The Connection between Out-of-Distribution Generalization and Privacy of ML Models

Supplementary Material

A EVALUATION METRICS

We evaluate the privacy and generalization properties of DG training methods using four different metrics: membership inference accuracy, attribute inference accuracy, out-of-domain task accuracy, and the ability to learn stable features.

A.1 MEMBERSHIP INFERENCE (MI) ATTACKS

Several methods have been proposed in the literature to compute MI attack accuracy based on the threat-model i.e., black-box or white-box and computational power of the attacker. All these methods identify the boundary that helps to distinguish between members and non-members. As our goal is to use MI attack accuracy as a measure for privacy and perform a comparison across several method, we use the loss-based attack to measure privacy.

Loss-based attack Yeom et al. (2018) Yeom et al. (2018) proposed MI attack that relies on the loss of the target model. The attacker observes the loss values of a few samples and identifies a threshold that distinguishes members from non-members. The intuition is that training data points will have a lower loss value as compared to test data points. This attack is computationally cheap and the attacker does not need to train shadow models or an attack classifier. However, the attack assumes access to the loss values of the target model i.e., it requires white-box access to the model. Our attack accuracy provides an upper bound as we compute the threshold using a subset of the training samples and is consistent across all our experiments.

A.2 ATTRIBUTE (PROPERTY) INFERENCE (AI) ATTACKS

We use a classifier-based attribute inference attack as our second metric to evaluate the privacy of different training techniques that aim for domain generalization. The attack is similar to the MI attack except that we build a classifier to learn the distinguishing boundary. We query a subset of data points to the target model and use their prediction probability vector as input feature for the attack classifier. The ground truth is the value of the sensitive attribute for the given input. The classifier is trained to predict the attribute value for a given input feature of probability vector. This attack works in a black-box setting.

A.3 OUT-OF-DOMAIN ACCURACY

To understand the generalization ability of a training technique, we use the accuracy as a measure when computed on inputs that are generated from domains that are *not* seen during training. This is different than standard test accuracy measure where often the validation and test data have the same distribution. Since our goal is to understand the connection between domain generalization techniques and privacy, we select out-of-domain accuracy as one of our evaluation metrics.

A.4 MEASURING STABLE FEATURES USING MEAN RANK

Measuring stability of learnt features is a hard task, since the ground-truth stable features are unknown for a given classification task. For example, in a MNIST task to classify the digit corresponding to an input image, the shape of the digit can be considered as the stable (or causal) feature whereeas its color or rotation are not stable features. Even if stable features are known, in image datasets, they are typically high-level features (such as shape) that themselves need to be learnt from data. Therefore, verifying stable features directly is a non-trivial task. We describe two different ways to measure stable features — mean rank and linear-Randomized AUC for our MNIST and synthetic slab dataset results below. However, we do not have a reasonable metric to measure stable features for ChestXray dataset and leave that to future work.

Mean Rank. We use the mean rank metric proposed by Mahajan et al. (2021) for measuring stable features where the base object of an image is known. If we can select pairs of inputs with the same base object, then they should share the same causal features. Such a pair is known as a perfect match. Note that a base object refers to the same semantic input such as a person or a handwritten digit. Input images may consist of the same person in different views or the same handwritten digit in different colors or rotations, but their base object remains the same. Note that there is a many-to-one relationship between an object and its class label. Each class label consists of many objects, which in turn consist of many input images that are differentiated by certain non-stable features like view or rotation or noise.

Formally, the mean rank metrics is computed as follows. For the matches (j, k) as per the ground-truth perfect match strategy Ω, compute the mean rank for the data point j w.r.t the learnt match strategy $\Omega^{'}$ i.e. $S_{\Omega'}(j)$

$$\frac{\sum_{\Omega(j,k)=1; d\neq d'} Rank[k \in S_{\Omega'}(j)]}{\sum_{\Omega(j,k)=1; d\neq d'} 1} \tag{2}$$

Linear-Randomized AUC. Slab dataset allows to capture the effect of a single feature on the outcome prediction by constructing S-Randomized metrics (Shah et al., 2020). The idea is to replace a subset of features by drawing random samples from its marginal distribution so that we destroy any relationship between the features in the set S and the outcome y. As defined in the work (Shah et al., 2020), consider the data point $x = (x^S, x^{S^c})$, and replace the features x^S in the actual dataset $((x^S, x^{S^c}), y) \sim \mathcal{D}$ with new samples $\bar{x}^S \sim \mathcal{D}_{\mathcal{S}}$, where $\mathcal{D}_{\mathcal{S}}$ is the marginal distribution of features in the subset S. Denote this new dataset as S-Randomized and compute metrics like accuracy and AUC on it. Since the relationship between the features in set S and the outcome y has been randomized, the S-Randomized AUC/Accuracy would be close to 0.50 if the model completely relied on the features in the set S for prediction.

Hence, we take the set S to be the spurious linear feature and compute the Linear-Randomized AUC score, which captures the extent to which a model relies on the spurious feature for its prediction. Therefore, models that capture more stable features would have higher values of Linear-Randmoized AUC, as the randomization in the prediction mechanism between the linear feature and the label would affect their performance comparatively less.

B EXPERIMENTAL SETUP

B.1 OOD TRAINING METHODS AND ERM

For all methods and the respective loss equations, we use S to denote the set of source domains, N_d as the total number of samples for domain d, f as the classification model, L_d as the classification loss, and x, y to represent the data point and its corresponding true class label.

ERM-Baseline: As our baseline, we use the empirical risk minimization approach to train the model, which minimizes the empirical average of loss over training data points. $\sum_{d\sim S, i\sim N_d} L_d(f(x_i), y_i)$

It treats the data from different domains as i.i.d and simply augments them. This may lead to issues with OOD generalization Arjovsky et al. (2019); Peters et al. (2016) as we need to learn representations that are robust to the changes in the domains. Hence, a variety of approaches (described below) augment the empirical average loss with regularizers to learn domain invariant models.

Random-Match (Mahajan et al., 2021; Motiian et al., 2017; Dou et al., 2019). Random-Match matches pairs of same-class data points randomly across domains to regularize the model. The idea behind matching across domains is to learn a representations that is invariant to the changes in the source domains, which may lead to better generalization performance. The training loss objective is given by,

$$\sum_{d\sim S, i\sim N_d} L_d(h \circ \phi(x_i), y_i) + \lambda * \sum_{\Omega(j,k)=1 | j\sim N_d, k\sim N_{d'}} Dist(\phi(x_j), \phi(x_k)) \tag{3}$$

where ϕ represents some layer of the network $f = h \circ \phi$, Ω represents the match function used to randomly pair the data points across the different domains. This may not necessarily enforce learning stable features.

CSD (Piratla et al., 2020). Common-Specific Low-Rank Decomposition (CSD) leads to effective OOD generalization by separating the domain specific and domain invariant parameters, and utilizes the domain invariant parameters for reliable prediction on OOD data. It decomposes the model's final classification layer parameters w as $w = w_s + W * \gamma$, where W represents the k-rank decomposition matrix, w_s represents the domain invariant parameters and γ represent the k domain specific parameters. It optimizes empirical average loss with both the domain invariant and domain specific parameters, along with an orthonormality regularizer that aims to make w_s orthogonal to the decomposition matrix W. Please refer to Algorithm 1 in their paper Piratla et al. (2020) for more details.

IRM (Arjovsky et al., 2019). Invariant Risk Minimization (IRM) aims to learn invariant predictors that simultaneously achieve optimal empirical risk on all the data domains. It minimizes the empirical average loss, and regularizes the model by the norm of gradient of the loss at each source domain as follows:

$$\sum_{d \sim S, i \sim N_d} L_d(w \circ \phi(x_i), y_i) + \lambda * \sum_{d \sim S} ||\nabla_{w|w=1.0} \sum_{i \sim N_d} L_d(w \circ \phi(x_i), y_i)||^2 \tag{4}$$

where $f = w \circ \phi$ and λ is a hyper parameter. In practice, ϕ is taken to be the final layer of the model f, (which makes ϕ and f to be the same). Hence, minimizing the above loss would lead to low norm of the domain specific loss function's gradient and guide the model towards learning an invariant classifier, which is optimal for all the source domains.

MatchDG (Mahajan et al., 2021). The algorithm enforces the same representation for pairs of data points from different domains that share the same causal features. It uses contrastive learning to learn a *matching* function to obtain pairs that share stable causal features between them. The loss function for the method is similar to that of Random-Match (Eq 3), with Ω representing the match function learnt by contrastive loss minimization. Hence, the algorithm consists of two phases; where it learns the matching function Ω in the first phase, and then minimizes the loss function in Eq 3 during the second phase to learn the final model. Please refer to the Algorithm 1 in Mahajan et al. Mahajan et al. (2021) for more details.

Perfect-Match (Mahajan et al., 2021; Hendrycks et al., 2019). Finally, we use an algorithm that can be considered to learn *true stable* features for given data, since it relies on knowledge of true base object for a subset of images (and thus guaranteed shared causal features between them). This approach again has a similar formulation to Eq 3, where the match function Ω is satisfied for data points from different domains that share the same base causal object. Hence, it aims to learn similar representations for two data points that only differ in terms of the domain specific attributes.

Hybrid (Mahajan et al., 2021). Perfect matches as explained above are often unobserved but given through Oracle access. However, in real datasets, augmentations can also provide perfect matches, leading to the *Hybrid* approach using both the MatchDG and augmented Perfect-Match. It learns two match functions, one on the different source domains as per MatchDG, and the other on the augmented domains using Perfect-Match.

Note that Perfect-Match is an ideal training algorithm that assumes knowledge of ground-truth matches across domains, and therefore cannot be applied in real-world settings. In contrast, the Hybrid algorithm depends on creating matches of the same base object using self-augmentations and can be used practically whenever augmentations are easy to create (such as in image datasets). In the experiments that follow, we use the PerfectMatch algorithm for the simulated Rotated-MNIST and Fashion-MNIST datasets, where it should be considered as an ideal method. For the real-world Chest X-rays dataset, we use the practical Hybrid algorithm since we have no knowledge about the true perfect matches.

Dataset	#Classes	#Domains	Source Domains	Target Domains	Samples/ Domain
Rotated-MNIST	10	7	$15^\circ, 30^\circ, 45^\circ, 60^\circ, 75^\circ$	$0^\circ, 90^\circ$	2000
Fashion-MNIST	10	7	$15^\circ, 30^\circ, 45^\circ, 60^\circ, 75^\circ$	$0^\circ, 90^\circ$	2000
ChestXray	7	3	NIH, ChexPert	RSNA	800

Table 1: Dataset details

Table 2: Hyperparamter details for all the datasets. We took the optimal hyperparameter for each method following Mahajan et al. (2021). Complete details regarding the grid range used for hyperparameter tuning can be found in Table 8 in their paper.

Dataset Range	Hyper Parameter	Optimal Value
Rotated & Fashion MNIST	Total Epochs	25
	Learning Rate	0.01
	Batch Size	16
	Weight Decay	0.0005
	Match Penalty	0.1
	IRM Penalty	1.0 (RotMNIST); 0.05 (FashionMNIST)
	IRM Threshold	5 (RotMNIST), 0 (FashionMNIST)
Chest X-ray	Total Epochs	40
	Learning Rate	0.001
	Batch Size	16
	Weight Decay	0.0005
	Match Penalty	10.0 (`RandMatch`), 50.0 (`MatchDG`, `MDGHybrid`)
	IRM Penalty	10.0
	IRM Threshold	5
Slab Dataset	Total Epochs	100
	Learning Rate	0.1
	Batch Size	128
	Weight Decay	0.0005
	Match Penalty	1.0
	IRM Penalty	10.0
	IRM Threshold	2

B.2 IMPLEMENTATION DETAILS

Model Training To summarize, we use the model ResNet-18 (no pre-training) for the Rotated-MNIST and Fashion-MNIST dataset, and we use pre -trained DenseNet-121 for the ChestXRay dataset. For the matching based methods (Random-Match, MatchDG, Perfect-Match), we use the final classification layer of the network as ϕ and for the matching loss regularizer (Eq 3). For all the methods across datasets, we use Cross Entropy for the classification loss (L_d), and use SGD to optimize the loss. Also, we use the data from the source domains for validation and never expose the model to any data from the target domains while training. The details regarding the domain and dataset sizes are described in the Table 1.

For the slab dataset, we sample 1k data points per domain and additional 250 data points per source domain for validation. We construct two source domains, with noise probabilities between the linear feature and the label as p=0.0 and p=0.1, while there is constant domain-independent noise (p=0.1) in prediction mechanism between the slab feature and the label. The model architecture consists of a representation layer (FC: input-dimension * 100; ReLU activation), followed by a classification layer (FC: 100*100; FC: 100*total-classes), where FC: a*b denotes a full connected layer with input and output dimension as a, b respectively. For the matching based methods (Random-Match, MatchDG, Perfect-Match), the representation layer is taken as ϕ.

SGD is used to optimize the training loss across all the datasets and methods, with the details regarding other hyperparameters given in the table 2.

Membership Inference (MI) Attacks. For implementing MI attacks, we first create the attack-train and attack-test dataset from the original train and test dataset for the ML model. We first sample N number of data points (N: $2,00$ for Rotated-MNIST & Fashion-MNIST, N: $1,000$ for ChestXRay, N :400 for Slab dataset) from both the original train and test dataset to create the attack-train dataset. Similarly, we sample an additional set of N data points from original train, test dataset to create the attack-test dataset. For the selection of the threshold τ required for the Loss-based attack, we compute the mean loss among all the members in the attack-train dataset. We use this threshold to evaluate performance on the attack-test dataset.

Attribute Inference (AI) Attacks. We use all the source and target domains, and sample data points from their training/test set to create the attack-train/attack-test dataset respectively. We then label data points in the attack-train and attack-test dataset based on the attribute, and then train a classifier to discriminate among different attributes. We train a 2-layer fully-connected network (with hidden dimensions 8, 4 respectively) to distinguish between different attributes for all the models. We use Adam Optimizer with learning rate 0.001, batch size 64 and 5000 steps / 80 epochs.

DP Training. We train the differentially-private models using Pytorch Opacus library support for DP-SGD training. We train models with ϵ ranging from 1 to 10. We use max clipping norm of 5.0 and the noise multiplier is inferred from the ϵ value. The other hyperparamters like total number of epochs, learning rate, etc. are same for the non DP case in Rotated-MNIST & Fashion-MNIST, with the exception of a higher batch size (10 × the usual batch size) as it typically helps with dp training.

C ADDITIONAL EXPERIMENTAL RESULTS

C.1 SLAB DATASET

Figure 7 shows the result for our synthetic slab dataset with an additional metric of train-test generalization gap. We find that for this dataset, the generalization gap metric can be used to measure the amount of stable features learnt which in turn correlates with MI attack accuracy for different training algorithms.

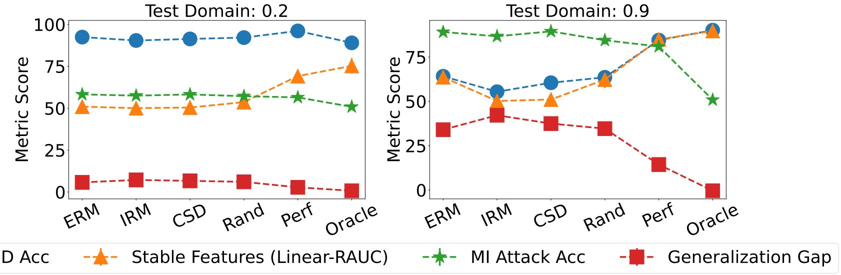

Figure 7: Results for synthetic slab dataset with generalization gap included.

Table 3: Attribute Attack Accuracy for the dataset Rotated-MNIST with color as an attribute. We consider the attribute as binary (0: no color; 1: color), thus the best case AI Attack accuracy would be 50% via random guess.

Metrics	ERM	Random-Match	IRM	CSD	MatchDG	Perfect-Match/ Hybrid
Train Accuracy	99.7 (0.02)	99.8 (0.05)	99.7 (0.02)	99.9 (0.01)	99.4 (0.10)	97.8 (0.31)
OOD Accuracy	95.2 (0.17)	**96.8 (0.06)**	95.4 (0.21)	96.6 (0.48)	95.5 (0.28)	95.7 (0.19)
AI Attack Accuracy	97.7 (0.41)	83.0 (3.09)	97.9 (0.30)	92.4 (1.30)	95.3 (3.80)	**69.9 (0.03)**
OOD (Permute) Accuracy	25.3 (0.17)	28.9 (1.81)	25.4 (0.21)	42.9 (2.30)	25.5 (0.28)	**93.4 (0.35)**

Table 4: Attribute Attack Accuracy for different datasets with domains as an attribute.

Dataset	Random Guess	ERM	Random-Match	IRM	CSD	MatchDG	Perfect-Match/ Hybrid
Rotated-MNIST	14.3%	27.6 (0.84)	20.4 (0.46)	26.6 (0.71)	25.0 (0.40)	19.3 (0.62)	**16.6 (0.55)**
Fashion-MNIST	14.3%	26.6 (0.59)	21.8 (0.77)	23.8 (0.86)	26.9 (0.93)	21.9 (0.29)	**21.5 (0.97)**
ChestXray	33.3%	61.8 (1.02)	58.4 (0.58)	63.4 (2.28)	67.8 (3.05)	57.9 (0.58)	**57.1 (0.21)**

C.2 ATTRIBUTE ATTACKS

Domain-Agnostic Attribute: We consider a domain-agnostic attribute — color in the rotated digit dataset which is not related to the prediction task and whose distribution does not depend on the domain. We randomly introduce color as an attribute with 70% probability to Rotated-MNIST dataset. The attacker's goal is to infer whether a given input is colored or not (binary task) with only access to the output prediction. This setting can be observed for example, where the user sends embedding of their input to the server instead of the original input (Song & Shmatikov, 2020).

Figure 6 shows our main results with additional results in Table 3. We observe that Perfect-Match has the lowest attack accuracy as compared to all other training algorithms, demonstrating that the ability to learn stable features provides inherent ability to defend against a harder task of attribute inference when the attribute itself is domain-agnostic. Note that the OOD accuracy does not convey much information about the privacy risk through an AI attack: all methods obtain OOD accuracy within a narrow range of 95.9%-97.5%, but the attack accuracy range from 70.2% for Perfect-Match to 99.3% for IRM.

We hypothesize that the difference in attack accuracy is because of the differing extents to which the ML models utilize color as a feature. To verify the hypothesis, we construct a new, *permuted* test domain where the color of each colored image is permuted randomly to a different value. Thus, the correlation in the training data between color and the class label is no longer present in the test domain. Under such a test domain, we observe larger differences in OOD accuracy that correspond to the AI attack accuracy reported above: Perfect-Match obtains the highest OOD accuracy (94.9%) as compared to other methods that range between 28% to 42%. The low accuracy of other DG methods motivates further research to introduce constraints for learning stable features.

Domain as a sensitive attribute: In this attack, we consider domain of the data from which it is generated as a sensitive attribute that the attacker is trying to learn. Table 4 shows the attack results across different training methods on all the three datasets. For Rotated-MNIST and Fashion-MNIST, we train the attack classifier to distinguish among all the 7 angles of rotation and hence the baseline accuracy with random guess 14.33% while for ChestXray, we aim to distinguish among 3 domains resulting in a baseline of 33.33%.

Similar to MI attacks, we observe that PerfectMatch/Hybrid approach has the lowest attack accuracy followed by MatchDG. PerfectMatch/Hybrid, having access to perfect matches based on self-augmentations, has access to the pairs of inputs that share the same causal features, and hence obtains both highest OOD accuracy (as we saw earlier) and the lowest AI attack accuracy. However, there is not a one-to-one correspondence between OOD accuracy and AI attack accuracy. CSD obtains one of the highest OOD accuracies on all datasets but its attack accuracy is one of the highest. We suspect that this discrepancy is due to the different objective of CSD algorithm compared to the matching algorithms of PerfectMatch/Hybrid and MatchDG: CSD does not aim to obtain the same representations across domains. Thus, when domains convey sensitive information, it is preferable to employ matching-based methods that can provide both high OOD accuracy and low attribute inference attack accuracy. Unlike in membership inference attacks, the standard ERM training algorithm does not perform well on attribute inference, obtaining one of the highest attack accuracies.

That said, it is somewhat intuitive that DG training approaches that are designed to be domain invariant are able to defend against attribute inference attacks as compared to ERM. Therefore, we present another attack with a domain-agnostic attribute to understand whether DG methods can mitigate such attacks well.

REFERENCES

Kaggle: Rsna pneumonia detection challenge, 2018. URL `https://www.kaggle.com/c/rsna-pneumonia-detection-challenge`.

Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pp. 308–318, 2016.

Kartik Ahuja, Karthikeyan Shanmugam, Kush Varshney, and Amit Dhurandhar. Invariant risk minimization games. *arXiv preprint arXiv:2002.04692*, 2020.

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

Nader Asadi, Amir M Sarfi, Mehrdad Hosseinzadeh, Zahra Karimpour, and Mahdi Eftekhari. Towards shape biased unsupervised representation learning for domain generalization. *arXiv preprint arXiv:1909.08245*, 2019.

Giuseppe Ateniese, Luigi V. Mancini, Angelo Spognardi, Antonio Villani, Domenico Vitali, and Giovanni Felici. Hacking smart machines with smarter ones: How to extract meaningful data from machine learning classifiers. 2015.

Nicholas Carlini, Chang Liu, Úlfar Erlingsson, Jernej Kos, and Dawn Song. The secret sharer: Evaluating and testing unintended memorization in neural networks. In *28th {USENIX} Security Symposium ({USENIX} Security 19)*, pp. 267–284, 2019.

Qi Dou, Daniel Coelho de Castro, Konstantinos Kamnitsas, and Ben Glocker. Domain generalization via model-agnostic learning of semantic features. In *Advances in Neural Information Processing Systems*, pp. 6447–6458, 2019.

Cynthia Dwork, Aaron Roth, et al. The algorithmic foundations of differential privacy. *Foundations and Trends® in Theoretical Computer Science*, 9(3–4):211–407, 2014.

Sreya Francis, Irene Tenison, and Irina Rish. Towards causal federated learning for enhanced robustness and privacy. *arXiv preprint arXiv:2104.06557*, 2021.

Karan Ganju, Qi Wang, Wei Yang, Carl A. Gunter, and Nikita Borisov. Property inference attacks on fully connected neural networks using permutation invariant representations. In *ACM Conference on Computer and Communications Security (CCS)*, 2018.

Muhammad Ghifary, W Bastiaan Kleijn, Mengjie Zhang, and David Balduzzi. Domain generalization for object recognition with multi-task autoencoders. In *Proceedings of the IEEE international conference on computer vision*, pp. 2551–2559, 2015.

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, 2016.

Ishaan Gulrajani and David Lopez-Paz. In search of lost domain generalization. *arXiv preprint arXiv:2007.01434*, 2020.

Jihun Hamm, Yingjun Cao, and Mikhail Belkin. Learning privately from multiparty data. In *International Conference on Machine Learning*, pp. 555–563, 2016.

Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. *arXiv preprint arXiv:1903.12261*, 2019.

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

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.

Maximilian Ilse, Jakub M Tomczak, Christos Louizos, and Max Welling. Diva: Domain invariant variational autoencoders. In *Medical Imaging with Deep Learning*, pp. 322–348. PMLR, 2020.

Jeremy Irvin, Pranav Rajpurkar, Michael Ko, Yifan Yu, Silviana Ciurea-Ilcus, Chris Chute, Henrik Marklund, Behzad Haghgoo, Robyn Ball, Katie Shpanskaya, et al. Chexpert: A large chest radiograph dataset with uncertainty labels and expert comparison. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pp. 590–597, 2019.

Bargav Jayaraman and David Evans. Evaluating differentially private machine learning in practice. In *28th {USENIX} Security Symposium ({USENIX} Security 19)*, pp. 1895–1912, 2019.

Jinyuan Jia, Ahmed Salem, Michael Backes, Yang Zhang, and Neil Zhenqiang Gong. Memguard: Defending against black-box membership inference attacks via adversarial examples. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, pp. 259–274, 2019.

Ya Li, Mingming Gong, Xinmei Tian, Tongliang Liu, and Dacheng Tao. Domain generalization via conditional invariant representations. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

Sara Magliacane, Thijs van Ommen, Tom Claassen, Stephan Bongers, Philip Versteeg, and Joris M Mooij. Domain adaptation by using causal inference to predict invariant conditional distributions. In *Advances in Neural Information Processing Systems*, pp. 10846–10856, 2018.

Divyat Mahajan, Shruti Tople, and Amit Sharma. Domain generalization using causal matching. In *International Conference on Machine Learning*, pp. 7313–7324. PMLR, 2021.

Luca Melis, Congzheng Song, Emiliano De Cristofaro, and Vitaly Shmatikov. Exploiting unintended feature leakage in collaborative learning. In *IEEE Symposium on Security and Privacy*, 2019.

Saeid Motiian, Marco Piccirilli, Donald A Adjeroh, and Gianfranco Doretto. Unified deep supervised domain adaptation and generalization. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 5715–5725, 2017.

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

Milad Nasr, Reza Shokri, and Amir Houmansadr. Machine learning with membership privacy using adversarial regularization. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, pp. 634–646. ACM, 2018.

Milad Nasr, Reza Shokri, and Amir Houmansadr. Comprehensive privacy analysis of deep learning: Stand-alone and federated learning under passive and active white-box inference attacks. In *IEEE Symposium on Security and Privacy*, 2019.

Nicolas Papernot, Martín Abadi, Ulfar Erlingsson, Ian Goodfellow, and Kunal Talwar. Semi-supervised knowledge transfer for deep learning from private training data. In *ICLR*, 2017.

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

Vihari Piratla, Praneeth Netrapalli, and Sunita Sarawagi. Efficient domain generalization via common-specific low-rank decomposition. *Proceedings of the International Conference of Machine Learning (ICML) 2020*, 2020.

Md Atiqur Rahman, Tanzila Rahman, Robert Laganière, Noman Mohammed, and Yang Wang. Membership inference attack against differentially private deep learning model. *Trans. Data Priv.*, 11(1):61–79, 2018.

Ahmed Salem, Yang Zhang, Mathias Humbert, Mario Fritz, and Michael Backes. Ml-leaks: Model and data independent membership inference attacks and defenses on machine learning models. In *NDSS*, 2019.

Harshay Shah, Kaustav Tamuly, Aditi Raghunathan, Prateek Jain, and Praneeth Netrapalli. The pitfalls of simplicity bias in neural networks. *arXiv preprint arXiv:2006.07710*, 2020.

Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership inference attacks against machine learning models. In *Security and Privacy (SP), 2017 IEEE Symposium on*, pp. 3–18. IEEE, 2017.

Congzheng Song and Vitaly Shmatikov. Overlearning reveals sensitive attributes. In *ICLR*, 2020.

Liwei Song and Prateek Mittal. Systematic evaluation of privacy risks of machine learning models. *arXiv preprint arXiv:2003.10595*, 2020.

Shruti Tople, Amit Sharma, and Aditya V. Nori. Alleviating privacy attacks via causal learning. In *ICML*, 2020.

Stacey Truex, Ling Liu, Mehmet Emre Gursoy, Lei Yu, and Wenqi Wei. Towards demystifying membership inference attacks. *arXiv preprint arXiv:1807.09173*, 2018.

Xiaosong Wang, Yifan Peng, Le Lu, Zhiyong Lu, Mohammadhadi Bagheri, and Ronald M Summers. 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*, pp. 2097–2106, 2017.

Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

Samuel Yeom, Irene Giacomelli, Matt Fredrikson, and Somesh Jha. Privacy risk in machine learning: Analyzing the connection to overfitting. In *2018 IEEE 31st Computer Security Foundations Symposium (CSF)*, pp. 268–282. IEEE, 2018.

Santiago Zanella-Béguelin, Lukas Wutschitz, Shruti Tople, Victor Rühle, Andrew Paverd, Olga Ohrimenko, Boris Köpf, and Marc Brockschmidt. Analyzing information leakage of updates to natural language models. In *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security*, 2020.

Wanrong Zhang, Shruti Tople, and Olga Ohrimenko. Dataset-level attribute leakage in collaborative learning. *arXiv preprint arXiv:2006.07267*, 2020.