Diversify and Disambiguate: Learning from Underspecified Data

Anonymous Authors[1]

Abstract

Many datasets are *underspecified*, meaning that there are several equally viable solutions to a given task. Underspecified datasets can be problematic for methods that learn a single hypothesis because different functions that achieve low training loss can focus on different predictive features and thus have widely varying predictions on out-of-distribution data. We propose DivDis, a simple two-stage framework that first learns a collection of diverse hypotheses for a task by leveraging unlabeled data from the test distribution. We then disambiguate by selecting one of the discovered hypotheses using minimal additional supervision, in the form of additional labels or inspection of function visualization. We demonstrate the ability of DivDis to find robust hypotheses in image classification and natural language processing problems with underspecification.

1. Introduction

Datasets are often *underspecified*: multiple plausible hypotheses each describe the data equally well (D'Amour et al., 2020), and the data offers no further evidence to prefer one over another. In the presence of such ambiguity, rigidly choosing a single "best" hypotheis can be suboptimal, causing failures when the data distribution is shifted. For example, examination of a chest X-ray dataset (Oakden-Rayner et al., 2020) has shown that many images of pneumothorax include a thin drain used to treat the disease. A classifier can thus erroneously identify such drains as a predictive feature of the disease, exhibiting degraded accuracy on the intended distribution of patients not yet being treated. To not suffer from such failures, it is desirable to have a model that can discover a diverse collection of alternate plausible hypotheses.

The standard empirical risk minimization (ERM, Vapnik (1992)) paradigm performs poorly when training on underspecified data. The main reason for such failures is that ERM tends to select the simplest solution without considering alternatives, using only the most salient features (Geirhos et al., 2020; Shah et al., 2020; Scimeca et al., 2021). Using an ensemble of ERM models (Hansen and Salamon, 1990; Lakshminarayanan et al., 2017) has this same problem because each model still suffers from simplicity bias. While many recent methods (Ganin et al., 2016; Sagawa et al., 2020; Liu et al., 2021) improve robustness in distribution shift settings, we find that they fail on data with more severe underspecification. This is because, similarly to ERM, these methods only consider a single solution even in situations where multiple explanations exist.

We propose Diversify and Disambiguate (DivDis), a two-stage framework for learning from underspecified data. Our key idea is to learn a collection of diverse functions that are consistent with training labels but disagree on unlabeled target data. DivDis is a single neural network consisting of a shared backbone feature extractor and multiple heads, each head representing a different function. As in regular training, each head is trained to predict labels for training data. An additional "diversification" loss trains the heads to make disagreeing predictions on a separate unlabeled dataset from the target distribution. At test time, we select one member of the diversified set of hypothesis by querying labels for the datapoints most informative for disambiguation. We visually outline this framework in Fig. 1. DivDis is well-suited for scenarios with underspecified data and distribution shift, and its heads will not yield a set of diverse functions in settings where only one function can achieve low training loss. We evaluate DivDis on several settings in which underspecification limits the performance of prior methods due to the existence of multiple solutions that achieve low predictive risk.

2. Diversify and Disambiguate

We now describe Diversify and Disambiguate (DivDis), a two-stage framework for learning from underspecified data. We first describe the general framework (Sec. 2.1), and then a specific implementation of the two DIVERSIFY (Sec. 2.2) and DISAMBIGUATE (Sec. 2.3) stages as used in our experiments.

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the SCIS workshop.

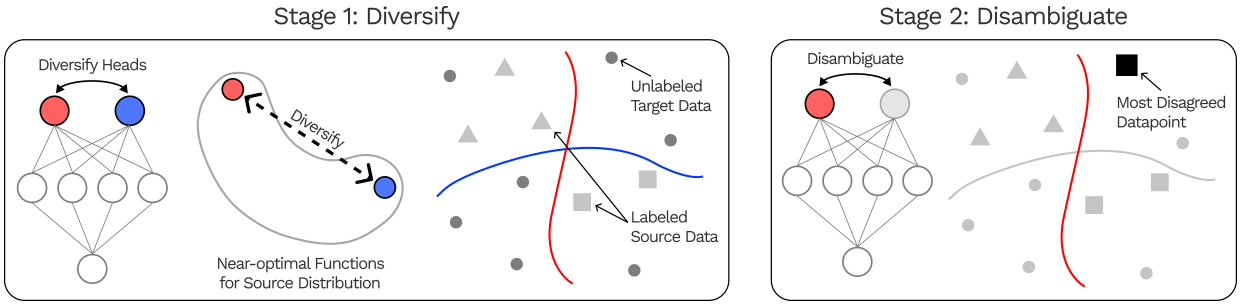


Figure 1. Our two-stage framework for learning from underspecified data. In the DIVERSIFY stage, we train each head in a multi-headed neural network to accurately predict the labels of source data while also outputting differing predictions for unlabeled target data. In the DISAMBIGUATE stage, we choose one of the heads by observing labels for an informative subset of the target data.

2.1. General Framework

As a running example to motivate our algorithm, consider an underspecified cow-camel image classification task in which the source data includes images of cows with grass backgrounds and camels with sand backgrounds. We can imagine two completely different classifiers each achieving perfect accuracy in the source distribution: one that classifies by animal and the other by background. The target dataset is most useful when it includes examples that would resolve the ambiguity if the labels were known, such as cows in the desert.

With this motivation, DivDis aims to first find a set of diverse functions and then choose the best member with minimal supervision. The DivDis framework consists of two stages. In the first stage, we DIVERSIFY by training a finite set of functions that together approximate the set of all functions that achieve low training loss on the source dataset. This stage uses both the source and target datasets for training. The source data ensures that all functions achieve low training loss, while the target data reveals whether or not the functions rely on different predictive features. In the second stage, we DISAMBIGUATE by choosing the best member among this set of functions, for example, by observing the label of a target datapoint for which one head is correct and the others are not.

2.2. DIVERSIFY : Train Disagreeing Heads

As described previously, the DIVERSIFY stage learns a diverse collection of functions by comparing predictions for the target set while minimizing training error. We use a labeled *source dataset* $\mathcal{D}_{\text{S}} = \{(x_1, y_1), \ldots\}$ along with an unlabeled *target dataset* $\mathcal{D}_{\text{T}} = \{x_1^t, \ldots\}$. We represent and train multiple functions using a multi-headed neural network with N heads. For an input datapoint x, we denote the prediction of head i as $f_i(x) = \widehat{y}_i$. We ensure that each head achieves low predictive risk on the source domain by minimizing the cross-entropy loss for each head $\mathcal{L}_{\text{xent}}(f_i) = \mathbb{E}_{x,y\sim\mathcal{D}_{\text{S}}}\left[l(f_i(x), y)\right]$.

We train the heads to produce predictions that are close to being statistically independent from each other. Concretely, we minimize the mutual information between each pair of predictions:

$$\mathcal{L}_{\text{MI}}(f_i, f_j) = D_{\text{KL}}\left(p(\widehat{y}_i, \widehat{y}_j) \,||\, p(\widehat{y}_i) \otimes p(\widehat{y}_j)\right), \quad (1)$$

where $D_{\text{KL}}\left(\cdot \,||\, \cdot\right)$ is the KL divergence and $\widehat{y}_i$ is the prediction $f_i(x)$ for $x \sim \mathcal{D}_{\text{T}}$. In practice, we optimize this quantity using empirical estimates of the distributions $p(\widehat{y}_i, \widehat{y}_j)$ and $p(\widehat{y}_i) \otimes p(\widehat{y}_j)$.

To prevent functions from collapsing to degenerate solutions such as predicting a single label for the entire target set while maintaining good source accuracy, we also include an optional regularization loss for each head which regularizes the marginal predicted label distribution on the target dataset:

$$\mathcal{L}_{\text{reg}}(f_i) = D_{\text{KL}}\left(p(\widehat{y}_i) \,||\, p(y)\right), \quad (2)$$

where $p(y)$ is the label distribution in the source dataset $\mathcal{D}_{\text{S}}$ without using an additional hyperparameter. The overall objective for the diversify stage is a linear combination with weight hyperparameters $\lambda_1, \lambda_2 \in \mathbb{R}$:

$$\sum_i \mathcal{L}_{\text{xent}}(f_i) + \lambda_1 \sum_{i\neq j} \mathcal{L}_{\text{MI}}(f_i, f_j) + \lambda_2 \sum_i \mathcal{L}_{\text{reg}}(f_i). \quad (3)$$

We note that the computation for (1) is easily parallelized in modern deep learning libraries; we provide an implementation in Appendix B. In practice, the cost of computing the objective (3) is dominated by the cost of feeding two batches—one source and one target—to the network. The time- and space- complexity of one step in the DIVERSIFY stage is approximately $\times 2$ compared to a standard SGD step in optimizing ERM with the source data, and both can be reduced by using a smaller batch size.

2.3. DISAMBIGUATE : Select the Best Head

Once we have learned a diverse set of functions that all achieve good training performance, we need to disambiguate by selecting one of the functions. This disambiguation stage requires information beyond the given source and

target datasets. For example, once our model has learned both the animal and background classifiers for the cow-camel task, we can quickly see which is right by asking for the ground-truth label of an image of a cow in the desert. We now present two different strategies for head selection during the DISAMBIGUATE stage.

Active querying. To select the best head with a minimal amount of supervision, we propose an active querying procedure, in which the model acquires labels for the most informative subset of the unlabeled target dataset $\mathcal{D}_{\text{T}}$. We sort each target datapoint $x \in \mathcal{D}_{\text{T}}$ according to the total distance between head predictions $\sum_{i \neq j} |f_i(x) - f_j(x)|$. We then select a subset of the target dataset, which has the m datapoints (i.e. $m \ll |\mathcal{D}_{\text{T}}|$) with the highest value of this metric. Finally, we select the head with the highest accuracy with respect to this labeled subset.

Random querying. A simple alternative to the active querying strategy is random querying, in which we label a random subset of the target dataset $\mathcal{D}_{\text{T}}$. Beyond its simplicity, an advantage of this procedure is that one can perform labeling in advance because the datapoints to be labeled do not depend on the results of the DIVERSIFY stage. However, random querying is substantially less label-efficient than active querying because the set will likely include unambiguous datapoints for which labels are less informative for head selection.

We emphasize that existing OOD methods tune hyperparameters using target set labels, and thus the active and random query strategies require no more information than previous approaches. Unless stated otherwise, we use the active querying strategy because of its superior label efficiency, requiring as little as a single label ($m = 1$). Additionally, as long as the best head is selected, the choice of disambiguation method only affects label efficiency, and the final performance does not change.

3. Experiments

Through our experimental evaluation, we aim to answer the following questions. (1) Can DivDis tackle image and language classification problems with severe underspecification, in which simplicity bias hinders the performance of existing approaches? (2) How sensitive is DivDis to hyperparameters, and what data assumptions are needed to tune DivDis? (3) How does DivDis compare to unsupervised domain adaptation algorithms, which also leverage unlabeled data from the target domain?

3.1. Tasks with Complete Spurious Correlation

We evaluate DivDis on datasets with a *complete correlation*, where the source distribution has a spuriously correlated attribute that can predict the label perfect accuracy.

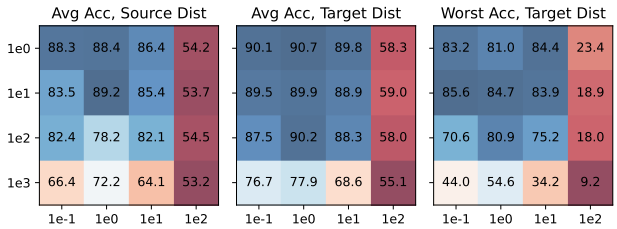


Figure 2. Hyperparameter grids for DivDis on the Waterbirds dataset. Rows and columns indicate λ_1 and λ_2, respectively.. We show three metrics: average accuracy on the source and target distributions and worst-group accuracy on the target distribution. The high correlation between the three metrics indicates that we can tune the hyperparameters of DivDis using only held-out labeled source data.

To make this problem tractable, we leverage unlabeled target data $\mathcal{D}_{\text{T}}$ for which the spurious attribute is not completely correlated with labels, as in the toy classification task (Fig. 3). Introducing complete correlations makes the problem considerably harder than existing subpopulation shift problems, because classifiers based on the spurious attribute can achieve perfect training loss. To our knowledge, complete correlation is not addressed by prior approaches and remains unsolved. In fact, in many of the problems we consider (Fig. 4a and Tab. 1) existing methods for subpopulation shift do not perform significantly better than random guessing.

Real data with complete correlation. We evaluate DivDis on existing benchmarks modified to exhibit complete correlation. Using the Waterbirds (Sagawa et al., 2020), CelebA (Liu et al., 2015), and MultiNLI (Gururangan et al., 2018) datasets, we alter the source dataset to include only majority groups while keeping target data intact. We denote these tasks as Waterbirds-CC, MultiNLI-CC, etc to distinguish from the original benchmarks. These -CC tasks are considerably more difficult than the original benchmarks, and introduce a specific challenge not addressed in the existing literature: leveraging the difference in source and target data distribution to encode and subsequently disambiguate tasks with high degrees of underspecification. To our best knowledge, no prior methods are designed to address such complete correlations.

As the closest existing problem setting is subpopulation shift, we show the performance of ERM, JTT (Liu et al., 2021), and Group DRO (Sagawa et al., 2020) as a naive point of comparison. We also include a random guessing baseline as a lower bound on performance. Quantitative results in Tab. 1 show that DivDis outperforms previous methods by 8% to 30% in worst-group accuracy. Prior methods for subpopulation shift showed subpar performance on these tasks: notably, ERM, JTT, and GDRO all fail to do better than random guessing in the Waterbirds-CC task. This is hardly surprising; methods based on loss upweight-

	Waterbirds-CC		CelebA-CC-1		CelebA-CC-2		MultiNLI-CC	
	Avg (%)	Worst (%)	Avg (%)	Worst (%)	Avg (%)	Worst (%)	Avg (%)	Worst (%)
Random	50.0	50.0	50.0	50.0	50.0	50.0	33.3	33.3
ERM	60.5 ± 1.6	7.0 ± 1.5	70.9 ± 2.0	57.0 ± 5.8	73.1 ± 0.9	41.1 ± 2.6	53.2 ± 1.5	22.8 ± 2.5
JTT (Liu et al., 2021)	44.6 ± 1.9	26.5 ± 1.4	71.4 ± 1.9	51.2 ± 5.4	78.7 ± 0.8	59.8 ± 1.1	80.0 ± 4.0	40.5 ± 2.3
GDRO (Sagawa et al., 2020)	55.6 ± 4.8	47.1 ± 8.9	71.6 ± 0.3	59.3 ± 2.6	71.6 ± 2.4	61.3 ± 2.3	79.1 ± 3.4	39.8 ± 1.4
DivDis - reg	87.2 ± 0.8	77.5 ± 4.7	91.0 ± 0.4	$\mathbf{85.9} \pm 1.0$	79.7 ± 0.4	$\mathbf{69.3} \pm 1.9$	80.3 ± 0.6	67.6 ± 4.0
DivDis	87.6 ± 1.4	$\mathbf{82.4} \pm 1.9$	90.8 ± 0.4	$\mathbf{85.6} \pm 1.1$	79.5 ± 0.2	$\mathbf{68.5} \pm 1.7$	79.9 ± 1.2	$\mathbf{71.5} \pm 2.5$

Table 1. Modified Waterbirds, CelebA, and MultiNLI datasets with complete correlation between labels and a spurious attribute. DivDis outperforms previous methods in terms of both average and worst-group accuracy.

	Waterbirds Worst Acc		CelebA Worst Acc	
Tuned with:	Worst	Average	Worst	Average
CVaR DRO (Levy et al., 2020)	75.9%	62.0%	64.4%	36.1%
LfF (Nam et al., 2020)	78.0%	44.1%	77.2%	24.4%
JTT (Liu et al., 2021)	86.7%	62.5%	81.1%	40.6%
DivDis	85.6%	**81.0**%	55.0%	**55.0**%

Table 2. Worst-group test accuracies in the Waterbirds and CelebA tasks, when tuning hyperparameters with respect to average and worst-group accuracies. DivDis is substantially more robust to hyperparameter choice in both tasks, allowing us to tune hyperparameters without group labels.

	Test Acc
Pseudo-Label (Lee et al., 2013)	67.7 ± 8.2
DANN (Ganin et al., 2016)	68.4 ± 9.2
FixMatch (Sohn et al., 2020)	71.0 ± 4.9
CORAL (Sun et al., 2016)	77.9 ± 6.6
NoisyStudent (Xie et al., 2019)	86.7 ± 1.7
DivDis (ours)	$\mathbf{90.4} \pm 1.8$

Table 3. Accuracy on the OOD test set of Camelyon17-WILDS. All methods in this table leverage unlabeled target data, and DivDis shows the best accuracy.

ing such as JTT and Group DRO are expected to perform poorly in these problems, because -CC tasks violate their implicit assumption of having minority points in the source data to upweight. In contrast, DivDis is well-suited to this challenging setting, and deals with complete correlation by leveraging unlabeled target data to find different predictive features of the labels.

3.2. Underspecification from Distribution Shift

Do we need group labels for hyperparameter tuning? Existing methods for learning from data with subpopulation shift (Levy et al., 2020; Nam et al., 2020; Liu et al., 2021) typically tune hyperparameters using group label annotations, making them only deployable in scenarios where group labels are available. To examine the dataset assumptions required to successfully tune DivDis's hyperparameters, we ran a hyperparameter sweep over (λ_1, λ_2). We measured three metrics using held-out data: (1) average accuracy on $\mathcal{D}_S$, (2) average accuracy on $\mathcal{D}_T$, and (3) worst-group accuracy on $\mathcal{D}_T$.

We see a clear correlation between these three metrics on the Waterbirds and CelebA datasets (Fig. 2, Fig. 9). Notably, we see that tuning the hyperparameters of DivDis with respect to average accuracy on $\mathcal{D}_S$ yields close an optimal model for worst-group accuracy on the target distribution. In Tab. 2, we contrast DivDis to existing methods reported by Liu et al. (2021). Note that this table corresponds to our second weakest data assumption (labeled target data), and Fig. 2 suggests that the hyperparameters of DivDis can even be tuned using labeled source data, which is readily available in all supervised learning problems. This experiment demonstrates that compared to previous methods for distribution shift, DivDis's hyperparameters require substantially less information to tune.

Comparison with unsupervised domain adaptation methods. Finally, we evaluate DivDis on the Camelyon17-WILDS benchmark (Sagawa et al., 2022), a tumor classification dataset where the objective is to generalize to images collected from a new hospital. This benchmark provides unlabeled data from the test domain hospitals, which DivDis can use during the DIVERSIFY stage. We compare against several approaches that can also leverage this unlabeled data: Pseudo-Label (Lee et al., 2013), FixMatch (Sohn et al., 2020), CORAL (Sun et al., 2016), and NoisyStudent (Xie et al., 2019). Quantitative results in Tab. 3 show that DivDis outperforms these methods, achieving above 90% OOD test set accuracy. This experiment demonstrates that DivDis can effectively leverage unlabeled data for underspecification arising from variation in real-world data collection conditions.

4. Conclusion

We proposed Diversify and Disambiguate (DivDis), a two-stage framework for learning from underspecified data. Our experiments show that DivDis has substantially higher performance when learning from datasets with high degrees of underspecification, at the modest cost of unlabeled target data and a few corresponding labels. To our knowledge, our method is the first to address this problem setting in the context of underspecification.

References

Arjovsky, M., Bottou, L., Gulrajani, I., and Lopez-Paz, D. (2019). Invariant risk minimization. *arXiv preprint arXiv:1907.02893*.

Arpit, D., Jastrzębski, S., Ballas, N., Krueger, D., Bengio, E., Kanwal, M. S., Maharaj, T., Fischer, A., Courville, A., Bengio, Y., et al. (2017). A closer look at memorization in deep networks. In *International Conference on Machine Learning*.

Cohn, D. A., Ghahramani, Z., and Jordan, M. I. (1996). Active learning with statistical models. *Journal of artificial intelligence research*, 4:129–145.

Creager, E., Jacobsen, J.-H., and Zemel, R. (2021). Environment inference for invariant learning. In *International Conference on Machine Learning*.

D'Amour, A., Heller, K., Moldovan, D., Adlam, B., Alipanahi, B., Beutel, A., Chen, C., Deaton, J., Eisenstein, J., Hoffman, M. D., et al. (2020). Underspecification presents challenges for credibility in modern machine learning. *arXiv preprint arXiv:2011.03395*.

Dietterich, T. G. (2000). Ensemble methods in machine learning. In *International workshop on multiple classifier systems*, pages 1–15. Springer.

Fisher, A., Rudin, C., and Dominici, F. (2019). All models are wrong, but many are useful: Learning a variable's importance by studying an entire class of prediction models simultaneously. *J. Mach. Learn. Res.*, 20(177):1–81.

Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., Marchand, M., and Lempitsky, V. (2016). Domain-adversarial training of neural networks. *The journal of machine learning research*, 17(1):2096–2030.

Geirhos, R., Jacobsen, J.-H., Michaelis, C., Zemel, R., Brendel, W., Bethge, M., and Wichmann, F. A. (2020). Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2(11):665–673.

Goudet, O., Kalainathan, D., Caillou, P., Guyon, I., Lopez-Paz, D., and Sebag, M. (2018). Learning functional causal models with generative neural networks. In *Explainable and interpretable models in computer vision and machine learning*, pages 39–80. Springer.

Gunasekar, S., Lee, J. D., Soudry, D., and Srebro, N. (2018). Implicit bias of gradient descent on linear convolutional networks. In Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R., editors, *Advances in Neural Information Processing Systems*.

Gururangan, S., Swayamdipta, S., Levy, O., Schwartz, R., Bowman, S. R., and Smith, N. A. (2018). Annotation artifacts in natural language inference data. *arXiv preprint arXiv:1803.02324*.

Hanneke, S. et al. (2014). Theory of disagreement-based active learning. *Foundations and Trends® in Machine Learning*, 7(2-3):131–309.

Hansen, L. K. and Salamon, P. (1990). Neural network ensembles. *IEEE transactions on pattern analysis and machine intelligence*, 12(10):993–1001.

Houlsby, N., Huszár, F., Ghahramani, Z., and Lengyel, M. (2011). Bayesian active learning for classification and preference learning. *arXiv preprint arXiv:1112.5745*.

Ke, N. R., Bilaniuk, O., Goyal, A., Bauer, S., Larochelle, H., Schölkopf, B., Mozer, M. C., Pal, C., and Bengio, Y. (2019). Learning neural causal models from unknown interventions. *arXiv preprint arXiv:1910.01075*.

Krogh, A., Vedelsby, J., et al. (1995). Neural network ensembles, cross validation, and active learning. *Advances in neural information processing systems*.

Lakshminarayanan, B., Pritzel, A., and Blundell, C. (2017). Simple and scalable predictive uncertainty estimation using deep ensembles. *Conference on Neural Information Processing Systems*.

Lee, D.-H. et al. (2013). Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on challenges in representation learning, ICML*.

Levy, D., Carmon, Y., Duchi, J. C., and Sidford, A. (2020). Large-scale methods for distributionally robust optimization. In *Advances in Neural Information Processing Systems*.

Liu, E. Z., Haghgoo, B., Chen, A. S., Raghunathan, A., Koh, P. W., Sagawa, S., Liang, P., and Finn, C. (2021). Just train twice: Improving group robustness without training group information. In *International Conference on Machine Learning*, pages 6781–6792. PMLR.

Liu, Z., Luo, P., Wang, X., and Tang, X. (2015). Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision*.

Louizos, C., Shalit, U., Mooij, J., Sontag, D., Zemel, R., and Welling, M. (2017). Causal effect inference with deep latent-variable models. *Conference on Neural Information Processing Systems*.

Nam, J., Cha, H., Ahn, S., Lee, J., and Shin, J. (2020). Learning from failure: Training debiased classifier from

biased classifier. *Conference on Neural Information Processing Systems*.

Oakden-Rayner, L., Dunnmon, J., Carneiro, G., and Ré, C. (2020). Hidden stratification causes clinically meaningful failures in machine learning for medical imaging. In *Proceedings of the ACM conference on health, inference, and learning*, pages 151–159.

Pagliardini, M., Jaggi, M., Fleuret, F., and Karimireddy, S. P. (2022). Agree to disagree: Diversity through disagreement for better transferability. *arXiv preprint arXiv:2202.04414*.

Pang, T., Xu, K., Du, C., Chen, N., and Zhu, J. (2019). Improving adversarial robustness via promoting ensemble diversity. In *International Conference on Machine Learning*.

Parker-Holder, J., Metz, L., Resnick, C., Hu, H., Lerer, A., Letcher, A., Peysakhovich, A., Pacchiano, A., and Foerster, J. (2020). Ridge rider: Finding diverse solutions by following eigenvectors of the hessian. *Conference on Neural Information Processing Systems*.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., and Chintala, S. (2019). Pytorch: An imperative style, high-performance deep learning library. *Conference on Neural Information Processing Systems*.

Pearl, J. (2000). *Causality: models, reasoning and inference*, volume 19. Cambridge university press.

Pezeshki, M., Kaba, S.-O., Bengio, Y., Courville, A., Precup, D., and Lajoie, G. (2021). Gradient starvation: A learning proclivity in neural networks. In Beygelzimer, A., Dauphin, Y., Liang, P., and Vaughan, J. W., editors, *Advances in Neural Information Processing Systems*.

Rame, A. and Cord, M. (2021). Dice: Diversity in deep ensembles via conditional redundancy adversarial estimation. *International Conference on Learning Representations*.

Rogozhnikov, A. (2022). Einops: Clear and reliable tensor manipulations with einstein-like notation. In *International Conference on Learning Representations*.

Sagawa, S., Koh, P. W., Hashimoto, T. B., and Liang, P. (2020). Distributionally robust neural networks for group shifts: On the importance of regularization for worst-case generalization. *International Conference on Learning Representations*.

Sagawa, S., Koh, P. W., Lee, T., Gao, I., Xie, S. M., Shen, K., Kumar, A., Hu, W., Yasunaga, M., Marklund, H., Beery, S., David, E., Stavness, I., Guo, W., Leskovec, J., Saenko, K., Hashimoto, T., Levine, S., Finn, C., and Liang, P. (2022). Extending the WILDS benchmark for unsupervised adaptation. In *International Conference on Learning Representations*.

Schölkopf, B. (2019). Causality for machine learning. *arXiv preprint arXiv:1911.10500*.

Scimeca, L., Oh, S. J., Chun, S., Poli, M., and Yun, S. (2021). Which shortcut cues will dnns choose? a study from the parameter-space perspective. *arXiv preprint arXiv:2110.03095*.

Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., and Batra, D. (2017). Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pages 618–626.

Semenova, L., Rudin, C., and Parr, R. (2019). A study in rashomon curves and volumes: A new perspective on generalization and model simplicity in machine learning. *arXiv preprint arXiv:1908.01755*.

Shah, H., Tamuly, K., Raghunathan, A., Jain, P., and Netrapalli, P. (2020). The pitfalls of simplicity bias in neural networks. *Conference on Neural Information Processing Systems*.

Sinha, S., Bharadhwaj, H., Goyal, A., Larochelle, H., Garg, A., and Shkurti, F. (2021). Dibs: Diversity inducing information bottleneck in model ensembles. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

Sohn, K., Berthelot, D., Li, C.-L., Zhang, Z., Carlini, N., Cubuk, E. D., Kurakin, A., Zhang, H., and Raffel, C. (2020). Fixmatch: Simplifying semi-supervised learning with consistency and confidence. *Advances in Neural Information Processing Systems*.

Sun, B., Feng, J., and Saenko, K. (2016). Correlation alignment for unsupervised domain adaptation. *arXiv preprint arXiv: Arxiv-1612.01939*.

Teney, D., Abbasnejad, E., Lucey, S., and Hengel, A. v. d. (2021). Evading the simplicity bias: Training a diverse set of models discovers solutions with superior ood generalization. *arXiv preprint arXiv:2105.05612*.

Tzeng, E., Hoffman, J., Zhang, N., Saenko, K., and Darrell, T. (2014). Deep domain confusion: Maximizing for domain invariance. *arXiv preprint arXiv:1412.3474*.

Vapnik, V. (1992). Principles of risk minimization for learning theory. In *Advances in neural information processing systems*, pages 831–838.

Wang, X., Peng, Y., Lu, L., Lu, Z., Bagheri, M., and Summers, R. M. (2017). Chestx-ray8: Hospital-scale chest x-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2097–2106.

Wortsman, M., Horton, M., Guestrin, C., Farhadi, A., and Rastegari, M. (2021). Learning neural network subspaces. *International Conference on Machine Learning*.

Xie, Q., Luong, M.-T., Hovy, E., and Le, Q. V. (2019). Self-training with noisy student improves imagenet classification. *Conference on Computer Vision and Pattern Recognition*.

Zhou, B., Lapedriza, A., Khosla, A., Oliva, A., and Torralba, A. (2017). Places: A 10 million image database for scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

A. Related Work

Underspecification. Prior works have discussed the underspecified nature of many datasets (D'Amour et al., 2020; Oakden-Rayner et al., 2020). Underspecification is especially problematic when the bias of deep neural networks towards simple functions (Arpit et al., 2017; Gunasekar et al., 2018; Shah et al., 2020; Geirhos et al., 2020; Pezeshki et al., 2021) is not aligned with the true function. Yet, these works do not present a general solution. As we find in Section 3, DivDis can address underspecified datasets, even when one viable solution is much simpler than another, since only one of the heads can represent the simplest solution. Our notion of near-optimal sets can be seen as an extension of *Rashomon sets* (Fisher et al., 2019; Semenova et al., 2019) to the unsupervised domain adaptation setting. Active learning methods (Cohn et al., 1996; Hanneke et al., 2014) are also related in that they handle underspecification by reducing ambiguity. Our MI-based diversity term resembles a common active learning criterion (Houlsby et al., 2011), but a key difference is that we directly optimize a set of models with respect to our criterion.

Ensemble methods. Our approach is related to ensemble methods (Hansen and Salamon, 1990; Dietterich, 2000; Lakshminarayanan et al., 2017), which aggregate the predictions of multiple learners. Ensembles have been shown to perform best when each member produces errors independently of one another (Krogh et al., 1995), a property we exploit by maximizing disagreement on unlabeled test data. Previous works have extended ensembles by learning a diversified set of functions (Pang et al., 2019; Parker-Holder et al., 2020; Wortsman et al., 2021; Rame and Cord, 2021; Sinha et al., 2021). While the DIVERSIFY stage similarly learns a collection of diverse functions, our approach differs in that we directly optimize for diversity on a separate target dataset. Two recent works leverage unlabeled target data to learn a set of diverse functions. Teney et al. (2021) introduce a gradient orthogonality constraint with respect to features from a pre-trained backbone. However, this approach does not consider classifier selection (i.e. the DISAMBIGUATE stage), relying on an oracle instead, and requires sufficiently compact pre-trained features. Concurrently to our work, Pagliardini et al. (2022) propose to sequentially train a set of functions with a diversity loss on target data. In contrast, DivDis requires a single network and training loop regardless of the number of heads. Furthermore, Sec. 3 demonstrates that DivDis scales to larger datasets.

Robustness and causality. Many recent methods aim to produce robust models that succeed even in conditions of distribution shift (Tzeng et al., 2014; Ganin et al., 2016; Arjovsky et al., 2019; Sagawa et al., 2020; Nam et al., 2020; Creager et al., 2021; Liu et al., 2021). While our work is similarly motivated, we address a class of problems that these previous methods fundamentally cannot handle. By nature of learning only one function, these robustness methods cannot disambiguate problems where the true function is truly ambiguous, in the sense that functions based on two different features can both be near-optimal. DivDis handles such scenarios by learning multiple functions in the DIVERSIFY stage and then choosing the correct one in the DISAMBIGUATE stage with minimal added supervision. This research direction is also related to inferring the causal structure (Pearl, 2000; Schölkopf, 2019) of observed attributes. Although many causality works focus on situations in which interventions are impossible, we explore inherently ambiguous problems where some form of intervention is necessary to succeed. Additionally, recent methods for extracting causality from observational data have been most successful in low-dimensional settings (Louizos et al., 2017; Goudet et al., 2018; Ke et al., 2019), whereas our method easily scales to large convolutional networks for image classification problems.

B. Parallel Implementation of Mutual Information Objective

```python
import torch
from einops import rearrange

def mutual_info_loss(probs):
    """ Input: predicted probabilites on target batch. """
    B, H, D = probs.shape  # B=batch_size, H=heads, D=pred_dim
    marginal_p = probs.mean(dim=0)  # H, D
    marginal_p = torch.einsum("hd,ge->hgde", marginal_p, marginal_p)  # H, H, D, D
    marginal_p = rearrange(marginal_p, "h g d e -> (h g) (d e)")  # H^2, D^2

    joint_p = torch.einsum("bhd,bge->bhgde", probs, probs).mean(dim=0)  # H, H, D, D
    joint_p = rearrange(joint_p, "h g d e -> (h g) (d e)")  # H^2, D^2

    kl_divs = joint_p * (joint_p.log() - marginal_p.log())
    kl_grid = rearrange(kl_divs.sum(dim=-1), "(h g) -> h g", h=H)  # H, H
    pairwise_mis = torch.triu(kl_grid, diagonal=1) # Get only off-diagonal KL divergences
```

```
    return pairwise_mis.mean()
```

This implementation is based on the PyTorch (Paszke et al., 2019) and einops (Rogozhnikov, 2022) libraries. It demonstrates that the mutual information objective (1) is easily parallelized across the input batch using standard tensor operations.

C. Experimental Setup

C.1. Detailed Dataset Descriptions

Toy classification task. Our toy binary classification data is constructed as follows. The source dataset $\mathcal{D}_{\text{S}}$ has binary labels with equal aggregate probability $p(y=0) = p(y=1) = \frac{1}{2}$. Each datapoint is a 2-dimensional vector, and the data distribution for each class in the source dataset is:

$$p(x \mid y=0) = \text{Unif}([-1,0] \times [0,1])$$
$$p(x \mid y=1) = \text{Unif}([0,1] \times [-1,0]).$$

In contrast, the data distribution for each class in the target dataset is:

$$p(x \mid y=0) = \text{Unif}([-1,0] \times [-1,1])$$
$$p(x \mid y=1) = \text{Unif}([0,1] \times [-1,1]).$$

Labels are balanced for the target dataset. Put differently, the target dataset has a larger span than the source dataset, and the labels of the target dataset reveal that the true decision boundary is the Y-axis.

CXR-14 pneumothorax classification. The CXR-14 dataset (Wang et al., 2017) is a large-scale dataset for pathology detection in chest radiographs. We evaluate on the binary pneumothorax classification task, which has been reported to suffer from hidden stratification: a subset of the images with the disease include a chest drain, a common treatment for the condition (Oakden-Rayner et al., 2020).

Waterbirds dataset. Each image in the Waterbirds dataset is constructed by pasting a waterbird or landbird image to a background drawn from the Places dataset (Zhou et al., 2017). There are two backgrounds in this dataset – water and land, where each category of birds is spuriously correlated with one background. Specifically, there are 4,795 training samples, where 3,498 samples are from "waterbirds in water" and 1,057 samples are from "landbirds in land". "Waterbirds in land" and "landbirds in water" are considered as minority groups, where 184 and 56 samples are included, respectively.

CelebA dataset. The CelebA dataset (Liu et al., 2015) is a large-scale image dataset with over $200,000$ images of celebrities, each with 40 attribute annotations. We construct four different completely correlated problem settings, each based on a pair of attributes. The pair of attributes consists of a label attribute and a spurious attribute, and we remove all examples from the two minority groups in the source dataset. The four problem settings are summarized below. Our task construction is similar to that of Sagawa et al. (2020), which uses hair color as the label and gender as the spurious attribute.

	Label attribute	Spurious attribute
CelebA-CC-1	Mouth_Slightly_Open	Wearing_Lipstick
CelebA-CC-2	Attractive	Smiling
CelebA-CC-3	Wavy_Hair	High_Cheekbones
CelebA-CC-4	Heavy_Makeup	Big_Lips

MultiNLI dataset. Given a hypothesis and a promise, the task of MultiNLI dataset is to predict if the hypothesis is entailed by, neutral with, or contradicts with the promise. The spurious correlation exists between contradictions and the presence of the negation words nobody, no, never, and nothing (Gururangan et al., 2018). The whole MultiNLI dataset is divided into six groups, where each spurious attribute belongs to {"no negation", "negation"} and each label belongs to {entailed, neutral, contradictory}. There are 206,175 samples in total, where the smallest group only has 1,521 samples (entailment with negations).

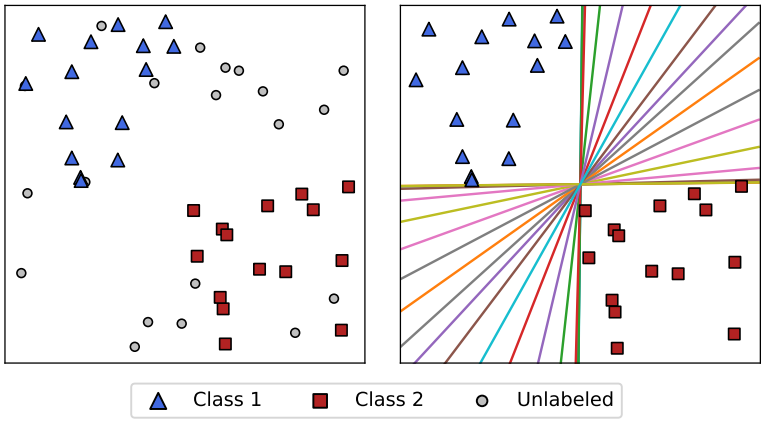


Figure 3. Left: synthetic 2D classification task with underspecification. Right: decision boundaries of 20 linear functions discovered by the DIVERSIFY stage. Together, these functions span the set of linear decision boundaries consistent with the data.

Camelyon17-WILDS dataset. This dataset is part of the U-WILDS benchmark (Sagawa et al., 2022). Input images of patches from lymph node sections are given, and the task is to classify as either a tumor or normal tissue. Evaluation is performed on OOD hospitals for which labels are unseen during training. The model is given unlabeled validation images from the OOD hospitals.

C.2. DivDis Hyperparameter Settings

We show below the hyperparameters used in our experiments:

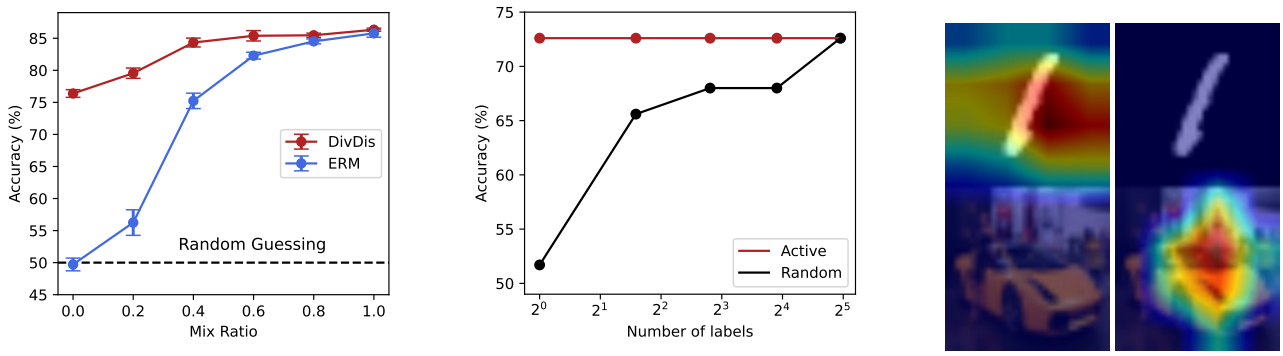
	Toy Clasification	MNIST-CIFAR	Waterbirds	CXR-14	Waterbirds-CC	CelebA-CC	MultiNLI-CC
N	$2, 20$	2	2	2	2	2	2
λ_1	10	10	$1, 10, 100, 1000$	10	10	10	1000
λ_2	10	10	$0.1, 1, 10, 100$	10	$0, 10$	$0, 10$	$0, 0.1$
m	1	1	16	16	16	16	16

D. Additional Experiments

2D classification task. We start with a synthetic 2D binary classification problem, with the goal of understanding the set of functions learned during the DIVERSIFY stage of DivDis. The task is shown in Fig. 3 (left): inputs are points in 2-dimensional space, and the source dataset has points only in the second and fourth quadrants, making the unlabeled points in the first and third quadrants ambiguous. We train a network with two heads and measure the target domain accuracy of each head throughout the DIVERSIFY stage of DivDis. Learning curves and decision boundaries, visualized in **??**, show that DivDis initially learns to fit the source data, after which the two heads diverge to functions based on different predictive features of the data. We show an extended visualization with additional metrics in the appendix (Fig. 5).

Coverage of near-optimal set on 2D classification. To further understand how well DivDis can cover the set of near-optimal functions, we trained a 20-head model on the same 2D classification task, where each head is a linear classifier. Results in Fig. 3 show that the heads together span the set of linear classifiers consistent with the source data. Note that this set includes the function learned by standard ERM, the diagonal decision boundary ($y = x$). This result suggests that given enough heads, the set of functions learned by DivDis sufficiently covers the set of near-optimal functions, including the simplest function typically learned by ERM.

Comparison with ensembles. We compare the diversity of the functions produced by the DIVERSIFY stage to that of independently trained models. On a 3-dimensional version of the binary classification task, we trained DivDis and ensembles with $\{2, 3, 5\}$ members. We measure how much each function relies on each of the three input dimensions through the Pearson correlation coefficient between each input dimension and the prediction, Due to space constraints, we show visualizations in Fig. 6 of Appendix D. Each of the functions learned by DivDis depend on different input



(a) Accuracy vs mix ratio. (b) Accuracy vs labels used. (c) Grad-CAM visualization of heads.

Figure 4. (a) Accuracy of ERM and DivDis on MNIST-CIFAR data. (b) Number of labels used for disambiguation with the active querying and random querying strategies. (c) Grad-CAM visualization of two learned heads on a randomly sampled datapoint from the MNIST-CIFAR source dataset. See Sec. 3.1 for details.

features, whereas independently trained models use all features equally. This experiment demonstrates that the diversity in a vanilla ensemble cannot effectively cover the set of near-optimal functions, and is therefore insufficient for underspecified problems.

Extended learning curves on toy task. In Fig. 5, we show an extended version of the learning curve shown in **??**. The extended plot includes cross-entropy loss and mutual information loss during training. The learning curves show that cross-entropy loss decreases first, at which point both of the heads represent functions similar to the ERM solution. Afterwards, the mutual information loss decreases, causing the functions represented by the two heads to diverge.

Visualization of functions on 3D toy task. We examine the extent to which the DIVERSIFY stage can produce different functions by visualizing which input dimension each function relies on. We modify the synthetic binary classification task to have 3-dimensional inputs and train DivDis with $\{2, 3, 5\}$ heads. For each head, we visualize the Pearson correlation coefficient between each input dimension and output. We normalize this 3-dimensional vector to sum to one and plot each model as a point on a 2-simplex in Fig. 6, with independently trained functions as a baseline. The results show that the DIVERSIFY stage acts as a repulsive force between the functions in function space, allowing the collection of heads to explore much closer to the vertices. This experiment also demonstrates why vanilla ensembling is insufficient for underspecified problems: the diversity due to random seed is not large enough to effectively cover the set of near-optimal functions.

Noisy dimension. To see if DivDis can effectively combat simplicity bias, we further evaluate on a harder variant of the 2D classification problem in which we add noise along the x-axis. This noise makes the "correct" decision boundary have positive non-zero risk, making it harder to learn than the other function. Results in Fig. 7 demonstrate that even in such a scenario, DivDis recovers both the x-axis and y-axis decision boundaries, suggesting that DivDis can be effective even in scenarios where ERM relies on spurious features due to simplicity bias.

Overcoming simplicity bias on MNIST-CIFAR data. The MNIST-CIFAR task was originally used by (Shah et al., 2020) as an extreme example for demonstrating severe simplicity bias in neural networks. Each datapoint is a concatenation of one image each from the MNIST and CIFAR datasets, and labels are binary. The source dataset is completely correlated: the first class consists of (MNIST zero, CIFAR car) images, and the second class (MNIST one, CIFAR truck). The unlabeled target dataset is not correlated: we take random samples from MNIST $\in \{\text{zero}, \text{one}\}$ and CIFAR $\in \{\text{car}, \text{truck}\}$ and concatenate them. We evaluate on variants of MNIST-CIFAR with different levels of underspecification. We denote the *mix ratio* of $\mathcal{D}_S$ as $r \in [0, 1]$, where $r = 0$ indicates completely correlated data as described above, and $r = 1$ indicates the distribution of the target set. Values of r between zero and one indicate a mixture of the two distributions. Fig. 4a shows the target domain accuracy of DivDis and ERM after training with mix ratios in $\{0, 0.2, 0.4, 0.6, 0.8, 1.0\}$. For this experiment, we use active querying with $m = 1$ for disambiguation. ERM fails to do better than random guessing (50%) in the completely correlated setting represented by mix ratio zero, whereas DivDis achieves over 75% accuracy in this challenging problem. For higher ratios, the performance of the two methods converge to a similar value, because mix ratio 1 represents an i.i.d. setting where the source and target distributions are identical.

Comparison of DISAMBIGUATE strategies on MNIST-CIFAR data. Using MNIST-CIFAR data with a complete correlation, we plot the average final accuracy after the DISAMBIGUATE stage for both the active query and random

	Acc	AUC	AUC (drain)	AUC (no-drain)
ERM	0.883	0.828	0.904	0.717
Pseudo-label	0.898	0.835	0.904	0.721
DivDis	0.934	0.836	0.902	**0.737**

Table 4. Pneumothorax classification metrics on the test set of CXR-14. In addition to overall accuracy and AUC, we measure AUC separately on the two subsets of the positive class, drain and no-drain. DivDis shows higher AUC on the no-drain subset, which is more indicative of the intended population of patients not yet being treated. Performance gains on the no-drain subset also contribute positively to the overall metrics (Acc and AUC).

Table 5. CelebA dataset with complete correlation between 4 different pairs of attributes. DivDis outperforms previous methods in all but one setting.

	CelebA-CC-1		CelebA-CC-2		CelebA-CC-3		CelebA-CC-4	
	Avg (%)	Worst (%)	Avg (%)	Worst (%)	Avg (%)	Worst (%)	Avg (%)	Worst (%)
ERM	70.9 ± 2.0	57.0 ± 5.8	73.1 ± 0.9	41.1 ± 2.6	87.0 ± 0.7	71.9 ± 2.6	63.9 ± 3.5	23.0 ± 1.4
JTT	44.6 ± 1.9	26.5 ± 1.4	71.4 ± 1.9	51.2 ± 5.4	64.8 ± 4.4	34.0 ± 10.2	67.4 ± 1.4	49.3 ± 8.2
GDRO	71.6 ± 0.3	59.3 ± 2.6	71.6 ± 2.4	61.3 ± 2.3	88.2 ± 0.6	83.7 ± 0.8	65.0 ± 1.6	21.7 ± 1.5
DivDis w/o reg	91.0 ± 0.4	85.9 ± 1.0	79.7 ± 0.4	69.3 ± 1.9	79.5 ± 0.6	62.0 ± 2.6	84.7 ± 0.5	67.4 ± 1.8
DivDis	90.8 ± 0.4	85.6 ± 1.1	79.5 ± 0.2	68.5 ± 1.7	80.6 ± 0.4	67.1 ± 1.9	84.8 ± 0.4	73.5 ± 2.6

query strategies, for different number of labels used. Fig. 4b shows that active querying in particular is very efficient, and one label suffices for finding the head with highest target data accuracy. We further show the viability of disambiguation on source data: Fig. 4c shows Grad-CAM (Selvaraju et al., 2017) visualizations of two heads on a randomly sampled image from the source dataset. Even though the two heads predict the same label, they respectively focus on distinct features of the data: the MNIST region and the CIFAR region. Since we know that the true predictive feature is the CIFAR image, we can select the second head based on this single datapoint, and nothing beyond what was used during training.

CelebA hyperparameter grid. In Fig. 9, we show an additional hyperparameter grid for the CelebA dataset. This grid shows a strong corrleation between metrics with respect to hyperparameter choice, indicating that DivDis can be tuned using only labeled source data.

Additional Grad-CAM plots on MNIST-CIFAR data. In Fig. 8, we show additional Grad-CAM plots for MNIST-CIFAR data, on 6 more random datapoints from the source dataset. Compared to the example given in the main text, these examples are just as informative in terms of which head is better.

Effect of ratio. We test various values between 0 and 1 for the regularizer loss (2), on the Waterbirds benchmark. Fig. 10 shows that even using a ratio of 0.1 yields close to 80% worst-group accuracy, demonstrating that the performance of DivDis is not very sensitive to this hyperparameter.

CXR pneumothorax classification. To investigate whether DivDis can disambiguate naturally occurring spurious correlations, we consider the CXR-14 dataset (Wang et al., 2017), a large-scale dataset for pathology detection in chest radiographs. We evaluate on the binary pneumothorax classification task, which has been reported to suffer from hidden stratification: a subset of the images with the disease include a chest drain, a common treatment for the condition (Oakden-Rayner et al., 2020). We train DivDis with two heads to see whether it can disambiguate between the visual features of chest drains and lungs as a predictor for pneumothorax. In addition to ERM, we compare against the semi-supervised learning method Pseudo-label (Lee et al., 2013), to see how much of the performace gain of DivDis can be attributed to the unlabeled target set alone. In Tab. 4, we show test split accuracy and AUC, along with AUC for the subset of positive samples with and without a chest drain. Our experiments show that DivDis achieves higher AUC in the no-drain split while doing marginally worse on the drain split, indicating that the chosen head is relying more on the visual features of the lung. The overall metrics (Acc and AUC) indicate that this performance gain in the no-drain subset also leads to better performance in overall metrics.

Table 6. CXR dataset test set metrics

	Accuracy	AUC	AUC (drain)	AUC (no-drain)
ERM	0.883 ± 0.006	0.828 ± 0.001	0.904 ± 0.008	0.717 ± 0.005
Pseudolabel	0.898 ± 0.015	0.835 ± 0.004	0.904 ± 0.007	0.721 ± 0.007
DivDis	0.934 ± 0.014	0.836 ± 0.007	0.902 ± 0.006	0.737 ± 0.001

E. Finite-hypothesis Generalization Bound for Head Selection

Proposition 1. *Let the N heads have risk $l_1 \leq l_2 \ldots \leq l_N \in \mathbb{R}$ on the target dataset, and let $\Delta = l_2 - l_1$. The required number of i.i.d. labels from the target set to select the best head with probability $\geq 1 - \delta$ is $m = \frac{2(\log 2N - \log \delta)}{\Delta^2}$.*

Proof. Given m i.i.d. samples, Hoeffding's inequality gives us for all $\epsilon > 0$,

$$\mathbb{P}\left[l - \hat{l} > \epsilon\right] \leq 2\exp\left(-2m\epsilon^2\right). \tag{4}$$

The event of failing to select the best head is a superset of the following event, for which we can bound the probability as:

$$\mathbb{P}\left[\left(\left|l_1 - \widehat{l_1}\right| > \frac{\Delta}{2}\right) \vee \left(\left|l_2 - \widehat{l_2}\right| > \frac{\Delta}{2}\right) \vee \ldots \vee \left(\left|l_N - \widehat{l_N}\right| > \frac{\Delta}{2}\right)\right] \leq 2N\exp\left(-\frac{m\Delta^2}{2}\right). \tag{5}$$

Solving for $\delta = 2N\exp\left(-\frac{m\Delta^2}{2}\right)$, we get the sample size bound

$$m^* = \frac{2(\log 2N - \log \delta)}{\Delta^2}. \tag{6}$$

□

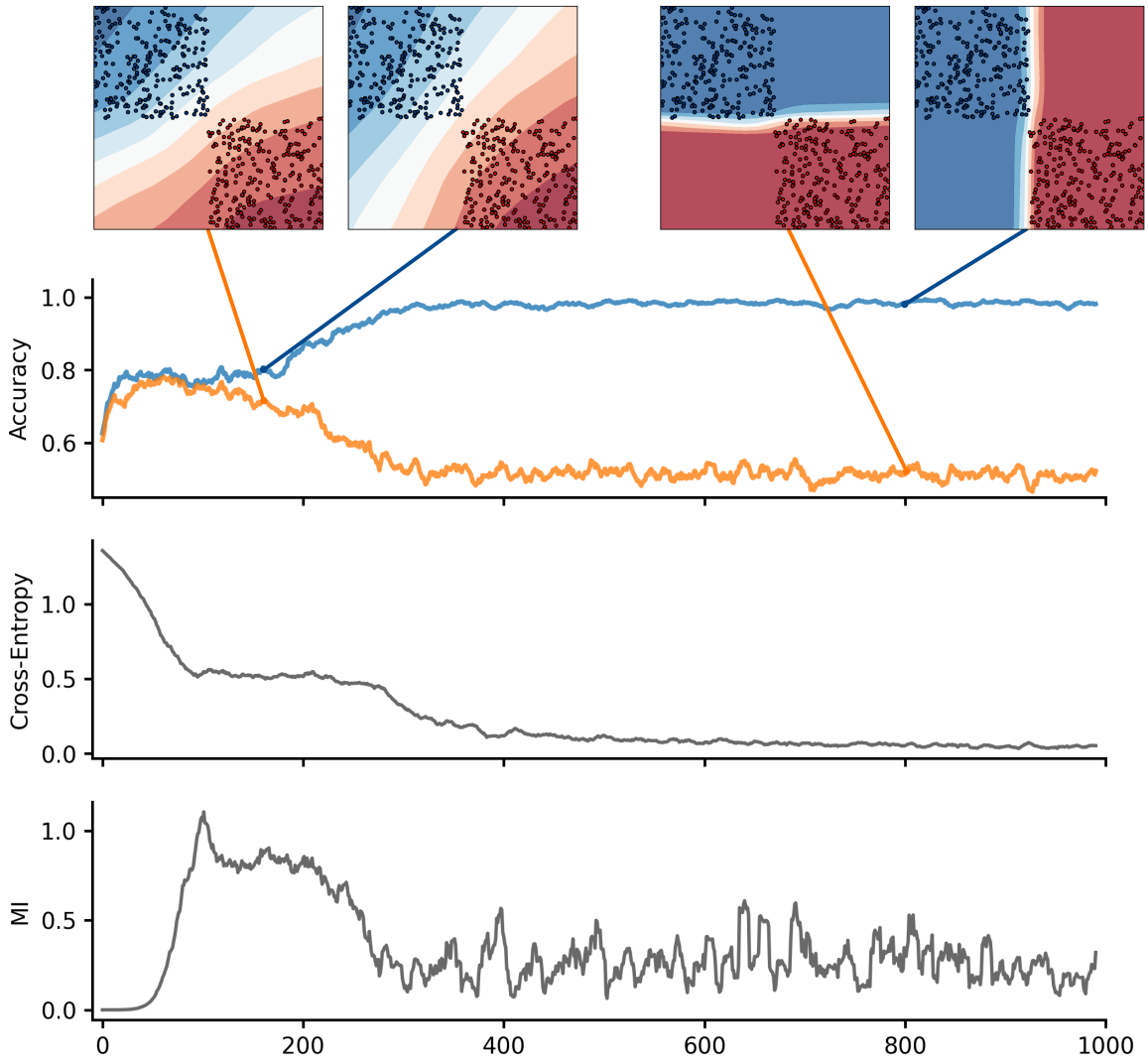


Figure 5. Extended visualization for the 2D classification task (Fig. 3), with additional curves for the cross-entropy and mutual information losses. Note that only accuracy is measured with target data, and the cross-entropy and mutual information losses are the training metrics for DIVERSIFY measured on source data. Until around iteration 100, the model initially decreases cross-entropy at the cost of increasing mutual information. The decision boundaries at this stage are similar for the two heads. Afterwards, both the mutual information and cross-entropy decrease, leading to the heads having very different decision boundaries.

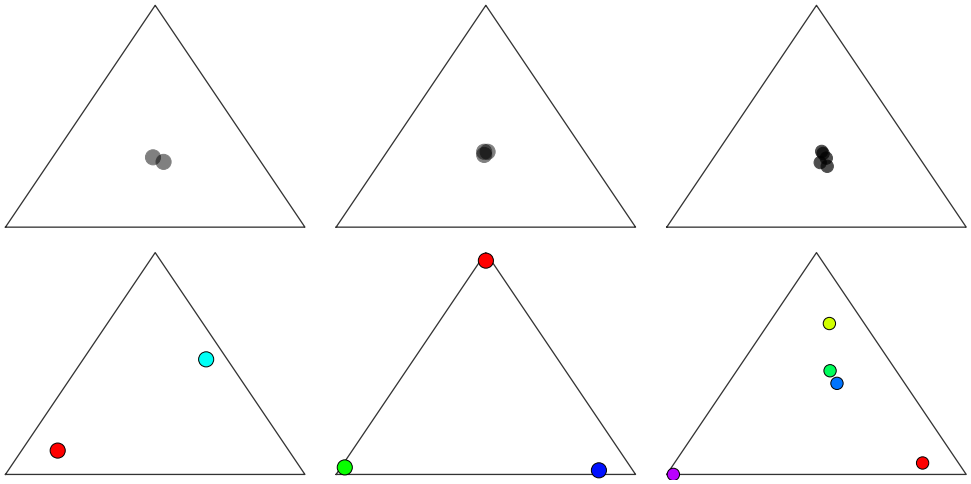


Figure 6. Visualization of $\{2, 3, 5\}$ functions trained independently (top row) and with DivDis (bottom row). Vertices of the 2-simplex represent the three dimensions of the input data. The functions learned by DivDis are much more diverse compared to independent training.

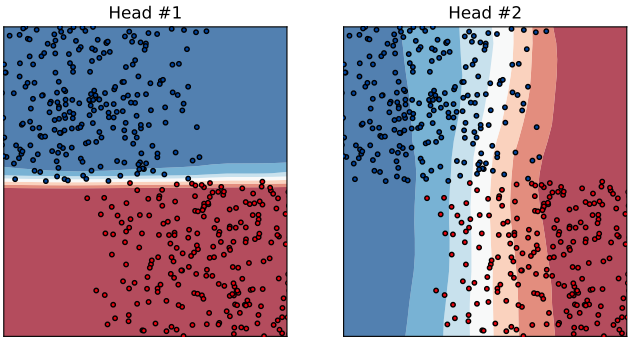


Figure 7. Functions learned by DivDis on a variant of the synthetic classification task, where the labeled source dataset has noise along the x-axis. The second head recovers the Y-axis decision boundary even though it is harder to learn due to the noise. This indicates that DivDis can successfully overcome simplicity bias and learn functions that ERM would not consider.

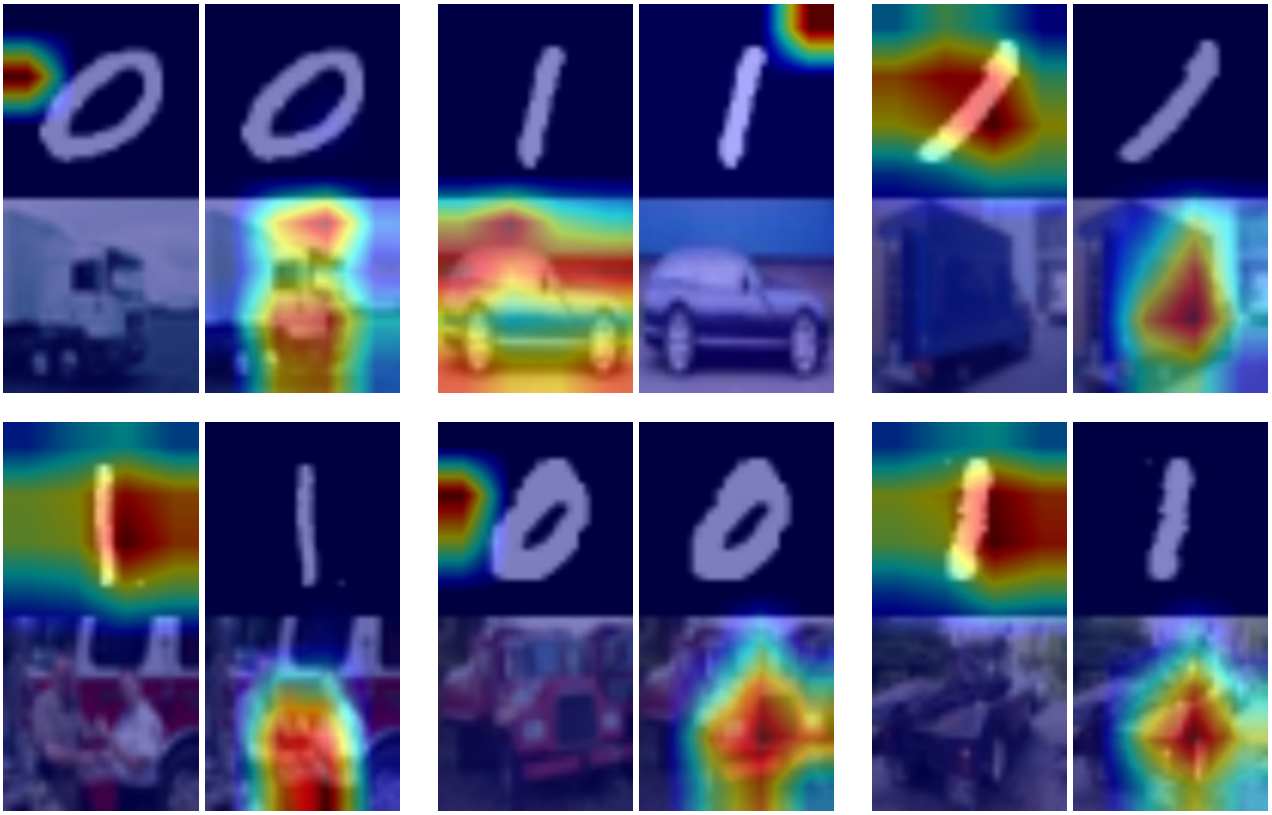


Figure 8. Additional GradCAM visualizations of two learned heads on 6 examples from the source dataset of the MNIST-CIFAR task. These examples sufficiently differentiate the best of the two heads, demonstrating the viability of the source data inspection strategy for the DISAMBIGUATE stage.

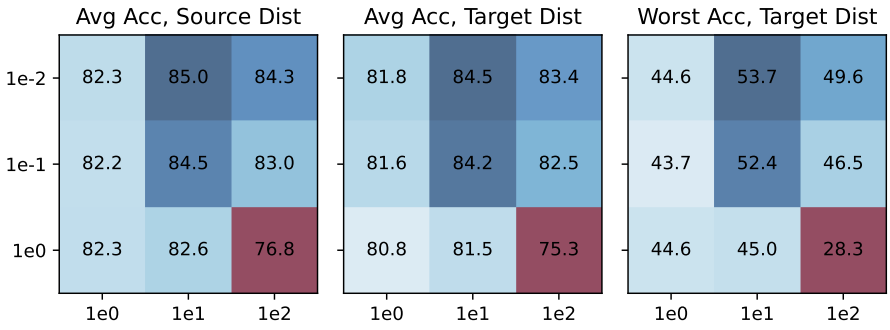


Figure 9. Grids for DivDis's two hyperparameters (λ_1, λ_2) on the CelebA dataset. Rows indicate λ_1 and columns indicate λ_2. We show three metrics measured with held-out datapoints: average accuracy on the source and target distributions and worst-group accuracy on the target distribution. We average each metric across three random seeds. The high correlation between the three metrics indicates that we can tune the hyperparameters of DivDis using only held-out labeled source data.

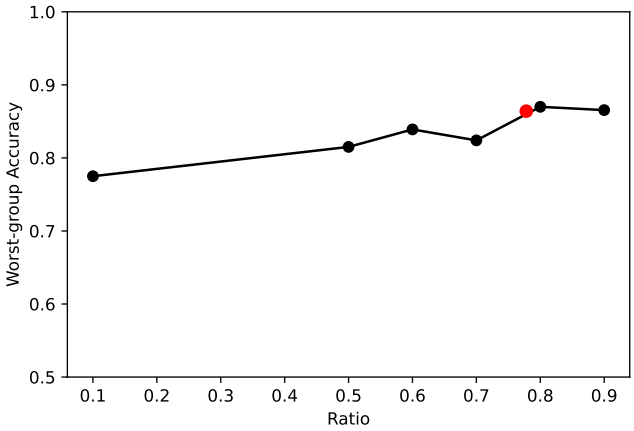


Figure 10. Worst-group accuracy on the Waterbirds benchmark when using different ratio values for $p(y)$ in the regularizer loss (2). The plot shows that the performance of DivDis is not very sensitive to this hyperparameter.