Z}}\big[p_{D \mid X}\big]$ using the fact that $D - X - Z$ forms a Markov chain, we obtain can the following variational distribution:

$$q_{\varphi,\mathbf{X}}(D \mid Z) = \mathbb{E}_{q_{\varphi,\mathbf{X}}}\big[p_{D \mid X}(D \mid X)\big] \tag{133}$$

which recovers $p_{D \mid Z}$ when $q_{\varphi,\mathbf{X}} = p_{X \mid Z}$. This is the case where $s_\varphi(X, Z) \propto \log p_{X,Z}(X, Z)$ and infinite samples $n \to \infty$. Note that $p_{D \mid X}$ is still not available, we can use a count estimate $\hat{p}_{\mathbf{D},\mathbf{X}}$ in practice. In particular, we obtain a collection $\mathbf{D}$ by taking each $X' \in \mathbf{X}$ and independently

sampling D' from $p_{D \mid X'}$ to get $\mathbf{D} := \{D, D_1^-, \dots, D_n^-\}$. In other words, (D, X) and (D_i^-, X_i^-) for $i \in [n] := \{1, \dots, n\}$ are all i.i.d. sampled from $p_{D,X}$. Then we use a count estimate

$$\hat{p}_{\mathbf{D},\mathbf{X}}(d \mid x) = \frac{\mathbb{I}\left(X = x, D = d\right) + \sum_{i=1}^{n} \mathbb{I}\left(X_i^- = x, D_i^- = d\right)}{\mathbb{I}\left(X = x\right) + \sum_{i=1}^{n} \mathbb{I}\left(X_i^- = x\right)} \tag{134}$$

which is an accurate estimate with infinite samples. This leads to our final variational distribution:

$$q_{\varphi,\mathbf{X},\mathbf{D}}(D \mid Z) = \sum_{X' \in \mathbf{X}} q_{\varphi,\mathbf{X}}(X' \mid Z) \hat{p}_{\mathbf{D},\mathbf{X}}(D \mid X') \tag{135}$$

Putting all together we get that the final loss:

$$\mathcal{L}_{\text{CAD}}(\varphi, \psi) := \mathbb{E}_{p_{\mathbf{D},\mathbf{X},\mathbf{A},Z}} \left[-\log q_{\psi,\mathbf{A}}(A^+ \mid Z) + \lambda \log \left(\sum_{X' \in \mathbf{X}} q_{\varphi,\mathbf{X}}(X' \mid Z) \hat{p}_{\mathbf{D},\mathbf{X}}(D \mid X') \right) \right]. \tag{136}$$

In practice, $\hat{p}_{\mathbf{D},\mathbf{X}}(D \mid X)$ is typically a dirac delta function since it is rare to have the same samples in a batch. Thus, in Eq. (135) we only need to sum $q_{\varphi,\mathbf{X}}(X' \mid Z)$ over those associated with the same domain label D as X, which reduces Eq. (136) to

$$\mathcal{L}_{\text{CAD}}(\varphi, \psi) := \mathbb{E}_{p_{\mathbf{D},\mathbf{X},\mathbf{A},Z}} \left[-\log q_{\psi,\mathbf{A}}(A^+ \mid Z) + \lambda \log \left(\sum_{X' \in \mathbf{X}_D} q_{\varphi,\mathbf{X}}(X' \mid Z) \right) \right]. \tag{137}$$

with a detailed algorithm in Algorithm 1. Note that it is easy to generalize Algorithm 1 to parallel computation within a batch of samples. Indeed, for *each* sample in the batch, we can view all other samples in the batch as negatives and compute the loss efficiently in parallel.

C.4 Conditional CAD $\mathrm{B}[Z, X, Y, D] = \mathrm{I}[Z; D \mid Y]$

The analysis of the CAD bottleneck also implies that we can minimize the conditional mutual information $\mathrm{I}[Z; D \mid M(X)]$ if we have access to $M(X)$. However, since $M(X)$ is typically not available in practice, we consider the special case where $M(X) = Y$. In particular, this is the case where the labels are available and the supervised augmentations are used (see Fig. 2c). This reduces the bottleneck to $\mathrm{B}[Z, X, Y, D] = \mathrm{I}[Z; D \mid Y]$ which is related to the conditional version of the domain-adversarial neural network [36]. In practice, minimizing $\mathrm{I}[Z; D \mid Y]$ could be easier for optimization than $\mathrm{I}[Z; D]$, as it does not require to remove the information that D has about Y. In the following, we derive the conditional CAD (C^2AD) bottleneck using a similar idea as CAD.

How In this case, we want to minimize

$$\begin{aligned}
\mathrm{I}[Z; D \mid Y] &= \mathrm{H}[D \mid Y] - \mathrm{H}[D \mid Z, Y] && (138)\\
&= (const) - \mathrm{H}[D \mid Z, Y] && (139)\\
&\geq (const) - \mathbb{E}_{p_{D,Z,Y}}[-\log q(D \mid Z, Y)] && (140)
\end{aligned}$$

where $q(D \mid Z, Y)$ is a variational distribution of $p_{D \mid Z,Y}$. Similar to the unconditional case, we also aim to use a non-parametric approximation that is tied with other parts of the model, and we obtain it using the fact $p_{D \mid Z,Y} = \mathbb{E}_{p_{X \mid Z,Y}}\left[p_{D \mid X}\right]$. Specifically, let $\mathbf{Y} := \{Y, Y_1^-, \dots, Y_n^-\}$ be the collection of labels obtained by independently sampling the label from $p_{Y \mid X'}$ for each $X' \in \mathbf{X}$. We collect samples associated with the label Y, i.e., $\mathbf{X}_Y := \{X\} \cup \{X_i^- \mid Y_i^- = Y, i \in [n]\}$ and obtain a variational distribution of $p_{X \mid Z,Y}$:

$$q_{\varphi,\mathbf{X},\mathbf{Y}}(X \mid Z, Y) := \frac{\exp s_\varphi(X, Z)}{\sum_{X' \in \mathbf{X}_Y} \exp s_\varphi(X', Z)} \tag{141}$$

where we use the same critic $s_\varphi(X, Z) := e_\varphi(X)^T Z$ that is tied with the encoder e_φ as before, but only take softmax over those samples with the same label Y. For the term $p_{D \mid X}$, we use the same count estimate $\hat{p}_{\mathbf{D},\mathbf{X}}$ in Eq. (134). Then we obtain the variational distribution of $p_{D \mid Z,Y}$:

$$q_{\varphi,\mathbf{X},\mathbf{D},\mathbf{Y}}(D \mid Z, Y) = \sum_{X' \in \mathbf{X}_Y} q_{\varphi,\mathbf{X},\mathbf{Y}}(X' \mid Z, Y) \hat{p}_{\mathbf{D},\mathbf{X}}(D \mid X') \tag{142}$$

Putting all together we get that the final loss:

$$\mathcal{L}_{\text{C}^2\text{AD}}(\varphi,\psi) := \mathbb{E}_{p_{\mathbf{D},\mathbf{X},\mathbf{A},\mathbf{Y},Z}}\left[-\log q_{\psi,\mathbf{A}}(A^+ \mid Z) + \lambda \log\left(\sum_{X'\in\mathbf{X}_Y} q_{\varphi,\mathbf{X},\mathbf{Y}}(X' \mid Z,Y)\hat{p}_{\mathbf{D},\mathbf{X}}(D \mid X')\right)\right]. \tag{143}$$

Again, since in practice $\hat{p}_{\mathbf{D},\mathbf{X}}(D \mid X)$ is typically a dirac delta function, the summation in Eq. (142) can be done only over those associated with the same label Y *and* the same domain label D as X. In particular, we collect samples $\mathbf{X}_{Y,D} := \{X\} \cup \{X_i^- \mid Y_i^- = Y, D_i^- = D, i \in [n]\}$ and obtained the simplified loss:

$$\mathcal{L}_{\text{C}^2\text{AD}}(\varphi,\psi) := \mathbb{E}_{p_{\mathbf{D},\mathbf{X},\mathbf{A},\mathbf{Y},Z}}\left[-\log q_{\psi,\mathbf{A}}(A^+ \mid Z) + \lambda \log\left(\sum_{X'\in\mathbf{X}_{Y,D}} q_{\varphi,\mathbf{X},\mathbf{Y}}(X' \mid Z,Y)\right)\right]. \tag{144}$$

A detailed algorithm is in Algorithm 3.

Algorithm 3 conditional CAD (C^2AD) objective

Require: $e_\varphi, s_\psi, D, X, Y, n$

$Z \leftarrow e_\varphi(X)$

$A^+ \leftarrow \text{sample}(p_{A \mid X})$

$\left\{(D_i^-, X_i^-, A_i^-, Y_i^-)\right\}_{i=1}^n \xleftarrow{\text{i.i.d.}} \text{sample}(p_{D,X,A,Y})$

$\mathbf{X}, \mathbf{A} \leftarrow \{X\} \cup \left\{X_i^-\right\}_{i=1}^n, \{A^+\} \cup \left\{A_i^-\right\}_{i=1}^n$

$\mathbf{X}_Y \leftarrow \{X\} \cup \left\{X_i^- \mid Y_i^- = Y, i \in [n]\right\}$

$\mathbf{X}_{Y,D} \leftarrow \{X\} \cup \left\{X_i^- \mid Y_i^- = Y, D_i^- = D, i \in [n]\right\}$

$\mathcal{L}_{\text{aug}} \leftarrow -\log \frac{\exp s_\psi(A^+, Z)}{\sum_{A'\in\mathbf{A}} \exp s_\psi(A', Z)}$ $\quad \triangleright \mathrm{H}[A \mid Z]$

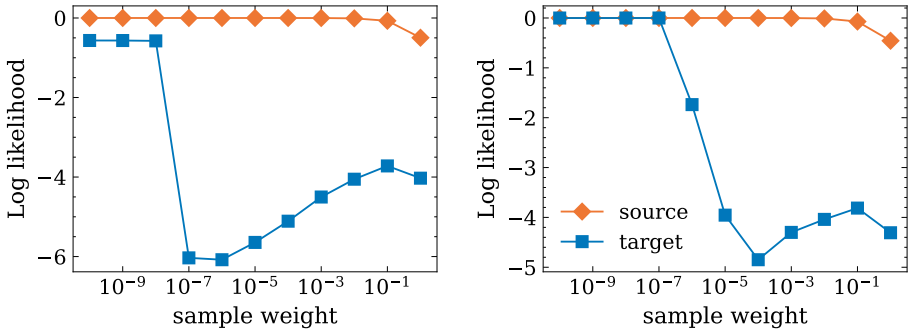
$\mathcal{L}_{\text{supp}} \leftarrow \log \frac{\sum_{X'\in\mathbf{X}_{Y,D}} \exp e_\varphi(X')^T Z}{\sum_{X''\in\mathbf{X}_Y} \exp e_\varphi(X'')^T Z}$ $\quad \triangleright \mathrm{I}[Z; D \mid Y]$

return $\mathcal{L}_{\text{C}^2\text{AD}} = \mathcal{L}_{\text{aug}} + \lambda \mathcal{L}_{\text{supp}}$

D Related work

Provably robust representations Z under covariate shift. Ben-David et al. [7] bounds the target risk using source risk and a divergence between source and target distributions. They do not consider representation learning, but in our setting, this implies that matching the marginal of Z while minimizing $\mathrm{R}[Y \mid Z]$ is sufficient for optimality. Ben-David et al. [8] suggests that $\mathrm{R}[Y \mid Z]$ is not sufficient, and Zhao et al. [58] also prove that one should minimize the joint $\mathrm{R}[Y \mid Z]$ instead of the source $\mathrm{R}^{d_s}[Y \mid Z]$ risk. Similarly, des Combes et al. [13] shows that matching the conditional $p_{Y \mid Z,d} = p_{Y \mid Z}$ is sufficient. Johansson et al. [27] take this further by proving that matching only the support of Z is also sufficient. Our work distinguishes itself from those and other related work on three key aspects: (i) We are the first to provide the set of *necessary* and sufficient conditions for robust representations; (ii) We prove that one can learn optimal Z^* with SSL using only large samples of inputs X and domain-covering augmentations A. (iii) We consider a general DG setting which deals with a less stringent *generalized* covariate shift and works for all standard losses and $\mathcal{Y}$ in ML. Still, our work is more specific than others, as we consider *idealized* DG and unrestricted predictors $\mathcal{H}$. Our theory could be combined with Dubois et al.'s [2020], who provide conditions for optimal generalization from finite samples and constrained $\mathcal{H}$ in supervised learning.

Practical objectives for DG. The most popular DG methods aim to learn domain-invariant representation by minimizing various divergernces between the conditionals $p_{Z \mid d}$ and marginals p_Z [37, 17, 49, 38, 35, 47, 39]. Others propose matching the conditional $p_{Z \mid y,d}$ across domains instead [19, 36, 50]. These regularizers would all be valid domain bottlenecks $\mathrm{B}[Z, D]$. Another line of work aims at learning Z with invariant predictors $p_{Y \mid z,d}$ across domains [e.g., 2, 31, 33]. However, none of these methods outperform ERM with fair model selections [22].



(a) CE-Base (b) SupCon-Base

Figure 3: Sweeping the sample weight using CE-Base and SupCon-Base. We selected 10^{-5} which seemed to be reasonble for both cases.

E Experimental Details

E.1 Scientific: exploring optimal representations for worst-case DG

In both the scientific setting and the following bridge setting, we consider rather unrealistic setups for verifying our theory where we have access to labels from all domains. We can choose to directly minimize the risk $\mathrm{R}[Y \mid Z]$ with the cross-entropy loss (denoted as CE henceafter), or minimize $\mathrm{H}[A \mid Z]$ (i.e., maximize $\mathrm{I}[A; Z]$) with supervised augmentations as in Fig. 2c detailed below.

Implementation of supervised augmentations When using supervised augmentations, for each sample we obtain its augmentations from within its label class across all domains. A constrastive loss with such augmentations will essentially reduce to the supervised contrast loss [SupCon, 28]. In particular, for a single sample in a batch, all samples in the batch with the same labels can be used as the positives (could come from the same domain or different domains) and others as the negatives. In Khosla et al. [28], two variants of SupCon loss were introduced for solving the issue of multi-positives depending on whether the summation over multi-positives was located inside (SupCon-In, Eq. (3) in Khosla et al. [28]) or outside (SupCon-Out, Eq. (2) in Khosla et al. [28]) the log. Though Khosla et al. [28] chose SupCon-Out because it worked better than SupCon-In, we hypothesized that this is because SupCon-Out has an implicit bottleneck effect. Intuitively, SupCon-Out upper bounds SupCon-In and achieves its optima only if the logits with positive samples are all the same by Jensen's inequality, which may encourage positive samples from different domains to get clustered. Since this might confound with the effect of our bottlenecks, we chose to use SupCon-In though it performed slightly worse in out initial experiments. For the implementation of SupCon, we followed Khosla et al. [28] except that no projection was used. Specifically, the temperature was set to 0.1, and normalization was applied when computing the logits.

In the scientific setting, we tried to simulate our theory to the greatest extent. In particular, we had two special considerations as detailed below:

Eliminating empirical generalization As our theory focuses on the idealized domain generalization that assumes access to population distribution, we considered the setup where the empirical generalization was eliminated. Specifically, we treated the dataset as the population distribution and used the same dataset for training the encoder and training/evaluating the predictor. The ResNet-18 encoder was trained to 300 epochs without any regularization, using the Adam optimizer [29] with a learning rate of 5e-5, a batch size of 192 (48 for each domain), and a cosine learning rate decay schedule.

Worst-case approximation To approximate the worst-case source predictor, we included the target data with randomly assigned *wrong* labels to the training set for training the source predictor. The target data samples were down-weighted with a sample weight that maximizes the target risk while keeping the source risk close to optima (which is 0). We selected the sample weight by sweeping over $[10^{-10}, 1]$ with a logarithmic scale using CE-Base and SupCon-Base, as shown in Fig. 3. As the sample weight increases, the target log likelihood (neg. risk) first decreases and then increases. We hypothesized that the increasing trend was due to that the source performance was already not optimal (though not visible from the figure), thus we selected the weight close to the turning point and 10^{-5}

seemed to be reasonable for both CE-Base and SupCon-Base. Although we did not adaptively select the sample weight for each setup due to the computational cost, the pre-specified sample turned out to be reasonable for all other losses and different lambda combinations. Furthermore, we also removed regularization when training the linear classifier and initialized the linear weight i.i.d. from $\mathcal{N}(0, 1)$.

Next, we provide other experimental details for reproducibility:

Implementation of standard augmentations We followed SimCLR [11] for implementing standard image augmentations. For a fair comparison between the cases when using standard augmentations (SimCLR) and supervised augmentations (SupCon), we kept the total batch size the same and also used the same configurations for computing the SupCon loss, i.e., temperature set to 0.1, no projection, and normalization applied.

Details of Fig. 4c In Fig. 4c, we considered different choices of augmentations. The 'Standard' augmentation implementation is described above (Appx. E.1). The 'Supervised' augmentation was essentially implemented using the SupCon loss as described in Appx. E.1. For other augmentations considered, we implemented them by *dropout* inter-domain supervised augmentations in SupCon. Specifically, for each sample in the batch, we randomly masked the samples from different domains (i.e., both inter-domain positives and negatives) i.i.d. with the specified dropout probability, while samples within the same domain were always kept. 'IntraDom' and 'ApproxDC' correspond to dropout probability 1 and 0.9, respectively. 'SingleDom' were implemented by dropout all inter-domain samples with probability 1 except for a fixed domain (the 'A' domain of PACS in our case).

E.2 Bridge: understanding how to learn optimal representations in practice

In the bridge setting (see Appx. F.2), we aimed to bridge the gap between our theoretical setup to the practical setup. The main differences from the scientific setups are that the empirical generalization gap is considered and the average-case source predictor is used, as detailed below:

Incorporating empirical generalization In practice, empirical-generalization gap should also be considered besides the source-target generalization gap. Thus, we randomly split the PACS dataset to 80% training and 20% validation splits for each domain. The training splits were used to train both the encoder and the source predictor, and the validation splits were used for encoder and source predictor selection as well as evaluation on target domains. We used the ResNet-50 model as the encoder and initialized it from ImageNet pretrained model. The encoder was trained to a maximum of 50 epochs with a 1e-5 weight decay, using the Adam optimizer [29] with a learning rate of 5e-5, a batch size of 112 (28 for each domain), and a cosine learning rate decay schedule.

Using average-case source predictor Instead of approximating the worst-case source predictor in the scientific setting, we considered the average-case[1] source predictor which is closer to the common practice. Specifically, we freezed the encoder and trained a SVM classifier with L2 regularization on the source training split. The regularization parameter was tuned over {1e-4, 1e-3, 1e-2, 1e-1, 1, 1e1, 1e2, 1e3} with the source validation accuracy.

Next, we provide other experimental details for reproducibility:

Selection of λ For all different setups considered in bridge settings, the CAD bottleneck was used and the λ was tuned over {1e-3, 1e-2, 1e-1, 1, 1e1} independently for each.

Staggered training setup We first trained model without any bottlenecks (i.e., with CE-Base) in the same way as described before. Then we freezed the trained model, added a 1-layer MLP with hidden size 1024 on top, and trained the model with CE-CAD using the same optimization procedure.

E.3 Approximating optimal representations with pretrained SSL

Datasets We used non-MNIST datasets on DomainBed that were non-synthetic, including VLCS [16], PACS [34], OfficeHome [54], TerraIncognita [6], and DomainNet [41]. For each dataset, we split it to 80%/20% training/validation set according to DomainBed.

[1] Here we have a slight abuse use of the phrase 'average-case' to distinguish from the 'worst-case' that we use in the scientific setting. In fact, the source predictor could be close to the 'best-case' since the max-margin classifier (SVM) was used.

SSL-based models & Training For all models based on pretrained SSL models (either CLIP-based or DINO-based) with staggered training in this experiment, we freezed the pretrained SSL model and added on top a 1-layer MLP with hidden size 1024, and residual connection. We used CLIP ResNet-50 (CLIP S) to obtain the best possible fair comparison with baselines from DomainBed, and CLIP ViT-B/32 (CLIP L) to achieve the best results. Note that the ResNet-50 model of CLIP S was modified as described in Radford et al. [43] and contained 38M parameters (more than 23M of the original CLIP). The model was trained to 300 epochs for DomainNet and 50 epochs on other datasets (an epoch is defined as a single pass over the smallest domain according to DomainBed). No data augmentation was used and the temperature for scaling the logits in CAD was fixed to 0.05. We used the Adam optimizer with a 1e-5 weight dacay, and a cosine learning rate decay schedule. The hyperparameter search space is:

- Learning rate: discrete set {1e-4, 3e-4, 1e-3, 3e-3}
- Batch size: discrete set {128, 256, 512} for DomainNet and OfficeHome, and {64, 128, 256} for other datasets
- MLP dropout: discrete set {0., 0.1, 0.5}
- Learning rate warmup: discrete set {True, False}

End-to-end models & Training In Table 4, we also included an end-to-end trained model without any pretrained SSL models. We used exactly the same model architecture (the original ResNet-50, initialized from ImageNet pretrained model), training procedure and evaluation protocal as baselines on DomainBed. Importantly, the linear classifier was jointly trained with the encoder, and no refitting was applied. The model was trained to a maximum of 5000 steps on each dataset, and data augmentations were applied. The Adam optimizer was used without any particular learning rate schedule. The hyperparameter search space is (same as DomainBed except we added the temperature):

- Learning rate: log-uniform over [1e-5, 1e-3.5]
- Batch size: log-uniform over [8, 64] for DomainNet, and [8, $2^{5.5}$] for other datasets
- MLP dropout: discrete set {0., 0.1, 0.5}
- Weight decay: log-uniform over [1e-6, 1e-2]
- Temperature: discrete set {0.05, 0.1}

Linear Probe Evaluation In all the experiments except for the end-to-end training setup, we always followed the procedure of two-stage training, where we first trained the encoder with specified objectives, and then *refit* the classifier. For datasets except DomainNet, we fitted the SVM classifier and tuned the regularization parameter over {1e-4, 1e-3, 1e-2, 1e-1, 1, 1e1, 1e2, 1e3} with source validation selection. Since DomainNet was too large and SVM cannot fit it efficiently, we used the logistic regression classifier which was trained with a batch size 512, the Adam optimizer with a learning rate 5e-4 and early stopping. Note that an alternative was to just use the linear head fitted when training the representor (as we used CE loss with source labels), and we found this could work better than refitting since the classifier was less overfitted to the source domain. However, we didn't do that since we wanted to stick to the representation learning protocol with two-stage training. We did that in our end-to-end training setup since we wanted it to be compeletely comparable to baselines on DomainBed (which did not do refitting).

Selection of λ In our experiments, we treated λ as a special hyperparamter. For each model, we selected λ on the PACS dataset, and then used the same λ value for all other datasets, because our bottleneck was fairly robust to the choice of λ. The λ values chosen for SSL-based models (i.e., CLIP S, CLIP L, DINO) and end-to-end ResNet-50 were 1e-1 and 1e-5, respectively.

E.4 Towards generic robust representations with SSL

Model We used the CLIP L model (i.e., CLIP ViT-B/32) with an additional network on top for staggered training. The additional network were two blocks of 2-layer MLP, each with hidden size 2048, pre-activation batch normalization, residual connection, and dropout probability 0.1. Note that the original CLIP L model was frozen and only the additional network was trained.

Dataset We used the LAION-400M dataset which contained 400 million image-text pairs for training. Though the dataset might not be as clean as the original CLIP training data (as evidenced by our experimental results), it was the largest publicly available image-text-pair dataset that we could get access to. As we froze the CLIP L model and only did staggered training, we used the 1TB preprocessed embeddings provided by LAION-400M[2]. No further preprocessing was applied.

Training We used the image-text contrastive loss as introduced in Radford et al. [43] for training model. The temperature was learnable which was initialized as 0.07 and clipped with a minimum 0.01. The model was trained for 1 epoch using the Adam optimizer with a batch size of 16384 and a cosine learning rate decay schedule. The learning rate was tuned over the set {3e-5, 1e-4, 3e-4, 1e-3, 3e-3, 1e-2} and the λ value for the Ent bottleneck was tuned over {1e-3, 1e-2, 1e-1, 1, 1e1}.

Evaluation For the evaluation on the ImageNet-related datasets, we followed a similar procedure in Radford et al. [43], where a linear classifier was fitted on ImageNet using the model representations and evaluated on 7 natural distribution shift datasets. In particular, we fitted a logistic regression classifier with 1e-5 L2 regularization on ImageNet training set which was trained with a batch size 512, the Adam optimizer with a learning rate 3e-4 and early stopping. Note that this was different from Radford et al. [43], where a logistic regression classifier was fitted using full-batch data with decent hyperparameter tuning, due to our computational budget. For evaluation on natural distribution shift datasets, we followed Taori et al. [51] and used their released testbed[3]. The evaluation datasets and their abbreviations used in Table 6 were: ImageNetV2 [IN-V2, 44], ImageNet-Sketch [IN-S, 55], Youtube-BB [YT-BB, 46], ImageNet-Vid [IN-Vid, 46], ObjectNet [5], ImageNet Adversarial [IN-A, 26], and ImageNet Rendition [IN-R 25].

F Additional Experimental Results

In our experiments, we aimed to: (i) verify our theoretical results in practice; (ii) investigate our proposed representation learning objectives in practical DG; (iii) take advantage of pretrained SSL models (in particular, CLIP) to achieve powerful models for DG. Unless stated otherwise, we consider a two-stage training setup. First, the representation learner ("the representor") trains an encoder $p_{Z \mid X}$ using a specified objective and freezes it. Then, the person performing predictions ("the learner") trains her predictor h from Z by minimizing the risk on source data. Finally, the representation Z and predictor h are evaluated on target data. In all experiments, the learner uses a linear classifier for h. For the Ent bottleneck, we used Ballé et al.'s (2018) entropy model. For the CAD bottleneck we used its conditional version whenever labels were available. When a model contains no domain bottleneck, we label it as "Base". For experimental details, see Appx. E.

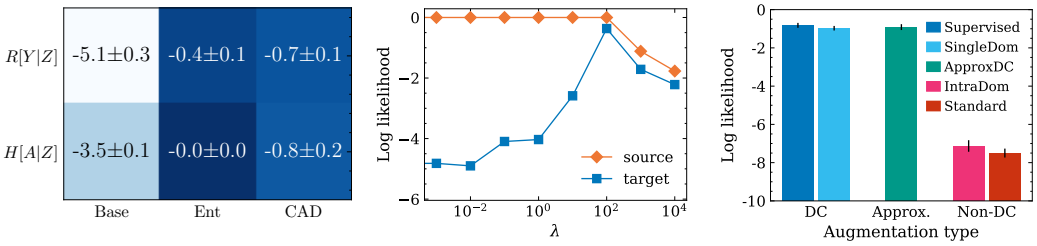
F.1 Scientific: exploring optimal representations for worst-case DG

To validate our theory, we studied optimal representations in a scientific setup that is as close to our IDG framework as possible with log-loss ℓ. In particular, we used the PACS dataset [34] and approximated the *idealized* DG by treating the dataset as the population distribution, i.e., we did not split datasets into train and test sets. To approximate the worst-case source predictor, we followed Dubois et al. [14] by incorporating the *wrongly labeled target* data to the source domain. The experimental setup goes as follows: (i) the representor trains a ResNet-18 [23] to minimize the objective on labeled data from all domains; (ii) the learner trains a *worst-case* source classifier h on every possible pair of (source, target); (iii) the negative target risk (log likelihood) for each h is evaluated. We reported the log likelihood averaged over 5 seeds. For more realistic scenarios (i.e. non-idealized average-case DG) see Appx. F.2 which replicates the following results.

Do our domain bottlenecks improve worst-case DG? In Fig. 4a, we compare IDG performance of representations trained with (Ent, CAD) and without (Base) domain bottlenecks. We see that both bottlenecks significantly improve the worst-case DG, and nearly achieve the source-domain performance (0 log likelihood). This shows the importance of support match (Thm. 2) and the effectiveness of our bottlenecks to enforce it. In Appx. F.2, we show that bottlenecks also helps in practical scenarios, i.e., non-idealized average-case DG evaluated with accuracy ($95.9\% \to 96.7\%$).

[2] See `https://laion.ai/laion-400-open-dataset/` for details.

[3] `https://github.com/modestyachts/imagenet-testbed`



(a) Effect of different objectives (b) Effect of λ (c) Effect of augmentations

Figure 4: (a) Adding bottlenecks significantly improves the worst-case DG performance and using domain-covering (DC) augmentations ($\mathrm{H}[A \mid Z]$) performs as well as with labels ($\mathrm{R}\,[Y \mid Z]$). (b) Increasing the domain bottleneck weight λ will improve target performance until it decreases source performance. (c) DC augmentations are crucial but approx. DC aug. might be also be sufficient.

What is the effect of λ? Fig. 4b shows the effect of the bottleneck weight λ on the worst-case target and source performance. We see that increasing λ will decrease the DG gap. As a result the target performance improves until $\lambda \approx 10^2$, where source performance starts to decrease.

What if the representor has access to domain-covering augmentations instead of labels? In Sec. 4, we provide a contrastive objective for using augmentations. To show the effectiveness of the objective, we compared minimizing $\mathrm{H}[A \mid Z]$ using Eq. (121) to standard supervised risk minimization $\mathrm{R}\,[Y \mid Z]$. We ensured domain coverage by using supervised augmentations (Fig. 2c). The 1st and 2nd row of Fig. 4a show that our objective performs similarly to direct label prediction.

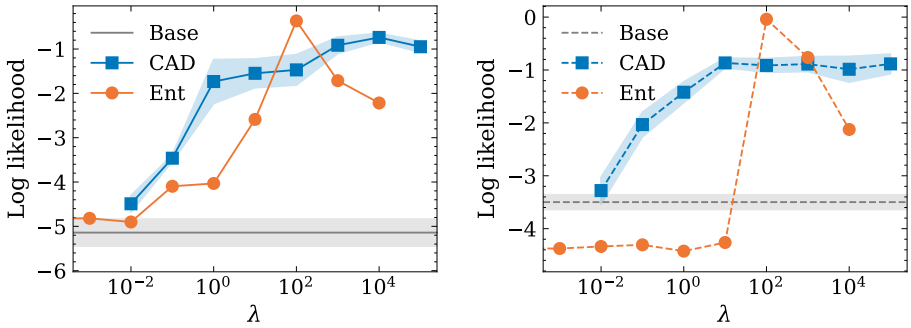
How important is the choice of augmentations? Prop. 2 shows that domain-covering (DC) augmentations are sufficient for achieving IDG, but it does not give necessary conditions. Here we investigate the effect of using our loss with different choices of augmentations. Specifically, we used $\mathcal{L}_{\text{CAD}}$ with five augmentations. The first two are DC. 'Supervised': augment inputs inside the label class across all domains as in Fig. 2c; 'SingleDom': augment inputs to same label samples from a fixed domain. The second two are not DC. 'Standard': standard SSL augmentations [11] as in Fig. 2b; 'IntraDom': augment inputs to same label and same domain samples. Finally, we consider 'ApproxDC', which is approximately DC by augmenting 10% of the time with 'Supervised' and 90% of the time with 'IntraDom'. Fig. 4c shows that the non-DC augmentations give terrible results compared to DC. Interestingly, 'ApproxDC' also performs very well, which suggests that approximately DC augmentations might be sufficient to learn optimal representations in practice.

What if the representor does not have access to target domains? Prop. 1 shows that DG without access to target domains is generally impossible. We empirically verified this by excluding a predefined target d_t domain from the representor's training set, i.e., $\mathcal{L}_{\text{CAD}}$ is optimized on 3 of the 4 domains. The learner then trains a predictor h on each source. We finally evaluate each h on the target domain d_t, and average over choices of d_t. The resulting worst-case log likelihood was -4.2 ± 0.2, which is significantly worse than when the representor had access to all domains (-0.8 ± 0.2).

What's the effect of λ for different objectives on the worst-case DG performance? In Fig. 5, the worst-case target log likelihood versus λ values for different objectives is shown. We found that Ent is much more sensitive to the choice of λ than CAD, which was part of the reason why we used the latter in most of our experiments. Note that for SupCon-Ent with small λ values, it was worse than SupCon-Base because of the discretization introduced by the Ent bottleneck, which we verified by observing that setting $\lambda = 0$ lead to similar results.

F.2 Bridge: understanding how to learn optimal representations in practice

The scientific setup is closer to our theory than what we do in practice in that worst-case predictor was considered and empirical generalization gap was ignored. Here we bridged these gaps with a more practical setup. In particular, we split the PACS dataset to training and validation splits for each domain and considered the setting: the representor trains the encoder on all-domain training splits with a validation loss selection; the learner trains the SVM predictor (average-case) on the source training split which is selected over the source validation split, and evaluates on the validation splits of other target domains. The target validation accuracy averaged over all (source, target) setups was reported. For simplicity, we will use CE to denote the objective with the cross-entropy loss that uses



(a) CE ($\mathrm{R}[Y \mid Z]$) objectives (b) SupCon ($\mathrm{H}[A \mid Z]$) objectives

Figure 5: The worst-case DG performance of Ent bottleneck is more sensitive to λ than CAD

labels to minimize $\mathrm{R}[Y \mid Z]$, and SupCon for the contrastive loss that uses supervised augmentations to minimize $\mathrm{H}[A \mid Z]$. We will use CAD in following experiments unless otherwise specified (chosen with initial experiments). Details in Appx. E.2.

Table 3: We repeated most empirical analysis (in the scientific setting) in the more practical bridge setting and observed similar results.

Setup	**Avg. target acc.**
CE-Base	95.9 ± 0.5
CE-CAD	96.7 ± 0.2
CE-CAD (partial domains)	82.6 ± 0.5
CE-CAD (staggered)	96.5 ± 0.0
SupCon-CAD	96.7 ± 0.4
SupCon-CAD (SingleDom)	96.7 ± 0.3
SupCon-CAD (ApproxDC)	96.6 ± 0.3
SupCon-CAD (IntraDom)	96.2 ± 0.7
SimCLR-CAD	61.7 ± 0.8

Does domain bottleneck improve the average-case DG performance? Though our theory focuses on the worst-case DG, we empirically showed that adding bottlenecks to enforce support match can also improve the average-case DG performance by comparing CE-Base and CE-CAD in Table 3.

What if the representor only has access to source domains? Similar to what we did in the scientific setting, we considered the setup where one single domain is specified as the target domain and excluded from the training set of the representor and used for evaluation with source predictors trained on other domains. This is denoted as CE+CAD (partial domains) in Table 3, which is much worse then CE-CAD. This shows the necessity of getting access to target domain information for DG.

What if the representor only has access to domain-covering augmentations? In Table 3, we also compared SupCon-CAD which used supervised augmentations through the labels with CE-CAD and they achieved the same performance. This shows that the representor can still learn good representations without labels but only domain-covering augmentations in practice.

Can we use standard augmentations? In Fig. 2, we point out that standard augmentations are not domain-covering and thus not suitable for SSL with our objectives. We empirically showed this by using augmentations of SimCLR (see Appx. E.1 for details) with our objectives (SimCLR-CAD). In Table 3, we indeed observed that using standard augmentations performed much worse than using desired augmentations (SupCon-CAD).

How do augmentations matter? Besides investigating the 'Supervised' augmentations (SupCon-CAD) and 'Standard' augmentations (SimCLR-CAD) above, we also compared other three augmentations as in the scientific section. Specifically, we considered the 'SingleDom', 'IntraDom', and 'ApproxDC' augmentations. As shown in Table 3, SupCon-CAD (SingleDom) and (ApproxDC)

maintained the DG performance but SupCon-CAD (IntraDom) was slightly worse (0.5 accuracy drop). We assumed the small gap was due to the specific dataset that we used (PACS). We did the same analysis on VLCS, and SupCon-CAD with 'Supervised', 'SingleDom', and 'IntraDom' augmentations gave 84.7 $\pm$ 0.4, 83.2 $\pm$ 0.3, and 77.5 $\pm$ 2.3, respectively. This shows the importance of using domain-covering augmentations in practice.

How is end-to-end training compared to staggered training? Since getting access to target domain data is not practical in practice, we can utilize SSL models pretrained on a very large support (as we did in practical and realistic settings). However, typical SSL models are trained without any bottlenecks and we can adopt the staggered training where we freeze the SSL models and train a small network on top with bottlenecks. To get an idea of how it compares to end-to-end training with bottlenecks, we considered a staggered version of CE-CAD in our setup. As shown in Table 3, it performed similarly to CE-CAD.

Do standard augmentations affect source performance? Previously, we showed that using standard augmentations hurt the DG performance measured by the average target accuracy. It is natural to ask whether using standard augmentations also hurt the source performance since we should also be interested in the 'effective robustness' [51]. Thus we also reported the average source accuracy of SupCon-CAD and SimCLR-CAD which were 96.9 $\pm$ 0.2 and 90.1 $\pm$ 0.2, respectively. The source performance using standard augmentations was indeed worse, but if we consider the source-target gap which was 0.2 for SupCon-CAD and 28.4 for SimCLR-CAD, which still verified that the non-domain-covering standard augmentations were harder to force support match. To be even more convincing, we did the same analysis on VLCS, and the average source accuracy of SupCon-CAD and SimCLR-CAD were 86.6 $\pm$ 0.1 and 84.6 $\pm$ 0.5 which were fairly close, but the average target accuracy were 84.7 $\pm$ 0.4 and 57.5 $\pm$ 1.7, respectively.

F.3 Approximating optimal representations with pretrained SSL

In this experiment, we used the standard DomainBed benchmark (with non-MNIST datasets) and protocol [22]. In particular, we left out a target domain for evaluation and used the union of other domains for training both the encoder and the classifier. Contrary to our scientific setting, the representor does not get access to the target domain. All our representations were evaluated by fitting a linear classifier on source domains with source validation selection. As in DomainBed we selected the encoder based on 'oracle selection' over 10 hyperparameters, and reported the target accuracy averaged over all choices of targets and 5 random seeds. Details in Appx. E.3.

We included the full result of Table 2 with all baselines on DomainBed as in Table 4. We considered most representative baselines from DomainBed, most of which considered learning invariant representations or optimal classifiers across domains. Specifically, we included IRM [2], GroupDRO [45], Mixup [57], CORAL [49], MMD [35], DANN [17], CDANN [36], and VREx [31]. We also included the result pretrained CLIP S model with a zero-shot classifier using text representations (CLIP S Zero Shot), which demonstrated better DG performance than CLIP S with linear probe. But we observed that it was outperformed by our CLIP S + CAD.

Can we approximate optimal representations by exploiting pretrained CLIP? The last row in Table 4 shows that finetuning a large pretrained CLIP model with our CAD achieves SOTA on nearly all DomainBed benchmarks by a very large margin (see 2nd row). Note that the poor performance on TerraIncognita is likely because CLIP's dataset did not cover such images (camera traps monitoring animals). In Appx. F.2, we estimated the non-idealized DG performance of optimal representations on PACS (with access to all-domain labeled data) to be 96.7%, which is only 2% higher than CLIP L + CAD. This suggests that our simple SSL encoder might already be close to optimal.

Are gains due to the architectural differences? DomainBed's baselines finetuned an ImageNet [12] pretrained ResNet-50. In contrast, CLIP L pretrained a larger ViT. To decouple gains due to our objective from architectural gains, we evaluated ResNet-50 pretrained. Table 4 shows that CLIP S still outperforms DomainBed baselines. Our theory does not constrain the encoder and so we expect larger encoders to be better. Table 4 shows that CLIP L indeed outperforms CLIP S.

What is the effect of domain bottlenecks? In the last six rows of Table 4, we investigated the effect of finetuning with our CAD bottleneck. We see that for both CLIP L and CLIP S, it improves results by around $1 \sim 2\%$. These gains are due to the bottleneck, rather than due to the additional MLP trained on source data as seen by 'CLIP S + Base'. Note that the raw CLIP S already significantly

Table 4: Finetuning CLIP with our CAD bottleneck to achieve SOTA performance on DomainBed with 'oracle' selection.

Algorithm	VLCS	PACS	OfficeHome	TerraIncognita	DomainNet
ERM	77.6 ± 0.3	86.7 ± 0.3	66.4 ± 0.5	53.0 ± 0.3	41.3 ± 0.1
IRM	76.9 ± 0.6	84.5 ± 1.1	63.0 ± 2.7	50.5 ± 0.7	28.0 ± 5.1
GroupDRO	77.4 ± 0.5	87.1 ± 0.1	66.2 ± 0.6	52.4 ± 0.1	33.4 ± 0.3
Mixup	78.1 ± 0.3	86.8 ± 0.3	68.0 ± 0.2	$\mathbf{54.4 \pm 0.3}$	39.6 ± 0.1
CORAL	77.7 ± 0.2	87.1 ± 0.5	68.4 ± 0.2	52.8 ± 0.2	41.8 ± 0.1
MMD	77.9 ± 0.1	87.2 ± 0.1	66.2 ± 0.3	52.0 ± 0.4	23.5 ± 9.4
DANN	79.7 ± 0.5	85.2 ± 0.2	65.3 ± 0.8	50.6 ± 0.4	38.3 ± 0.1
CDANN	79.9 ± 0.2	85.8 ± 0.8	65.3 ± 0.5	50.8 ± 0.6	38.5 ± 0.2
VREx	78.1 ± 0.2	87.2 ± 0.6	65.7 ± 0.3	51.4 ± 0.5	30.1 ± 3.7
CAD	77.3 ± 0.6	87.8 ± 0.5	67.9 ± 0.7	53.6 ± 1.4	41.7 ± 0.1
DINO + CAD	69.6 ± 0.6	76.1 ± 0.1	56.9 ± 0.5	25.9 ± 1.2	33.6 ± 0.1
CLIP S	81.1 ± 0.5	90.3 ± 0.2	70.6 ± 0.1	29.6 ± 0.8	47.7 ± 0.0
CLIP S (Zero-Shot)	80.9 ± 0.1	91.8 ± 0.1	70.4 ± 0.2	19.1 ± 0.1	46.9 ± 0.0
CLIP S + Base	81.6 ± 0.3	91.1 ± 0.3	70.6 ± 0.4	36.4 ± 0.7	46.7 ± 0.2
CLIP S + CAD	$\mathbf{82.2 \pm 0.3}$	92.4 ± 0.3	71.7 ± 0.6	36.1 ± 0.8	48.7 ± 0.1
CLIP L	80.7 ± 0.4	93.7 ± 0.8	79.9 ± 0.1	36.9 ± 0.6	52.8 ± 0.1
CLIP L + CAD	81.4 ± 0.8	$\mathbf{94.7 \pm 0.4}$	$\mathbf{80.2 \pm 0.2}$	39.7 ± 1.1	$\mathbf{54.1 \pm 0.1}$

outperforms baselines. We hypothesize that this could be because SGD training of neural networks favors support match, e.g., by minimizing $\mathrm{I}[X; Z]$ as suggested by Shwartz-Ziv & Tishby [48].

Which pretrained SSL model to use? Our theory suggests that we can exploit pretrained SSL models as long as their augmentations are domain-covering and their training set covers desired domains. We investigated the effect of adapting SSL models that do not satisfy those properties by finetuning DINO [10], the current SOTA on SSL ImageNet. DINO only pretraiend on ImageNet using standard augmentations. As a result, Table 4 shows that the finetuned DINO+CAD significantly underperforms compared to CLIP S and DomainBed baselines.

What is the impact of CLIP pretraining? To ensure that our gains are *not only* due to a novel CAD bottleneck, but the synergy between enforcing support constraint and using desired SSL models, we investigated CAD using the standard DomainBed protocol denoted as CAD in the table. It shows that CAD on its own performs similarly with DomainBed baselines (see Table 4 for a full comparison).

Why 'oracle' selection? In the main body, we provided the results with 'oracle selection' which was the closest to our theory among the model selection methods in DomainBed (in the sense that we needed target domain information to achieve IDG). Here, we also provided results with 'source validation' selection in Table 5. Source validation selection relies on the assumption that source and target data follow similar distributions [22] thus source and target accuracy are highly correlated, which is not really true in practice. We found some issues with source validation selection results:

- The selected model with the highest source validation accuracy tends to overfit the source domain, thus possibly leads to worse performance on the target domain. This can be probed by the fact that the staggered trained CLIP models (CLIP + Base or CLIP + CAD) were generally worse than the original CLIP model;
- Selecting model with source validation accuracy tends to diminish the effect of bottlenecks. This can be seen by the fact that the gap between CLIP + Base and CLIP + CAD of source validation selection is much smaller than that of oracle selection;
- The source accuracy is not a good indicator of target accuracy thus its result has a larger variance.

Table 5: Results on DomainBed with 'source validation' selection. Source validation selected model tends to overfit more to the source domain and diminish the effect of bottlenecks.

Algorithm	VLCS	PACS	OfficeHome	TerraIncognita	DomainNet
ERM	77.5 ± 0.4	85.5 ± 0.2	66.5 ± 0.3	46.1 ± 1.8	40.9 ± 0.1
IRM	78.5 ± 0.5	83.5 ± 0.8	64.3 ± 2.2	47.6 ± 0.8	33.9 ± 2.8
GroupDRO	76.7 ± 0.6	84.4 ± 0.8	66.0 ± 0.7	43.2 ± 1.1	33.3 ± 0.2
Mixup	77.4 ± 0.6	84.6 ± 0.6	68.1 ± 0.3	47.9 ± 0.8	39.2 ± 0.1
CORAL	78.8 ± 0.6	86.2 ± 0.3	68.7 ± 0.3	47.6 ± 1.0	41.5 ± 0.1
MMD	77.5 ± 0.9	84.6 ± 0.5	66.3 ± 0.1	42.2 ± 1.6	23.4 ± 9.5
DANN	78.6 ± 0.4	83.6 ± 0.4	65.9 ± 0.6	46.7 ± 0.5	38.3 ± 0.1
CDANN	77.5 ± 0.1	82.6 ± 0.9	65.8 ± 1.3	45.8 ± 1.6	38.3 ± 0.3
VREx	78.3 ± 0.2	84.9 ± 0.6	66.4 ± 0.6	46.4 ± 0.6	33.6 ± 2.9
CAD	77.7 ± 0.1	84.7 ± 1.0	68.5 ± 0.2	47.6 ± 1.7	41.4 ± 0.1
DINO + CAD	68.9 ± 0.9	75.4 ± 0.5	56.4 ± 0.7	23.6 ± 1.2	31.0 ± 2.3
CLIP S	81.1 ± 0.5	90.3 ± 0.2	70.6 ± 0.1	29.6 ± 0.8	47.7 ± 0.0
CLIP S Zero-Shot	80.9 ± 0.1	91.8 ± 0.1	70.4 ± 0.2	19.1 ± 0.1	46.9 ± 0.0
CLIP S + Base	81.0 ± 0.5	90.1 ± 0.3	70.4 ± 0.2	29.0 ± 1.4	44.7 ± 1.5
CLIP S + CAD	81.3 ± 0.3	90.0 ± 0.6	70.5 ± 0.2	29.4 ± 0.3	45.9 ± 2.1
CLIP L	80.7 ± 0.4	93.7 ± 0.8	79.9 ± 0.1	36.9 ± 0.6	52.8 ± 0.1
CLIP L + CE-CAD	80.5 ± 0.5	94.0 ± 0.6	79.8 ± 0.1	37.4 ± 1.2	52.3 ± 1.8

F.4 Towards generic robust representations with SSL

In the previous section, we finetuned CLIP in a task specific fashion by optimizing $\mathrm{R}\left[Y \mid Z\right]$ and our CAD bottleneck. To get generic (task agnostic) robust representations, one should instead directly use our objectives on a sufficiently large dataset with image-text augmentations. Unfortunately, we cannot fully train CLIP with our bottlenecks as we do not have access to CLIP's original dataset and sufficient compute. In this section, we aim to emulate such training of generic robust representations.

To do so we used LAION-400M [32] which is a public dataset that contains 400M web-crawled image-text pairs. Due to our computational budget, we again froze the pretrained CLIP L and only finetuned an additional MLP with our $\mathcal{L}_{\text{Ent}}$. We used $\mathcal{L}_{\text{Ent}}$ as it only requires access to paired image X and text A but no prior information about domain D. As in CLIP's paper, we evaluated the learned representation Z in Taori et al.'s (2020) realistic setting, where a linear classifier h from Z is trained on ImageNet and tested on 7 natural distribution shift datasets. Details in Appx. E.4.

Would training CLIP with a bottleneck have improved its robustness? As shown in the last 2 rows of Table 6, finetuning CLIP L on LAION with $\mathcal{L}_{\text{Ent}}$ (LAION + Ent) outperforms finetuning without bottleneck (LAION + Base) on all 7 distribution shift datasets. This suggests that directly training CLIP with our Ent bottleneck would improve the robustness of learned representations. We hypothesize that the gains could be larger if SSL models trained $\mathcal{L}_{\text{Ent}}$ end-to-end. In Table 7, we show similar results on DomainBed, where we followed exactly the same linear evaluation protocal discussed in Appx. E.3. Note that both models underperform the original CLIP L, likely due to non-end-to-end training and LAION data with (possibly) lower quality than CLIP's data.

Table 6: Finetuning CLIP L on LAION with an entropy bottleneck (LAION + Ent) improves its robustness compared to finetuning without (LAION + Base) on 7 distribution shift datasets. CLIP L is still better likely due to end-to-end training with higher quality data. IN denotes ImageNet.

	IN	IN-V2	IN-S	YT-BB	IN-Vid	ObjectNet	IN-A	IN-R	Avg.
CLIP L	75.2	64.2	41.0	58.4	71.6	42.8	27.5	62.9	52.6
LAION + Base	73.8	62.1	37.0	56.9	68.8	41.3	26.0	58.1	50.0
LAION + Ent	74.2	62.7	38.9	58.1	70.1	42.1	26.2	60.8	51.3

Table 7: Finetuning CLIP L on LAION with an entropy bottleneck (LAION + Ent) performs better on DomainBed than finetuning without (LAION + Base).

Algorithm	**VLCS**	**PACS**	**OfficeHome**	**TerraIncognita**	**DomainNet**
CLIP L	80.7 ± 0.4	93.7 ± 0.8	79.9 ± 0.1	36.9 ± 0.6	52.8 ± 0.1
LAION + Base	79.2 ± 0.7	93.4 ± 0.3	77.2 ± 0.5	36.1 ± 0.4	51.2 ± 0.1
LAION+ Ent	80.7 ± 0.4	94.3 ± 0.8	78.2 ± 0.2	36.8 ± 0.4	52.2 ± 0.1