FISHR: INVARIANT GRADIENT VARIANCES FOR OUT-OF-DISTRIBUTION GENERALIZATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Learning robust models that generalize well under changes in the data distribution is critical for real-world applications. To this end, there has been a growing surge of interest to learn simultaneously from multiple training domains — while enforcing different types of invariance across those domains. Yet, all existing approaches fail to show systematic benefits under controlled evaluation protocols. In this paper, we introduce a new regularization — named Fishr — that enforces domain invariance in the space of the gradients of the loss: specifically, the domain-level variances of gradients are matched across training domains. Our approach is based on the close relations between the gradient covariance, the Fisher Information and the Hessian of the loss: in particular, we show that Fishr eventually aligns the domain-level loss landscapes locally around the final weights. Extensive experiments demonstrate the effectiveness of Fishr for out-of-distribution generalization. Notably, Fishr improves the state of the art on the DomainBed benchmark and performs consistently better than Empirical Risk Minimization. Our code is available anonymously at https://anonymous.4open.science/r/fishr-anonymous-EBB6/.

1 INTRODUCTION

The success of deep neural networks in supervised learning (Krizhevsky et al., 2012) relies on the crucial assumption that the train and test data distributions are identical. In particular, the tendency of networks to rely on simple features (Kalimeris et al., 2019; Valle-Perez et al., 2019; Geirhos et al., 2020) is generally a desirable behavior reflecting Occam's razor. However, in case of distribution shift, this simplicity bias deteriorates performance since more complex features are needed (Tenenbaum, 2018; Shah et al., 2020). For example, in the fight against Covid-19, most of the deep learning methods developed to detect coronavirus from chest scans were shown useless for clinical use (DeGrave et al., 2021; Roberts et al., 2021): indeed, networks exploited simple bias in the training datasets such as patients' age or body position rather than 'truly' analyzing medical pathologies.

To better generalize under distribution shifts, most works such as Blanchard et al. (2011) or Muandet et al. (2013) assume that the training data is divided into different training domains in which there is a constant underlying causal mechanism (Peters et al., 2016). To remove the domain-dependent explanations, different **invariance criteria across those training domains** have been proposed. (Ganin et al., 2016; Sun et al., 2016; Sun & Saenko, 2016) enforce similar feature distributions, others (Arjovsky et al., 2019; Krueger et al., 2021) force the classifier to be simultaneously optimal across all domains. Yet, despite the popularity of this research topic, none of these methods perform significantly better than the classical Empirical Risk Minimization (ERM) when applied with controlled model selection and restricted hyperparameter search (Gulrajani & Lopez-Paz, 2021; Ye et al., 2021). These failures motivate the need for new ideas.

To foster the emergence of a shared mechanism with consistent generalization properties, our intuition is that learning should progress consistently and similarly across domains. Besides, the learning procedure of deep neural networks is dictated by the distribution of the gradients with respect to the network weights (Yin et al., 2018; Sankararaman et al., 2020) — usually backpropagated in the network during gradient descent. Thus, we seek distributional invariance across domains in the gradient space: **domain-level gradients should be similar, not only in average direction, but most importantly in statistics such as covariance and dispersion**.

In this paper, we propose the Fishr regularization for out-of-distribution generalization in classification — summarized in Fig. 1. We **match the domain-level gradient variances**, *i.e.*, the second moment of the gradient distributions. In contrast, previous gradient-based works such as Fish (Shi et al., 2021) only match the domain-level gradients means, *i.e.*, the first moment.

Moreover, our strategy is also motivated by the close relations between the gradient variance, the Fisher Information (Fisher, 1922) and the Hessian. This explains the name of our work, Fishr, using gradients as in Fish and related to the Fisher Information Matrix. Notably, we will study how **Fishr forces the model to have similar domain-level Hessians** and promotes consistent explanations — by generalizing the inconsistency formalism introduced in Parascandolo et al. (2021).

To reduce the computational cost, we justify an approximation that only considers the gradients in the classifier. This is simple to implement with the BackPACK (Dangel et al., 2020) package.

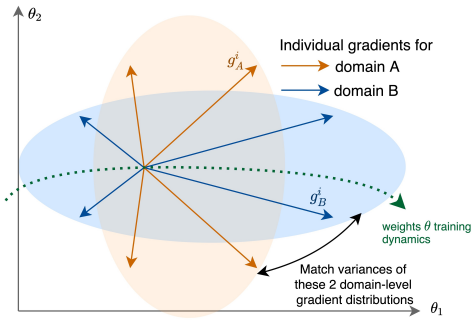

Figure 1: **Fishr principle**. Fishr tackles the individual (per-sample) gradients of the loss in the network weights θ. Indeed, Fishr matches the domain-level gradient variances of the distributions across the two training domains: A ($\{\boldsymbol{g}_A^i\}_{i=1}^{n_A}$ in orange) and B ($\{\boldsymbol{g}_B^i\}_{i=1}^{n_B}$ in blue). We will show how this regularization during the learning of θ improves the out-of-distribution generalization properties by aligning the domain-level loss landscapes at convergence.

We summarize our contributions as follows:

- We introduce the Fishr regularization that brings closer the domain-level gradient variances.
- Based on the relation between the gradient covariance, the Fisher Information and the Hessian, we show that Fishr matches domain-level Hessians and improves generalization by reducing inconsistencies across domains.
- We justify a simple and scalable implementation.

Empirically, we first validate that Fishr tackles distribution shifts on the synthetic Colored MNIST (Arjovsky et al., 2019). Then, we show that Fishr performs best on the DomainBed benchmark (Gulrajani & Lopez-Paz, 2021) with the 'Oracle' model selection method and third with the 'Training' model selection when compared with state-of-the-art counterparts. Critically, Fishr is the only method to perform better (on VLCS, OfficeHome, TerraIncognita and DomainNet) or similarly (on PACS) than ERM with both selection methods on all 'real' datasets.

2 Context and Related Work

We first describe our task and provide the notations used along our paper. Then we remind some important related works to understand how our Fishr stands in a rich literature.

Problem definition and notations We study out-of-distribution (OOD) generalization for classification. Our model is a deep neural network (DNN) f_θ (parametrized by θ) made of a deep features extractor Φ_ϕ on which we plug a dense linear classifier w_ω : $f_\theta = w_\omega \circ \Phi_\phi$ and $\theta = (\phi, \omega)$. In training, we have access to different domains $\mathcal{E}$: for each domain $e \in \mathcal{E}$, the dataset $\mathcal{D}_e = \left\{\left(\boldsymbol{x}_e^i, \boldsymbol{y}_e^i\right)\right\}_{i=1}^{n_e}$ contains n_e i.i.d. (input, labels) samples drawn from a domain-dependent probability distribution. Combined together, the datasets $\{\mathcal{D}_e\}_{e\in\mathcal{E}}$ are of size $n = \sum_{e\in\mathcal{E}} n_e$. Our goal is to learn weights θ so that f_θ predicts well on a new test domain, unseen in training. As described in Koh et al. (2020) and Ye et al. (2021), most common distribution shifts are diversity shifts — where the training and test distributions comprise data from related but distinct domains — or correlation shifts — where the distribution of the covariates at test time differs from the one during training. To generalize well despite these distribution shifts, f_θ should ideally capture an invariant mechanism across training domains. Following standard notations, $\|\boldsymbol{M}\|_F^2$ denotes the Frobenius norm of matrix $\boldsymbol{M}$; $\|\boldsymbol{v}\|_2^2$ denotes the euclidean norm of vector $\boldsymbol{v}$; $\mathbf{1}$ is a column vector with all elements equal to 1.

The standard **Expected Risk Minimization** (ERM) (Vapnik, 1999) framework simply minimizes the average empirical risk over all training domains, *i.e.*, $\frac{1}{|\mathcal{E}|}\sum_{e\in\mathcal{E}}\mathcal{R}_e(\theta)$ where $\mathcal{R}_e(\theta) = \frac{1}{n_e}\sum_{i=1}^{n_e}\ell\left(f_\theta\left(\boldsymbol{x}_e^i\right), \boldsymbol{y}_e^i\right)$ and ℓ is the loss, usually the negative log-likelihood. Many approaches try to exploit some external source of knowledge (Xie et al., 2021), in particular the domain information. As a side note, these partitions may be inferred if not provided (Creager et al., 2021). Some works explore data augmentations to mix samples from different domains (Wang et al., 2020; Wu et al., 2020), some re-weight the training samples to favor underrepresented groups (Sagawa et al., 2020a;b; Zhang et al., 2021) and others include domain-dependent weights (Ding & Fu, 2017; Mancini et al., 2018). Yet, most recent works promote invariance via a regularization criterion and only differ by the choice of the statistics to be matched across training domains. They can be categorized into three groups: these methods enforce agreement either (1) in features (2) in predictors or (3) in gradients.

First, some approaches aim at extracting **domain-invariant features** and were extensively studied for unsupervised domain adaptation. The features are usually aligned with adversarial methods (Ganin et al., 2016; Gong et al., 2016; Li et al., 2018b;c) or with kernel methods (Muandet et al., 2013; Long et al., 2014). Yet, the simple covariance matching in CORAL (Sun et al., 2016; Sun & Saenko, 2016) performs best on various tasks for OOD generalization (Gulrajani & Lopez-Paz, 2021). With $\boldsymbol{Z}_e^{ij}$ the j-th dimension of the features extracted by Φ_ϕ for the i-th example $\boldsymbol{x}_e^i$ of domain $e \in \mathcal{E} = \{A, B\}$, CORAL minimizes $\|\mathrm{Cov}(\boldsymbol{Z}_A) - \mathrm{Cov}(\boldsymbol{Z}_B)\|_F^2$ where $\mathrm{Cov}(\boldsymbol{Z}_e) = \frac{1}{n_e-1}(\boldsymbol{Z}_e^\top\boldsymbol{Z}_e - \frac{1}{n_e}\left(\boldsymbol{1}^\top\boldsymbol{Z}_e\right)^\top\left(\boldsymbol{1}^\top\boldsymbol{Z}_e\right))$ is the feature covariance matrix. CORAL is more powerful than mere feature matching $\left\|\frac{1}{n_A}\boldsymbol{1}^\top\boldsymbol{Z}_A - \frac{1}{n_B}\boldsymbol{1}^\top\boldsymbol{Z}_B\right\|_2^2$ as in Deep Domain Confusion (DDC) (Tzeng et al., 2014). Yet, Johansson et al. (2019) and Zhao et al. (2019) show that these approaches are insufficient to guarantee good generalization.

Motivated by arguments from causality (Pearl, 2009) and the idea that statistical dependencies are epiphenomena of an underlying causal structure, Invariant Risk Minimization (IRM) (Arjovsky et al., 2019) explains that the **predictor should be invariant** (Peters et al., 2016; Rojas-Carulla et al., 2018), *i.e.*, simultaneously optimal across all domains. Among many suggested improvements (Chang et al., 2020; Idnani & Kao, 2020; Teney et al., 2020; Ahmed et al., 2021), Risk Extrapolation (V-REx) (Krueger et al., 2021) argues that training risks from different domains should be similar and thus penalizes $|\mathcal{R}_A - \mathcal{R}_B|^2$ when $\mathcal{E} = \{A, B\}$. These ideas have been applied in semi-supervised learning (Li et al., 2021). Yet, recent works point out pitfalls of IRM (Javed et al., 2020; Guo et al., 2021; Kamath et al., 2021), that does not provably work with non-linear data (Rosenfeld et al., 2021) and could not improve over ERM when hyperparameter selection is restricted (Koh et al., 2020; Gulrajani & Lopez-Paz, 2021; Ye et al., 2021).

A third and most recent line of work promotes **agreements between gradients** with respect to θ. Gradient agreements help batches from different tasks to cooperate, and have been previously employed for multitasks (Du et al., 2018; Yu et al., 2020), continual (Lopez-Paz & Ranzato, 2017), meta (Finn et al., 2017; Zhang et al., 2020) and reinforcement (Zhang et al., 2019) learning. In OOD generalization, Koyama & Yamaguchi (2020); Parascandolo et al. (2021); Shi et al. (2021) try to find minimas in the loss landscape that are shared across domains. Specifically, these works tackle the domain-level expected gradients:

$$\boldsymbol{g}_e = \mathbb{E}_{(\boldsymbol{x}_e, \boldsymbol{y}_e)\sim\mathcal{D}_e}\nabla_\theta\ell\left(f_\theta(\boldsymbol{x}_e), \boldsymbol{y}_e\right). \tag{1}$$

When $\mathcal{E} = \{A, B\}$, IGA (Koyama & Yamaguchi, 2020) minimizes $||\boldsymbol{g}_A - \boldsymbol{g}_B||_2^2$; Fish (Shi et al., 2021) increases $\boldsymbol{g}_A \cdot \boldsymbol{g}_B$; AND-mask (Parascandolo et al., 2021) and others (Mansilla et al., 2021; Shahtalebi et al., 2021) update weights only when $\boldsymbol{g}_A$ and $\boldsymbol{g}_B$ point to the same direction.

Along with the increased computation cost, the main limitation of these gradient-based methods is the per-domain batch averaging of gradients, that removes more granular statistics; in particular, this averaging removes the information from pairwise interactions between gradients from samples in a same domain. In opposition, our new regularization for OOD generalization keeps extra information from individual gradients and matches across domains the domain-level gradient variances. In a nutshell, Fishr is similar to the covariance-based CORAL (Sun et al., 2016; Sun & Saenko, 2016) but in the gradient space rather than in the feature space.

3 FISHR

3.1 GRADIENT VARIANCE MATCHING

The **individual gradient** $\boldsymbol{g}_e^i = \nabla_\theta \ell \left(f_\theta(\boldsymbol{x}_e^i), \boldsymbol{y}_e^i \right)$ is the first-order derivative for the i-th data example $\left(\boldsymbol{x}_e^i, \boldsymbol{y}_e^i \right)$ from domain $e \in \mathcal{E}$ with respect to the weights θ. Previous methods have matched the gradient means $\boldsymbol{g}_e = \frac{1}{n_e} \sum_{i=1}^{n_e} \boldsymbol{g}_e^i$ for each domain $e \in \mathcal{E}$ of size $|\theta|$, usually computed for gradient descent during the learning procedure. Leveraging the full matrix $\boldsymbol{G}_e = [\boldsymbol{g}_e^i]_{i=1}^{n_e}$ of size $n_e \times |\theta|$, we compute the **domain-level gradient variance** vector of size $|\theta|$:

$$\boldsymbol{v}_e = \mathrm{Var}(\boldsymbol{G}_e) = \frac{1}{n_e - 1} \sum_{i=1}^{n_e} \left(\boldsymbol{g}_e^i - \boldsymbol{g}_e \right)^\top \left(\boldsymbol{g}_e^i - \boldsymbol{g}_e \right). \tag{2}$$

To reduce the distribution shifts in the network f_θ across domains, we bring the domain-level gradient variances $\{\boldsymbol{v}_e\}_{e \in \mathcal{E}}$ closer. Hence, our Fishr regularization is:

$$\mathcal{L}_{\text{Fishr}}(\theta) = \frac{1}{|\mathcal{E}|} \sum_{e \in \mathcal{E}} \| \boldsymbol{v}_e - \boldsymbol{v} \|_2^2, \tag{3}$$

the square of the euclidean distance between the gradient variance from the different domains $e \in \mathcal{E}$ and the mean gradient variance $\boldsymbol{v} = \frac{1}{|\mathcal{E}|} \sum_{e \in \mathcal{E}} \boldsymbol{v}_e$. Balanced with a hyperparameter coefficient $\lambda > 0$, this Fishr penalty complements the original ERM objective, *i.e.*, the empirical training risks:

$$\mathcal{L}(\theta) = \frac{1}{|\mathcal{E}|} \sum_{e \in \mathcal{E}} \mathcal{R}_e(\theta) + \lambda \mathcal{L}_{\text{Fishr}}(\theta). \tag{4}$$

Remark Gradients $\boldsymbol{g}_e^i$ were derived on all network weights θ. Yet, to reduce the memory and training costs in our experiments, they will often be computed only on a subset of θ, *e.g.*, only on classification weights ω. This approximation is discussed in Section 4.2.1 and Appendix C.2.2.

Motivations We choose to tackle gradient variances for two main reasons. The *first* reason is because this strategy aligns gradient distributions across domains — as discussed in Section 3.2. *Second and most importantly*, our approach is driven by the links between the gradient covariance, the Fisher Information and the Hessian: we show in Section 3.3 that Fishr aligns the domain-level Hessians and the domain-level loss landscapes at convergence. A prerequisite for these two motivations is to observe that $\boldsymbol{v}_e$ is the diagonal of the **gradient covariance matrix** $\boldsymbol{C}_e = \frac{1}{n_e - 1} \left(\boldsymbol{G}_e^\top \boldsymbol{G}_e - \frac{1}{n_e} \left(\boldsymbol{1}^\top \boldsymbol{G}_e \right)^\top \left(\boldsymbol{1}^\top \boldsymbol{G}_e \right) \right)$ of size $|\theta| \times |\theta|$. Compared to matching $\{\boldsymbol{C}_e\}_{e \in \mathcal{E}}$, this scales down the number of targeted components from $|\theta|^2$ to $|\theta|$. We empirically show in Appendix B.2.4 that ignoring or not the off-diagonal parts perform similarly. This approximation is mainly motivated by the empirical similarities between $\boldsymbol{C}$ and the Hessian, that will be highlighted in Section 3.3. Indeed, diagonally approximating the Hessian is common: *e.g.*, for OOD generalization (Parascandolo et al., 2021), optimization (LeCun et al., 2012; Kingma & Ba, 2014), continual learning (Kirkpatrick et al., 2017) and pruning (LeCun et al., 1990; Theis et al., 2018). This is based on the empirical evidence (Becker & Le Cun, 1988) that Hessians are diagonally dominant at the end of training. Our diagonal approximation is also motivated by the critical importance of $\mathrm{Tr}(\boldsymbol{C})$ (Jastrzebski et al., 2021; Faghri et al., 2020) to analyze the generalization properties of DNNs.

3.2 FISHR MATCHES THE DOMAIN-LEVEL GRADIENT DISTRIBUTIONS

The first motivation for Fishr is the independence between the gradient and the domain random variables. This is achieved by matching the diagonals of the covariance of the empirical gradient distributions $\left\{ \boldsymbol{g}_e^i \right\}_{i=1}^{n_e}$ across training domains $e \in \mathcal{E}$. Indeed, this is an efficient and well-suited method to align distributions. This was recently highlighted by the success of the covariance-based CORAL (Sun et al., 2016) on the DomainBed benchmark: matching covariance performed better than adversarial methods to align feature distributions. Therefore, this motivates the use of gradient variance: more complex strategies to align gradient distributions are best left for future works.

We now provide four — perhaps intuitive — reasons to enforce distributional invariance in gradients rather than in features. *First and foremost*, having similar domain-level gradient distributions is critical so that the DNN has shared properties across domains. Indeed, gradient disagreements and pairwise relations are key to the optimization procedure of DNNs: for instance, gradient confusion slows down convergence (Sankararaman et al., 2020) even though gradient diversity improves generalization (Yin et al., 2018). As a side note, the gradient mean can capture the average learning direction but can not capture these refined statistics. *Second*, gradients are more expressive and richer than features. Specifically, gradients were shown to better cluster semantically close inputs (Fort et al., 2019; He & Su, 2020). When comparing the features extracted for two inputs (Johnson et al., 2016), a small difference in activation may be multiplied by large subsequent weights and lead to distant predictions. On the contrary when comparing gradients, each activation is weighted by its true importance for the prediction (Charpiat et al., 2019). *Third*, gradients take into account the label Y, which appears as an argument for the loss ℓ. Hence, gradient-based approaches are 'label-aware' by design. In contrast, seminal feature-based methods were shown to fail in case of label shifts, because they do not consider Y (Johansson et al., 2019; Zhao et al., 2019). *Lastly*, matching gradient distributions also matches training risks, as motivated in V-REx (Krueger et al., 2021) for OOD generalization. Indeed, gradient amplitudes are directly weighted by the loss values. This is theoretically proved in Appendix A.3.2, and empirically validated in Appendix B.2.1: Fishr induces $|\mathcal{R}_A - \mathcal{R}_B|^2 \to 0$ when $\mathcal{E} = \{A, B\}$ for Colored MNIST.

3.3 Fishr reduces inconsistencies across the domain-level loss landscapes

In Section 3.3.1, we argue (based on previous works) and show (in Table 1) that Fishr matches the domain-level Hessians. The gradient covariance, computable efficiently with a unique backpropagation, serves as a proxy for the Hessian. Thus, we use the second moment of the first-order derivatives to regularize the second-order derivatives. Then, we justify in Section 3.3.2 why similar domain-level Hessians reduces inconsistencies in the loss landscape and improves generalization.

3.3.1 Fishr matches the domain-level Hessians

The Hessian matrix $\boldsymbol{H} = \sum_{i=1}^{n} \nabla_\theta^2 \ell \left(f_\theta(\boldsymbol{x}^i), \boldsymbol{y}^i \right)$ captures the second-order derivatives of the objective and is of key importance for deep learning methods. Yet, $\boldsymbol{H}$ is computationally demanding and can not be computed directly in general. Recent methods (Izmailov et al., 2018; Foret et al., 2021) tackle the Hessian indirectly by modifying the learning procedure. In contrast, we use the fact that the Fisher Information Matrix $\boldsymbol{F} = \sum_{i=1}^{n} \mathbb{E}_{\hat{\boldsymbol{y}} \sim P_\theta(\cdot|\boldsymbol{x}^i)} \left[\nabla_\theta \log p_\theta(\hat{\boldsymbol{y}}|\boldsymbol{x}^i) \nabla_\theta \log p_\theta(\hat{\boldsymbol{y}}|\boldsymbol{x}^i)^\top \right]$ (Fisher, 1922; C.R., 1945) approximates $\boldsymbol{H}$, with theoretically probably bounded errors under mild assumptions (Schraudolph, 2002). Yet, $\boldsymbol{F}$ remains costly as it demands one backpropagation per class. That's why most empirical works (*e.g.*, in compression (Frantar et al., 2021; Liu et al., 2021) and optimization (Dangel et al., 2021)) approximate $\boldsymbol{H}$ with the 'empirical' Fisher Information Matrix $\tilde{\boldsymbol{F}} = \boldsymbol{G}_e^\top \boldsymbol{G}_e = \sum_{i=1}^{n} \nabla_\theta \log p_\theta(\boldsymbol{y}^i|\boldsymbol{x}^i) \nabla_\theta \log p_\theta(\boldsymbol{y}^i|\boldsymbol{x}^i)^\top$ (Martens, 2014) where $p_\theta(\cdot|\boldsymbol{x})$ is the density predicted by f_θ on input $\boldsymbol{x}$. While $\boldsymbol{F}$ uses the model distribution $P_\theta(\cdot|X)$, $\tilde{\boldsymbol{F}}$ uses the data distribution $P(Y|X)$. Despite this key difference, $\tilde{\boldsymbol{F}}$ and $\boldsymbol{F}$ were shown to share the same structure and to be similar up to a scalar factor (Thomas et al., 2020). This was discussed in Li et al. (2020) and further highlighted even at early stages of training (before overfitting) in the Fig. 1 and the Appendix S3 of Singh & Alistarh (2020).

Critically, $\tilde{\boldsymbol{F}}$ is nothing else than the unnormalized uncentered covariance matrix when ℓ is the negative log-likelihood. Thus, $\boldsymbol{C}$ and $\tilde{\boldsymbol{F}}$ are equivalent (up to the multiplicative constant n) at any first-order stationary point: so $\boldsymbol{C} \propto \tilde{\boldsymbol{F}}$. Overall, this suggests that $\boldsymbol{C}$ and $\boldsymbol{H}$ are closely related. In our multi-domain framework, we define the domain-level matrices with the subscript e. Table 1 empirically confirms that matching $\{\text{Diag}(\boldsymbol{C}_e)\}_{e \in \mathcal{E}}$ with Fishr forces the domain-level Hessians $\{\text{Diag}(\boldsymbol{H}_e)\}_{e \in \mathcal{E}}$ to be aligned at convergence. This will be further validated in Fig. 3 and in Appendix B.2.2, in the classifier w_ω for computational reasons.

Table 1: **Invariance analysis at convergence in Colored MNIST** across the two training domains $\mathcal{E} = \{90\%, 80\%\}$. Details in Appendix B.1 and B.2.2. Compared to ERM, Fishr enforces invariance in Hessians ($\text{Diag}(\boldsymbol{H}_{90\%}) \approx \text{Diag}(\boldsymbol{H}_{80\%})$) by matching the gradient variance ($\text{Diag}(\boldsymbol{C}_{90\%}) \approx \text{Diag}(\boldsymbol{C}_{80\%})$).

	ERM	Fishr
$\|\text{Diag}(\boldsymbol{C}_{90\%} - \boldsymbol{C}_{80\%})\|_F^2$	3.1×10^{-3}	2.2×10^{-6}
$\|\text{Diag}(\boldsymbol{H}_{90\%} - \boldsymbol{H}_{80\%})\|_F^2$	2.6×10^{-4}	2.0×10^{-6}

Even so, we acknowledge that approximating $\boldsymbol{H}$ by $\tilde{\boldsymbol{F}}$ is not fully justified theoretically (Martens, 2014; Kunstner et al., 2019). One would hope that when overfitting occurs, $P_\theta(\cdot|X) \to P(Y|X)$ and then $\boldsymbol{F} \to \tilde{\boldsymbol{F}}$. Though, this requires strong assumptions such as χ^2 convergence (see Proposition 1 in Thomas et al. (2020)). In this paper, we trade off theoretical guarantees for computational efficiency and consider $\boldsymbol{C}$ and $\tilde{\boldsymbol{F}}$. Notably, Appendix B.2.5 shows that matching the diagonals of $\{\boldsymbol{C}_e\}_{e\in\mathcal{E}}$ or $\{\tilde{\boldsymbol{F}}_e\}_{e\in\mathcal{E}}$ — *i.e.*, centering or not the statistics — perform similarly.

3.3.2 Invariant Hessians for loss consistency

To understand why having similar domain-level Hessians at convergence improves generalization, we leverage the **inconsistency** formalism developed in AND-mask (Parascandolo et al., 2021). They argue that "patchwork solutions sewing together different strategies" for different domains may not generalize well: good weights should be optimal on all domains and "hard to vary" (Deutsch, 2011). They formalize this intuition with an inconsistency score $\mathcal{I}^\epsilon(\theta^*) = \max_{(A,B)\in\mathcal{E}^2} \max_{\theta\in N^\epsilon_{A,\theta^*}} |\mathcal{R}_B(\theta) - \mathcal{R}_A(\theta^*)|$ where $\theta \in N^\epsilon_{A,\theta^*}$ if there is a path in the weights space between θ and θ^* where the risk $\mathcal{R}_A$ remains in an $\epsilon > 0$ interval around $\mathcal{R}_A(\theta^*)$. $\mathcal{I}$ increases with conflicting geometries in the loss landscapes around θ^* as in Fig. 2: *i.e.*, when another 'close' solution θ is equivalent to the current solution θ^* in a domain A but yields different losses in B.

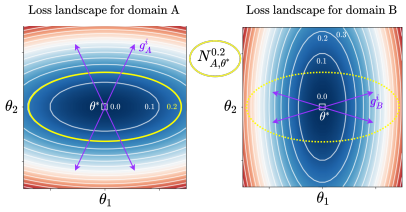

Figure 2: **Loss landscapes around an inconsistent** θ^* at convergence. $N^{0.2}_{A,\theta^*}$ contains weights θ for which $\mathcal{R}_A(\theta)$ is low (≤ 0.2) but $\mathcal{R}_B(\theta)$ is high (≥ 1.0). This inconsistency is due to conflicting domain-level loss landscapes, and is visible in the disagreements across the variances of gradients $\{\boldsymbol{g}^i_A\}_{i=1}^{n_A}$ and $\{\boldsymbol{g}^i_B\}_{i=1}^{n_B}$.

Moreover, the Hessian approximates the local curvature of the loss landscape around θ^*. Using this fact and a second-order Taylor expansion of the loss, we derive in Appendix A.1 a new upper bound of domain inconsistencies. Assuming that domain-level Hessians are co-diagonalizable for simplicity, inconsistency increases when an eigenvalue is small in $\boldsymbol{H}_A$ but large in $\boldsymbol{H}_B$: indeed, a small weight perturbation in the direction of the associated eigenvector would change the loss slightly in the domain A but drastically in B. In conclusion, "inconsistency is lowest when shapes [...] are similar" (Parascandolo et al., 2021), *i.e.*, when $\boldsymbol{H}_A = \boldsymbol{H}_B$. AND-mask minimizes $\mathcal{I}$ by zeroing out gradients with inconsistent directions across domains. However, this masking strategy introduces dead zones (Shahtalebi et al., 2021) in weights where the model could get stuck, ignores gradient magnitudes and empirically performs poorly with 'real' datasets from DomainBed. In place, **Fishr reduces inconsistencies in the loss landscapes across domains by matching the domain-level Hessians**, using domain-level gradient variances as a proxy.

Finally, we refer the readers to Appendix A.2 where we leverage the Neural Tangent Kernel (NTK) (Jacot et al., 2018) theory to further motivate the gradient variance matching during the optimization process — and not only at convergence. In brief, as $\boldsymbol{F}$ and the NTK matrices share the same non-zero eigenvalues, similar $\{\boldsymbol{C}_e\}_{e\in\mathcal{E}}$ during training reduce the simplicity bias by preventing the learning of different domain-dependent shortcuts at different training speeds: this favors a shared mechanism that predicts the same thing for the same reasons across domains.

4 Experiments

4.1 Proof of concept on Colored MNIST

The task in Colored MNIST (Arjovsky et al., 2019) is to predict whether the digit is below or above 5. Moreover, the labels are flipped with 25% probability (except in Appendix B.2.3). Critically, the digits' colors spuriously correlate with the labels: the correlation strength varies across the two training domains $\mathcal{E} = \{90\%, 80\%\}$. To test whether the model has learned to ignore the color, this correlation is reversed at test time. In brief, a biased model that only considers the color would have 10% test accuracy whereas an oracle model that perfectly predicts the shape would have 75%. As previously done in V-REx (Krueger et al., 2021), we **strictly** follow the IRM implementation and just replace the IRM penalty by our Fishr penalty. This means that we use the exact same MLP and hyperparameters, notably the same **two-stage scheduling** selected in IRM for the regularization strength λ, that is low until epoch 190 and then jumps to a large value: details in Appendix B.1.

Table 2 reports the accuracy averaged over 10 runs with standard deviation. In test, Fishr$_\theta$ (*i.e.*, applying Fishr on all weights θ) reaches 71.2%, and 70.2% when digits are grayscale. Moreover, computing the gradients only in the classifier w_ω performs almost as well (69.5% for Fishr$_\omega$) while reducing drastically the computational cost. Finally, Fishr$_\phi$ only in the features extractor ϕ works best in test, though it has lower train accuracy. These results highlight the effectiveness of gradient variance matching — even with standard hyperparameters — but should not be considered as a proof of Fishr superiority precisely because of the absence of hyperparameter search.

The main advantage of this synthetic dataset is the possibility of empirically validating some theoretical insights. For example, the training dynamics in Fig. 3 show that the domain-level Hessians get closer once the Fishr$_\omega$ gradient variance matching loss is activated after step 190. Consequently, this sharply increases test accuracy. This confirms insights from Section 3.3.1. Additional experiments can be found in Appendix B.2. Yet, the main drawback of Colored MNIST is its insufficiency to ensure generalization for real-world datasets. Overall, it should be considered as a first proof-of-concept.

Table 2: **Colored MNIST** results. All methods use hyperparameters optimized for IRM.

Method	Train acc.	Test acc.	Gray test acc.
ERM	86.4 ± 0.2	14.0 ± 0.7	71.0 ± 0.7
IRM	71.0 ± 0.5	65.6 ± 1.8	66.1 ± 0.2
V-REx	71.7 ± 1.5	67.2 ± 1.5	68.6 ± 2.2
Fishr$_\theta$	69.6 ± 0.9	71.2 ± 1.1	70.2 ± 0.7
Fishr$_\omega$	71.0 ± 0.9	69.5 ± 1.0	70.2 ± 1.1
Fishr$_\phi$	65.6 ± 1.3	73.8 ± 1.0	70.0 ± 0.9

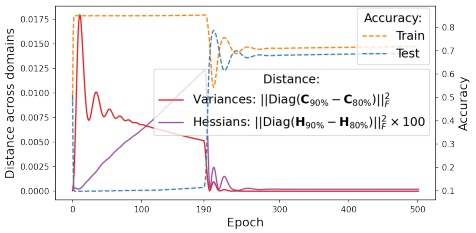

Figure 3: **Colored MNIST dynamics**. At epoch 190, λ strongly steps up: then the Fishr$_\omega$ regularization matches the domain-level gradient variances (red) and Hessians (purple) across $\mathcal{E} = \{90\%, 80\%\}$. This reduces train accuracy (orange) but increases test accuracy (blue) as the network learns to predict the digit's shape.

4.2 DomainBed benchmark

4.2.1 Implementation details

We conduct extensive experiments on **the DomainBed benchmark** (Gulrajani & Lopez-Paz, 2021) to rigorously compare the different strategies. In addition to the synthetic Colored MNIST (Arjovsky et al., 2019) and Rotated MNIST (Ghifary et al., 2015), the multi-domain image classification datasets are the real-world VLCS (Fang et al., 2013), PACS (Li et al., 2017), OfficeHome (Venkateswara et al., 2017), TerraIncognita (Beery et al., 2018) and DomainNet (Peng et al., 2019).

Critically, we systematically apply Fishr only in the classifier w_ω in DomainBed. Indeed, keeping individual gradients in memory for ϕ from a ResNet-50 was impossible for computational reasons. This is motivated by the similar performances of Fishr$_\theta$ and Fishr$_\omega$ in Section 4.1. Moreover, as highlighted in Appendix C.2.2, this relaxation may improve results for real-world datasets. Indeed, while Colored MNIST is a correlation shift challenge, the other datasets mostly demonstrate diversity shifts where "each domain represents a certain spectrum of diversity in data" (Ye et al., 2021). Then, as the pixels distribution are quite different across domains, low-level layers may need to adapt to these domain-dependent peculiarities. Moreover, this last-layer approximation is consistent with the IRM condition (Arjovsky et al., 2019) and is classical for unsupervised domain adaptation (Ganin et al., 2016). Finally, if we used all weights $\theta = (\phi, \omega)$ to compute gradient variances, the invariance in w_ω may be overshadowed by Φ_ϕ due to $|\omega| \ll |\phi|$.

Fishr relies on three **hyperparameters**. *First*, the λ coefficient controls the regularization strength: with $\lambda = 0$ we recover ERM while a high λ may cause underfitting. *Second* the warmup iteration defines the step at which we activate the regularization. This warmup strategy is taken from previous works such as IRM (Arjovsky et al., 2019), V-REx (Krueger et al., 2021) or Spectral Decoupling (Pezeshki et al., 2020). Before that step, the DNN is trained with ERM to learn predictive features. After that step, the Fishr regularization encourages the DNN to have invariant gradient variances. *Lastly*, the domain-level gradient variances are more accurate when estimated over more data points. Rather than increasing the batch size, we follow Le Roux et al. (2011) and leverage an exponential moving average for computing stable gradient variances. Therefore our third hyperparameter is the coefficient γ that controls the update speed: at step t, we match $\bar{\boldsymbol{v}}_e^t = \gamma \bar{\boldsymbol{v}}_e^{t-1} + (1-\gamma)\boldsymbol{v}_e^t$ rather than of $\boldsymbol{v}_e^t$ from Eq. 2. The closer γ is to 1, the smoother the variance is along training. $\bar{\boldsymbol{v}}_e^{t-1}$

from previous step $t-1$ is 'detached' from the computational graph. Similar strategies have already been used for OOD generalization (Nam et al., 2020; Blanchard et al., 2021; Zhang et al., 2021). The memory overhead is $(|\mathcal{E}| \times |\omega|)$. Finally, we study by **ablation** the importance of this warmup strategy and this γ in Appendices C.2.1 and C.2.2.

Algorithm 1: Training procedure for Fishr on DomainBed.

Input: DNN f_θ, observations $\mathcal{D}_e = \left\{\left(\boldsymbol{x}_e^i, \boldsymbol{y}_e^i\right)\right\}_{i=1}^{n_e}$ for domains $e \in \mathcal{E}$, regularization weight λ, warmup iteration i_{warmup}, exponential moving average γ and batch size b_s

Initialize moving averages: $\forall e \in \mathcal{E}, \boldsymbol{v}_e^{\text{mean}} \leftarrow 0$

for *iter* ***from*** 1 ***to*** *#iters* **do**

`/* Step 1: standard ERM procedure */`

  for $e \in \mathcal{E}$ **do**

    Randomly select batch: $\{(\boldsymbol{x}_e^i, \boldsymbol{y}_e^i)\}_{i \in \mathcal{B}}$ of size b_s

    Compute individual predictions: $\forall i \in \mathcal{B}, \hat{\boldsymbol{y}}_e^i \leftarrow f_\theta(\boldsymbol{x}_e^i)$

    Compute domain-level empirical risks: $\mathcal{R}_e(\theta) \leftarrow \sum_{i \in \mathcal{B}} \ell\left(\hat{\boldsymbol{y}}_e^i, \boldsymbol{y}_e^i\right)$

  $\mathcal{L}(\theta) = \frac{1}{|\mathcal{E}|} \sum_{e \in \mathcal{E}} \mathcal{R}_e(\theta)$

`/* Step 2: gradient variances in classifier */`

  for $e \in \mathcal{E}$ **do**

    Compute individual gradients in w_ω with BackPACK: $\forall i \in \mathcal{B}, \boldsymbol{g}_e^i \leftarrow \nabla_\omega \ell\left(\hat{\boldsymbol{y}}_e^i, \boldsymbol{y}_e^i\right)$

    Compute domain-level gradient variances $\boldsymbol{v}_e$ from Eq. 2

    Update moving average: $\boldsymbol{v}_e^{\text{mean}} = \boldsymbol{v}_e \leftarrow \gamma \boldsymbol{v}_e^{\text{mean}} + (1-\gamma)\boldsymbol{v}_e^{\text{iter}}$

  if *iter* $\geq i_{warmup}$ **then**

    $\mathcal{L}(\theta)$ += $\lambda \mathcal{L}_{\text{Fishr}}(\theta)$ from Eq. 3

`/* Step 3: gradient descent in the whole network */`

  Backpropagate gradients $\nabla_\theta \mathcal{L}(\theta)$ in the network f_θ with standard PyTorch

Fishr is simple to implement (see the Algorithm 1) using the BackPACK (Dangel et al., 2020) package. While PyTorch (Paszke et al., 2019) can compute efficiently batch gradients, BackPACK optimizes the computation of individual gradients, sample per sample, at almost no time overhead. Thus, Fishr is also at low computational costs. For example, on PACS (7 classes and $|\omega| = 14,343$) with a ResNet-50 and batch size 32, Fishr induces an overhead in memory of +0.2% and in training time of +2.7% (with a Tesla V100) compared to ERM; on the larger-scale DomainNet (345 classes and $|\omega| = 706,905$), the overhead is +7.0% in memory and +6.5% in training time. As a side note, keeping the full covariance of size $|\omega|^2 \approx 5 \times 10^8$ on DomainNet would not have been possible. In contrast, Fish is impractical as $|\mathcal{E}|$ times longer to train than ERM.

To limit access to test domain, the framework enforces that all methods are trained with only 20 different configurations of hyperparameters and for the same number of steps. Results are averaged over three trials. This experimental setup is further described in Appendix C.1; the hyperparameter distributions are analyzed in Appendix C.2.3; results are detailed per dataset in Appendix C.3.

4.2.2 Results

As performances depend heavily on the hyperparameter choice, the model selection strategy is critical. Table 3 summarizes the results on DomainBed using the 'Oracle' model selection: the validation set follows the same distribution as the test domain. ERM remains a strong baseline and all previous methods are far from the best score on at least one dataset. Moreover, 'invariant predictors' and 'gradient masking' approaches perform poorly on 'real' datasets. Contrarily, Fishr is the only method to systematically perform better than ERM on all 'real' datasets: the differences are over standard errors on VLCS (78.2% vs. 77.6%), OfficeHome (68.2% vs. 66.4%) and on the larger-scale DomainNet (41.8% vs. 41.3%). On synthetic datasets, Fishr outperforms ERM on Colored MNIST (68.8% vs. 57.8%) and performs similarly on Rotated MNIST. After averaging, Fishr outperforms all concurrent approaches and reaches 70.8% vs. 69.1% for Fish (see Appendix C.2.2 for further comparisons with gradient-mean approaches). Most importantly, Fishr is consistently among the best methods, with a mean ranking of 3.9 and a median ranking of second.

Table 3: **Test-domain model selection**. We format **first**, second and worse than ERM results.

Algorithm	Accuracy (↑)								Ranking (↓)				
	CMNIST	RMNIST	VLCS	PACS	OfficeHome	TerraInc	DomainNet	Avg	Arith. mean	Geom. mean	Median	Best	Worst
ERM	57.8 ± 0.2	97.8 ± 0.1	77.6 ± 0.3	86.7 ± 0.3	66.4 ± 0.5	53.0 ± 0.3	41.3 ± 0.1	68.7	9.1	8.1	8	3	16
IRM	67.7 ± 1.2	97.5 ± 0.2	76.9 ± 0.6	84.5 ± 1.1	63.0 ± 2.7	50.5 ± 0.7	28.0 ± 5.1	66.9	14.7	12.4	16	2	18
GroupDRO	61.1 ± 0.9	97.9 ± 0.1	77.4 ± 0.5	87.1 ± 0.1	66.2 ± 0.6	52.4 ± 0.1	33.4 ± 0.3	67.9	8.6	7.5	8	3	15
Mixup	58.4 ± 0.2	98.0 ± 0.1	78.1 ± 0.3	86.8 ± 0.3	68.0 ± 0.2	**54.4** ± 0.3	39.6 ± 0.1	69.0	5.3	3.9	4	**1**	13
MLDG	58.2 ± 0.4	97.8 ± 0.1	77.5 ± 0.1	86.8 ± 0.4	66.6 ± 0.3	52.0 ± 0.1	41.6 ± 0.1	68.7	9.1	8.2	9	4	14
CORAL	58.6 ± 0.5	98.0 ± 0.0	77.7 ± 0.2	87.1 ± 0.5	**68.4** ± 0.2	52.8 ± 0.2	41.8 ± 0.1	69.2	4.6	3.4	3	**1**	**10**
MMD	63.3 ± 1.3	98.0 ± 0.1	77.9 ± 0.1	**87.2** ± 0.1	66.2 ± 0.3	52.0 ± 0.4	23.5 ± 9.4	66.9	7.0	4.9	6	**1**	18
DANN	57.0 ± 1.0	97.9 ± 0.1	79.7 ± 0.5	85.2 ± 0.2	65.3 ± 0.8	50.6 ± 0.4	38.3 ± 0.1	67.7	11.9	9.6	15	2	18
CDANN	59.5 ± 2.0	97.9 ± 0.0	**79.9** ± 0.2	85.8 ± 0.8	65.3 ± 0.5	50.8 ± 0.6	38.5 ± 0.2	68.2	9.6	7.4	10	**1**	15
MTL	57.6 ± 0.3	97.9 ± 0.1	77.7 ± 0.5	86.7 ± 0.2	66.5 ± 0.4	52.2 ± 0.4	40.8 ± 0.1	68.5	8.4	7.8	7	5	17
SagNet	58.2 ± 0.3	97.9 ± 0.0	77.6 ± 0.1	86.4 ± 0.4	67.5 ± 0.2	52.5 ± 0.4	40.8 ± 0.2	68.7	8.0	7.2	6	4	14
ARM	63.2 ± 0.7	**98.1** ± 0.1	77.8 ± 0.3	85.8 ± 0.2	64.8 ± 0.4	51.2 ± 0.5	36.0 ± 0.2	68.1	9.9	7.5	12	**1**	17
V-REx	67.0 ± 1.3	97.9 ± 0.1	78.1 ± 0.2	**87.2** ± 0.6	65.7 ± 0.3	51.4 ± 0.5	30.1 ± 3.7	68.2	7.7	5.5	5	**1**	16
RSC	58.5 ± 0.5	97.6 ± 0.1	77.8 ± 0.6	86.2 ± 0.5	66.5 ± 0.6	52.1 ± 0.2	38.9 ± 0.6	68.2	9.9	9.4	9	6	15
AND-mask	58.6 ± 0.4	97.5 ± 0.0	76.4 ± 0.4	86.4 ± 0.4	66.1 ± 0.2	49.8 ± 0.4	37.9 ± 0.6	67.5	13.4	13.1	12	10	18
SAND-mask	62.3 ± 1.0	97.4 ± 0.1	76.2 ± 0.5	85.9 ± 0.4	65.9 ± 0.5	50.2 ± 0.1	32.2 ± 0.6	67.2	14.3	13.5	15	6	18
Fish	61.8 ± 0.8	97.9 ± 0.1	77.8 ± 0.6	85.8 ± 0.6	66.0 ± 2.9	50.8 ± 0.4	**43.4** ± 0.3	69.1	8.4	6.6	7	**1**	14
Fishr	**68.8** ± 1.4	97.8 ± 0.1	78.2 ± 0.2	86.9 ± 0.2	68.2 ± 0.2	53.6 ± 0.4	41.8 ± 0.2	**70.8**	**3.9**	**2.8**	**2**	**1**	12

Table 4: **Training-domain model selection**. We format **first**, second and worse than ERM results.

Algorithm	Accuracy (↑)								Ranking (↓)				
	CMNIST	RMNIST	VLCS	PACS	OfficeHome	TerraInc	DomainNet	Avg	Arith. mean	Geom. mean	Median	Best	Worst
ERM	51.5 ± 0.1	98.0 ± 0.0	77.5 ± 0.4	85.5 ± 0.2	66.5 ± 0.3	46.1 ± 1.8	40.9 ± 0.1	66.6	7.0	5.9	7	2	12
IRM	52.0 ± 0.1	97.7 ± 0.1	78.5 ± 0.5	83.5 ± 0.8	64.3 ± 2.2	47.6 ± 0.8	33.9 ± 2.8	65.4	10.7	8.5	14	3	18
GroupDRO	52.1 ± 0.0	98.0 ± 0.0	76.7 ± 0.6	84.4 ± 0.8	66.0 ± 0.7	43.2 ± 1.1	33.3 ± 0.2	64.8	11.3	8.4	14	2	18
Mixup	52.1 ± 0.2	98.0 ± 0.1	77.4 ± 0.6	84.6 ± 0.6	68.1 ± 0.3	47.9 ± 0.8	39.2 ± 0.1	66.7	5.7	4.2	3	2	13
MLDG	51.5 ± 0.1	97.9 ± 0.0	77.2 ± 0.4	84.9 ± 1.0	66.8 ± 0.6	47.7 ± 0.9	41.2 ± 0.1	66.7	8.0	7.0	8	3	15
CORAL	51.5 ± 0.1	98.0 ± 0.1	**78.8** ± 0.6	86.2 ± 0.3	**68.7** ± 0.3	47.6 ± 1.0	41.5 ± 0.1	**67.5**	**3.6**	**2.5**	**2**	**1**	12
MMD	51.5 ± 0.2	97.9 ± 0.0	77.5 ± 0.9	84.6 ± 0.5	66.3 ± 0.1	42.2 ± 1.6	23.4 ± 9.5	63.3	12.3	11.8	10	8	18
DANN	51.5 ± 0.3	97.8 ± 0.1	78.6 ± 0.4	83.6 ± 0.4	65.9 ± 0.6	46.7 ± 0.5	38.3 ± 0.1	66.1	10.3	8.8	12	2	16
CDANN	51.7 ± 0.1	97.9 ± 0.1	77.5 ± 0.1	82.6 ± 0.9	65.8 ± 1.3	45.8 ± 1.6	38.3 ± 0.3	65.6	11.1	10.7	10	8	18
MTL	51.4 ± 0.1	97.9 ± 0.0	77.2 ± 0.4	84.6 ± 0.5	66.4 ± 0.5	45.6 ± 1.2	40.6 ± 0.1	66.2	10.9	10.2	10	6	17
SagNet	51.7 ± 0.0	98.0 ± 0.0	77.8 ± 0.5	**86.3** ± 0.2	68.1 ± 0.1	**48.6** ± 1.0	40.3 ± 0.1	67.2	4.0	3.0	3	**1**	**8**
ARM	**56.2** ± 0.2	**98.2** ± 0.1	77.6 ± 0.3	85.1 ± 0.4	64.8 ± 0.3	45.5 ± 0.3	35.5 ± 0.2	66.1	8.7	5.6	9	**1**	17
V-REx	51.8 ± 0.1	97.9 ± 0.1	78.3 ± 0.2	84.9 ± 0.6	66.4 ± 0.6	46.4 ± 0.6	33.6 ± 2.9	65.6	8.3	7.7	8	4	15
RSC	51.7 ± 0.2	97.6 ± 0.1	77.1 ± 0.5	85.2 ± 0.9	65.5 ± 0.9	46.6 ± 1.0	38.9 ± 0.5	66.1	11.4	10.6	9	6	17
AND-mask	51.3 ± 0.2	97.6 ± 0.1	78.1 ± 0.9	84.4 ± 0.9	65.6 ± 0.4	44.6 ± 0.3	37.2 ± 0.6	65.5	13.6	12.7	15	5	18
SAND-mask	51.8 ± 0.2	97.4 ± 0.1	77.4 ± 0.2	84.6 ± 0.9	65.8 ± 0.4	42.9 ± 1.7	32.1 ± 0.6	64.6	13.4	12.7	13	6	18
Fish	51.6 ± 0.1	98.0 ± 0.0	77.8 ± 0.3	85.5 ± 0.3	68.6 ± 0.4	45.1 ± 1.3	**42.7** ± 0.2	67.1	5.6	3.8	3	**1**	14
Fishr	52.0 ± 0.2	97.8 ± 0.0	77.8 ± 0.1	85.5 ± 0.4	67.8 ± 0.1	47.4 ± 1.6	41.7 ± 0.0	67.1	5.6	4.8	5	2	13

In Table 4, the validation set is formed by collecting 20% of each training domain. With this 'Training' model selection, Fishr performs better than ERM on all 'real' datasets (over standard errors for OfficeHome and DomainNet), except for PACS where the two reach 85.5%. In average, Fishr (67.1%) finishes third and is above most methods such as V-REx (65.6%). Fishr median ranking is fifth, with a mean ranking of 5.6.

Limitations Although Fishr remains stronger than ERM in the 'Training' setup, the improvements are smaller than in 'Oracle'. Indeed, besides the arguably low number of hyperparameter trials (20), the 'Training' setup suffers from underspecification: "predictors with equivalently strong held-out performance in the training domain [...] can behave very differently" in test (D'Amour et al., 2020). This underspecification favors low regularization thus low values of λ. To select the model with the best generalization properties, future benchmarks may consider the training calibration (Wald et al., 2021) rather than merely selecting the model with the best training accuracy.

5 Conclusion

In this paper, we addressed the task of out-of-distribution generalization for classification in computer vision. Motivated by the empirical success of CORAL and the inconsistency formalism from Parascandolo et al. (2021), we derive a new and simple regularization — Fishr — that matches the gradient variances across domains as a proxy for matching domain-level Hessians. This reaches state-of-the-art performances on DomainBed when samples from the test domain are available for model selection. Our empirical experiments suggest that Fishr would consistently improve a deep classifier in real-world applications when dealing with data from multiple domains. More generally, the criterion of domain invariance in gradients opens up new perspectives: for example, future work could consider adversarial strategies (Goodfellow et al., 2014) to align gradient distributions.

Reproducibility statement To facilitate reproducibility, our code is available at https://anonymous.4open.science/r/fishr-anonymous-EBB6/. We followed standard experimental protocols from previous works, with restricted hyperparameter search and controlled model selection. Notably, we included Fishr in the DomainBed benchmark. Moreover, we systematically performed multiple runs and reported mean and standard deviations. We have proposed an empirical approach, justified with arguments from previous theoretical works and our new upper bound for domain inconsistencies in Appendix A.1. Some are proven for the linear case in Appendix A.3. Finally, these theoretical insights are validated empirically on the Colored MNIST dataset, notably in Fig. 3 and Appendix B.2.

REFERENCES

Faruk Ahmed, Yoshua Bengio, Harm van Seijen, and Aaron Courville. Systematic generalisation with group invariant predictions. In *ICLR*, 2021. (page 3).

Martin Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz. Invariant risk minimization. *arXiv preprint*, 2019. (pages 1, 2, 3, 6, 7, 18, 21, 22, 24).

Sue Becker and Yann Le Cun. Improving the convergence of back-propagation learning with second order methods. In *Connectionist models summer school*, 1988. (page 4).

Sara Beery, Grant Van Horn, and Pietro Perona. Recognition in terra incognita. In *ECCV*, 2018. (pages 7, 22).

Gilles Blanchard, Gyemin Lee, and Clayton Scott. Generalizing from several related classification tasks to a new unlabeled sample. In *NeurIPS*, 2011. (page 1).

Gilles Blanchard, Aniket Anand Deshmukh, Urun Dogan, Gyemin Lee, and Clayton Scott. Domain generalization by marginal transfer learning. *Journal of Machine Learning Research*, 2021. (pages 8, 21, 23).

Shiyu Chang, Yang Zhang, Mo Yu, and Tommi Jaakkola. Invariant rationalization. In *ICML*, 2020. (page 3).

Guillaume Charpiat, Nicolas Girard, Loris Felardos, and Yuliya Tarabalka. Input similarity from the neural network perspective. In *NeurIPS*, 2019. (page 5).

Rao C.R. Information and accuracy attainable in the estimation of statistical parameters. In *Bulletin of the Calcutta Mathematical Society*, 1945. (page 5).

Elliot Creager, Jörn-Henrik Jacobsen, and Richard Zemel. Environment inference for invariant learning. In *ICML*, 2021. (page 3).

Alexander D'Amour, Katherine Heller, Dan Moldovan, Ben Adlam, Babak Alipanahi, Alex Beutel, Christina Chen, Jonathan Deaton, Jacob Eisenstein, Matthew D Hoffman, et al. Underspecification presents challenges for credibility in modern machine learning. *arXiv preprint*, 2020. (page 9).

Felix Dangel, Frederik Kunstner, and Philipp Hennig. BackPACK: Packing more into backprop. In *ICLR*, 2020. (pages 2, 8, 19).

Felix Dangel, Lukas Tatzel, and Philipp Hennig. Vivit: Curvature access through the generalized gauss-newton's low-rank structure. *arXiv preprint*, 2021. (page 5).

Alex J DeGrave, Joseph D Janizek, and Su-In Lee. Ai for radiographic covid-19 detection selects shortcuts over signal. *Nature Machine Intelligence*, 2021. (page 1).

D. Deutsch. The beginning of infinity: Explanations that transform the world. *Penguin UK*, 2011. (page 6).

Zhengming Ding and Yun Fu. Deep domain generalization with structured low-rank constraint. In *TIP*, 2017. (page 3).

Laurent Dinh, Razvan Pascanu, Samy Bengio, and Yoshua Bengio. Sharp minima can generalize for deep nets. In *ICML*, 2017. (page 19).

Yunshu Du, Wojciech M Czarnecki, Siddhant M Jayakumar, Mehrdad Farajtabar, Razvan Pascanu, and Balaji Lakshminarayanan. Adapting auxiliary losses using gradient similarity. *arXiv preprint*, 2018. (page 3).

Fartash Faghri, David Duvenaud, David J Fleet, and Jimmy Ba. A study of gradient variance in deep learning. *arXiv preprint*, 2020. (page 4).

Chen Fang, Ye Xu, and Daniel N Rockmore. Unbiased metric learning: On the utilization of multiple datasets and web images for softening bias. In *ICCV*, 2013. (pages 7, 22).

Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *ICML*, 2017. (page 3).

Ronald A Fisher. On the mathematical foundations of theoretical statistics. *Philosophical Transactions of the Royal Society of London.*, 1922. (pages 2, 5).

Pierre Foret, Ariel Kleiner, Hossein Mobahi, and Behnam Neyshabur. Sharpness-aware minimization for efficiently improving generalization. In *ICLR*, 2021. (page 5).

Stanislav Fort, Paweł Krzysztof Nowak, Stanislaw Jastrzebski, and Srini Narayanan. Stiffness: A new perspective on generalization in neural networks. *arXiv preprint*, 2019. (page 5).

Elias Frantar, Eldar Kurtic, and Dan Alistarh. Efficient matrix-free approximations of second-order information, with applications to pruning and optimization. *arXiv preprint*, 2021. (page 5).

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*, 2016. (pages 1, 3, 7, 21).

Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard Zemel, Wieland Brendel, Matthias Bethge, and Felix A Wichmann. Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2020. (page 1).

Muhammad Ghifary, W Bastiaan Kleijn, Mengjie Zhang, and David Balduzzi. Domain generalization for object recognition with multi-task autoencoders. In *ICCV*, 2015. (pages 7, 17, 22).

Mingming Gong, Kun Zhang, Tongliang Liu, Dacheng Tao, Clark Glymour, and Bernhard Schölkopf. Domain adaptation with conditional transferable components. In *ICML*, 2016. (page 3).

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *NeurIPS*, 2014. (page 9).

Ishaan Gulrajani and David Lopez-Paz. In search of lost domain generalization. In *ICLR*, 2021. (pages 1, 2, 3, 7, 20, 21).

Ruocheng Guo, Pengchuan Zhang, Hao Liu, and Emre Kiciman. Out-of-distribution prediction with invariant risk minimization: The limitation and an effective fix. *arXiv preprint*, 2021. (page 3).

Guy Gur-Ari, Daniel A Roberts, and Ethan Dyer. Gradient descent happens in a tiny subspace. *arXiv preprint*, 2018. (page 17).

Hangfeng He and Weijie Su. The local elasticity of neural networks. In *ICLR*, 2020. (page 5).

Tom Heskes. On "natural" learning and pruning in multilayered perceptrons. *Neural Computation*, 2000. (page 20).

Zeyi Huang, Haohan Wang, Eric P Xing, and Dong Huang. Self-challenging improves cross-domain generalization. In *ECCV*, 2020. (page 21).

Daksh Idnani and Jonathan C Kao. Learning robust representations with score invariant learning. In *ICML UDL Workshop*, 2020. (page 3).

Pavel Izmailov, Dmitrii Podoprikhin, Timur Garipov, Dmitry Vetrov, and Andrew Gordon Wilson. Averaging weights leads to wider optima and better generalization. In *UAI*, 2018. (page 5).

Arthur Jacot, Franck Gabriel, and Clement Hongler. Neural tangent kernel: Convergence and generalization in neural networks. In *NeurIPS*, 2018. (pages 6, 17).

Stanislaw Jastrzebski, Devansh Arpit, Oliver Astrand, Giancarlo B Kerg, Huan Wang, Caiming Xiong, Richard Socher, Kyunghyun Cho, and Krzysztof J Geras. Catastrophic fisher explosion: Early phase fisher matrix impacts generalization. In *ICML*, 2021. (page 4).

Khurram Javed, Martha White, and Yoshua Bengio. Learning causal models online. *arXiv preprint*, 2020. (page 3).

Fredrik D Johansson, David Sontag, and Rajesh Ranganath. Support and invertibility in domain-invariant representations. In *AISTATS*, 2019. (pages 3, 5).

Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *ECCV*, 2016. (page 5).

Dimitris Kalimeris, Gal Kaplun, Preetum Nakkiran, Benjamin Edelman, Tristan Yang, Boaz Barak, and Haofeng Zhang. Sgd on neural networks learns functions of increasing complexity. In *NeurIPS*, 2019. (page 1).

Pritish Kamath, Akilesh Tangella, Danica Sutherland, and Nathan Srebro. Does invariant risk minimization capture invariance? In *AISTATS*, 2021. (page 3).

Ryo Karakida, Shotaro Akaho, and Shun-ichi Amari. Pathological spectra of the fisher information metric and its variants in deep neural networks. *arXiv preprint*, 2019. (page 17).

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint*, 2014. (pages 4, 18, 21).

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. In *PNAS*, 2017. (page 4).

Pang Wei Koh, Shiori Sagawa, Henrik Marklund, Sang Michael Xie, Marvin Zhang, Akshay Balsubramani, Weihua Hu, Michihiro Yasunaga, Richard Lanas Phillips, Irena Gao, et al. Wilds: A benchmark of in-the-wild distribution shifts. *arXiv preprint*, 2020. (pages 2, 3).

Dmitry Kopitkov and Vadim Indelman. Neural spectrum alignment: Empirical study. *arXiv preprint*, 2019. (page 17).

Masanori Koyama and Shoichiro Yamaguchi. Out-of-distribution generalization with maximal invariant predictor. *arXiv preprint*, 2020. (pages 3, 20, 21, 23).

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *NeurIPS*, 2012. (page 1).

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 *ICML*, 2021. (pages 1, 3, 5, 6, 7, 16, 18, 21, 23).

Frederik Kunstner, Philipp Hennig, and Lukas Balles. Limitations of the empirical fisher approximation for natural gradient descent. In *NeurIPS*, 2019. (page 6).

Nicolas Le Roux, Yoshua Bengio, and Andrew Fitzgibbon. Improving first and second-order methods by modeling uncertainty. *Optimization for Machine Learning*, 2011. (pages 7, 22, 25).

Yann LeCun, J. S. Denker, Sara A. Solla, R. E. Howard, and L.D. Jackel. Optimal brain damage. In *NeurIPS*, 1990. (page 4).

Yann LeCun, Corinna Cortes, and Chris Burges. Mnist handwritten digit database, 2010. (pages 18, 22).

Yann LeCun, Léon Bottou, Genevieve B. Orr, and Klaus-Robert Müller. Efficient backprop. In *Neural Networks*. 2012. (page 4).

Bo Li, Yezhen Wang, Shanghang Zhang, Dongsheng Li, Kurt Keutzer, Trevor Darrell, and Han Zhao. Learning invariant representations and risks for semi-supervised domain adaptation. In *CVPR*, 2021. (page 3).

Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy M Hospedales. Deeper, broader and artier domain generalization. In *ICCV*, 2017. (pages 7, 22).

Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy Hospedales. Learning to generalize: Meta-learning for domain generalization. In *AAAI*, 2018a. (page 21).

Haoliang Li, Sinno Jialin Pan, Shiqi Wang, and Alex C Kot. Domain generalization with adversarial feature learning. In *CVPR*, 2018b. (pages 3, 21).

Xinyan Li, Qilong Gu, Yingxue Zhou, Tiancong Chen, and Arindam Banerjee. Hessian based analysis of sgd for deep nets: Dynamics and generalization. In *SIAM*, 2020. (pages 5, 19).

Ya Li, Mingming Gong, Xinmei Tian, Tongliang Liu, and Dacheng Tao. Domain generalization via conditional invariant representations. In *AAAI*, 2018c. (pages 3, 21).

Liyang Liu, Shilong Zhang, Zhanghui Kuang, Aojun Zhou, Jing-Hao Xue, Xinjiang Wang, Yimin Chen, Wenming Yang, Qingmin Liao, and Wayne Zhang. Group fisher pruning for practical network compression. In *ICML*, 2021. (page 5).

Mingsheng Long, Jianmin Wang, Guiguang Ding, Jiaguang Sun, and Philip S Yu. Transfer joint matching for unsupervised domain adaptation. In *CVPR*, 2014. (page 3).

David Lopez-Paz and Marc' Aurelio Ranzato. Gradient episodic memory for continual learning. In *NeurIPS*, 2017. (page 3).

Wesley J Maddox, Shuai Tang, Pablo Garcia Moreno, Andrew Gordon Wilson, and Andreas Damianou. On transfer learning via linearized neural networks. In *NeurIPS workshop*, 2019. (page 17).

Massimiliano Mancini, Samuel Rota Bulo, Barbara Caputo, and Elisa Ricci. Best sources forward: domain generalization through source-specific nets. In *ICIP*, 2018. (page 3).

Lucas Mansilla, Rodrigo Echeveste, Diego H. Milone, and Enzo Ferrante. Domain generalization via gradient surgery. In *ICCV*, 2021. (page 3).

James Martens. New insights and perspectives on the natural gradient method. *arXiv preprint*, 2014. (pages 5, 6).

James Martens and Roger Grosse. Optimizing neural networks with kronecker-factored approximate curvature. In *ICML*, 2015. (page 20).

Krikamol Muandet, David Balduzzi, and Bernhard Schölkopf. Domain generalization via invariant feature representation. In *ICML*, 2013. (pages 1, 3).

Hyeonseob Nam, HyunJae Lee, Jongchan Park, Wonjun Yoon, and Donggeun Yoo. Reducing domain gap by reducing style bias. In *CVPR*, 2021. (page 21).

Junhyun Nam, Hyuntak Cha, Sungsoo Ahn, Jaeho Lee, and Jinwoo Shin. Learning from failure: De-biasing classifier from biased classifier. In *NeurIPS*, 2020. (pages 8, 23).

Giambattista Parascandolo, Alexander Neitz, Antonio Orvieto, Luigi Gresele, and Bernhard Schölkopf. Learning explanations that are hard to vary. In *ICLR*, 2021. (pages 2, 3, 4, 6, 9, 16, 21, 25).

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *NeurIPS*, 2019. (page 8).

Judea Pearl. *Causality*. Cambridge university press, 2009. (page 3).

Xingchao Peng, Qinxun Bai, Xide Xia, Zijun Huang, Kate Saenko, and Bo Wang. Moment matching for multi-source domain adaptation. In *ICCV*, 2019. (pages 7, 22).

Jonas Peters, Peter Bühlmann, and Nicolai Meinshausen. Causal inference by using invariant prediction: identification and confidence intervals. *Journal of the Royal Statistical Society*, 2016. (pages 1, 3).

Mohammad Pezeshki, Sékou-Oumar Kaba, Yoshua Bengio, Aaron Courville, Doina Precup, and Guillaume Lajoie. Gradient starvation: A learning proclivity in neural networks. *arXiv preprint*, 2020. (pages 7, 17).

Michael Roberts, Derek Driggs, Matthew Thorpe, Julian Gilbey, Michael Yeung, Stephan Ursprung, Angelica I Aviles-Rivero, Christian Etmann, Cathal McCague, Lucian Beer, et al. Common pitfalls and recommendations for using machine learning to detect and prognosticate for covid-19 using chest radiographs and ct scans. *Nature Machine Intelligence*, 2021. (page 1).

Mateo Rojas-Carulla, Bernhard Schölkopf, Richard Turner, and Jonas Peters. Invariant models for causal transfer learning. *The Journal of Machine Learning Research*, 2018. (page 3).

Elan Rosenfeld, Pradeep Kumar Ravikumar, and Andrej Risteski. The risks of invariant risk minimization. In *ICLR*, 2021. (page 3).

Shiori Sagawa, Pang Wei Koh, Tatsunori B. Hashimoto, and Percy Liang. Distributionally robust neural networks. In *ICLR*, 2020a. (pages 3, 21).

Shiori Sagawa, Aditi Raghunathan, Pang Wei Koh, and Percy Liang. An investigation of why overparameterization exacerbates spurious correlations. In *ICML*, 2020b. (page 3).

Karthik Abinav Sankararaman, Soham De, Zheng Xu, W Ronny Huang, and Tom Goldstein. The impact of neural network overparameterization on gradient confusion and stochastic gradient descent. In *ICML*, 2020. (pages 1, 5).

Nicol N Schraudolph. Fast curvature matrix-vector products for second-order gradient descent. In *Neural computation*, 2002. (page 5).

Harshay Shah, Kaustav Tamuly, Aditi Raghunathan, Prateek Jain, and Praneeth Netrapalli. The pitfalls of simplicity bias in neural networks. In *NeurIPS*, 2020. (page 1).

Soroosh Shahtalebi, Jean-Christophe Gagnon-Audet, Touraj Laleh, Mojtaba Faramarzi, Kartik Ahuja, and Irina Rish. Sand-mask: An enhanced gradient masking strategy for the discovery of invariances in domain generalization. In *ICML UDL Workshop*, 2021. (pages 3, 6, 20, 21, 25).

Yuge Shi, Jeffrey Seely, Philip HS Torr, N Siddharth, Awni Hannun, Nicolas Usunier, and Gabriel Synnaeve. Gradient matching for domain generalization. *arXiv preprint*, 2021. (pages 2, 3, 17, 20, 21, 24, 25).

Sidak Pal Singh and Dan Alistarh. Woodfisher: Efficient second-order approximation for neural network compression. In *NeurIPS*, 2020. (pages 5, 19).

Baochen Sun and Kate Saenko. Deep coral: Correlation alignment for deep domain adaptation. In *ECCV*, 2016. (pages 1, 3, 21).

Baochen Sun, Jiashi Feng, and Kate Saenko. Return of frustratingly easy domain adaptation. In *AAAI*, 2016. (pages 1, 3, 4).

Josh Tenenbaum. Building machines that learn and think like people. In *AAMAS*, 2018. (page 1).

Damien Teney, Ehsan Abbasnejad, and Anton van den Hengel. Unshuffling data for improved generalization. *arXiv preprint*, 2020. (page 3).

L. Theis, I. Korshunova, A. Tejani, and F. Huszár. Faster gaze prediction with dense networks and fisher pruning. *arXiv preprint*, 2018. (page 4).

Valentin Thomas, Fabian Pedregosa, Bart van Merriënboer, Pierre-Antoine Manzagol, Yoshua Bengio, and Nicolas Le Roux. On the interplay between noise and curvature and its effect on optimization and generalization. In *AISTATS*, 2020. (pages 5, 6, 19).

Ryan Turner, David Eriksson, Michael McCourt, Juha Kiili, Eero Laaksonen, Zhen Xu, and Isabelle Guyon. Bayesian optimization is superior to random search for machine learning hyperparameter tuning: Analysis of the black-box optimization challenge 2020. In *NeurIPS*, 2021. (page 25).

Eric Tzeng, Judy Hoffman, Ning Zhang, Kate Saenko, and Trevor Darrell. Deep domain confusion: Maximizing for domain invariance. In *CoRR*, 2014. (page 3).

Guillermo Valle-Perez, Chico Q. Camargo, and Ard A. Louis. Deep learning generalizes because the parameter-function map is biased towards simple functions. In *ICLR*, 2019. (page 1).

Vladimir N Vapnik. An overview of statistical learning theory. In *TNN*, 1999. (pages 3, 21).

Hemanth Venkateswara, Jose Eusebio, Shayok Chakraborty, and Sethuraman Panchanathan. Deep hashing network for unsupervised domain adaptation. In *CVPR*, 2017. (pages 7, 22).

Yoav Wald, Amir Feder, Daniel Greenfeld, and Uri Shalit. On calibration and out-of-domain generalization. *arXiv preprint*, 2021. (page 9).

Yufei Wang, Haoliang Li, and Alex C Kot. Heterogeneous domain generalization via domain mixup. In *ICASSP*, 2020. (page 3).

Yuan Wu, Diana Inkpen, and Ahmed El-Roby. Dual mixup regularized learning for adversarial domain adaptation. In *ECCV*, 2020. (page 3).

Sang Michael Xie, Ananya Kumar, Robbie Jones, Fereshte Khani, Tengyu Ma, and Percy Liang. In-n-out: Pre-training and self-training using auxiliary information for out-of-distribution robustness. In *ICLR*, 2021. (page 3).

Shen Yan, Huan Song, Nanxiang Li, Lincan Zou, and Liu Ren. Improve unsupervised domain adaptation with mixup training. *arXiv preprint*, 2020. (page 21).

Greg Yang and Hadi Salman. A fine-grained spectral perspective on neural networks. *arXiv preprint*, 2019. (page 17).

Nanyang Ye, Kaican Li, Lanqing Hong, Haoyue Bai, Yiting Chen, Fengwei Zhou, and Zhenguo Li. Ood-bench: Benchmarking and understanding out-of-distribution generalization datasets and algorithms. *arXiv preprint*, 2021. (pages 1, 2, 3, 7).

Dong Yin, Ashwin Pananjady, Max Lam, Dimitris Papailiopoulos, Kannan Ramchandran, and Peter Bartlett. Gradient diversity: a key ingredient for scalable distributed learning. In *AISTATS*, 2018. (pages 1, 5).

Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. Gradient surgery for multi-task learning. In *NeurIPS*, 2020. (page 3).

Marvin Zhang, Henrik Marklund, Nikita Dhawan, Abhishek Gupta, Sergey Levine, and Chelsea Finn. Adaptive risk minimization: A meta-learning approach for tackling group distribution shift. *arXiv preprint*, 2020. (pages 3, 21).

Xingxuan Zhang, Peng Cui, Renzhe Xu, Linjun Zhou, Yue He, and Zheyan Shen. Deep stable learning for out-of-distribution generalization. In *CVPR*, 2021. (pages 3, 8, 23).

Yunbo Zhang, Wenhao Yu, and Greg Turk. Learning novel policies for tasks. In *ICML*, 2019. (page 3).

Han Zhao, Remi Tachet Des Combes, Kun Zhang, and Geoffrey Gordon. On learning invariant representations for domain adaptation. In *ICML*, 2019. (pages 3, 5).

Appendices

These Appendices follow a similar order as their related sections in the main paper.

1. We first detail some theoretical points. Appendix A.1 shows the Taylor expansion of the inconsistency score while Appendix A.2 motivates this project with intuitions from the Neural Tangent Kernel theory. Appendix A.3 proves the effectiveness of our approach in the linear setting.
2. Appendix B enriches the Colored MNIST experiment in the IRM setup. In detail, we first describe the experimental setup in Appendix B.1. We then validate in Appendix B.2 some insights provided in the main paper; in particular, Appendix B.2.4 motivates the diagonal approximation of the gradient covariance.
3. Appendix C enriches the DomainBed experiments. After a description of the benchmark protocols in Appendix C.1, Appendix C.2 provides additional experiments to analyze key components of our approach. Specifically, C.2.1 analyzes the exponential moving average; C.2.2 compares gradient mean matching versus gradient variance matching and also motivates ignoring the gradients in the features extractor; C.2.3 discusses the methodology to select hyperparameter distributions. Then, Appendix C.3 provides the per-dataset results.

A Additional theoretical analysis

A.1 Taylor expansion of the inconsistency score

We further detail the Taylor expansion of the inconsistency score (Parascandolo et al., 2021) used in Section 3.3.2. In the simplified setting $\mathcal{E} = \{A, B\}$, the inconsistency score goes back to:

$$\mathcal{I}^{\epsilon}\left(\theta^*\right) = \max\left\{ \max_{|\mathcal{R}_A(\theta) - \mathcal{R}_A(\theta^*)| \leqslant \epsilon} \left|\mathcal{R}_B(\theta) - \mathcal{R}_A(\theta^*)\right|, \max_{|\mathcal{R}_B(\theta) - \mathcal{R}_B(\theta^*)| \leqslant \epsilon} \left|\mathcal{R}_A(\theta) - \mathcal{R}_B(\theta^*)\right| \right\}.$$

The second-order Taylor expansion of $\mathcal{R}_e$ around $\theta^* = 0$ (with a change of variable) gives:

$$\mathcal{R}_e(\theta) = \mathcal{R}_e(\theta^*) + \theta^\top \nabla_\theta \mathcal{R}_e\left(\theta^*\right) + \frac{1}{2}\theta^\top H_e \theta + \mathcal{O}(\|\theta\|_2^2),$$

for $e \in \{A, B\}$. We assume simultaneous convergence, *i.e.*, θ^* is a local minima across all training domains: $\nabla_\theta \mathcal{R}_A(\theta^*) = \nabla_\theta \mathcal{R}_B(\theta^*) = 0$. Thus:

$$\begin{aligned} \max_{|\mathcal{R}_A(\theta) - \mathcal{R}_A(\theta^*)| \leqslant \epsilon} \left|\mathcal{R}_B(\theta) - \mathcal{R}_A(\theta^*)\right| &\approx \max_{\left|\frac{1}{2}\theta^\top H_A \theta\right| \leqslant \epsilon} \left|\mathcal{R}_B(\theta^*) + \frac{1}{2}\theta^\top H_B \theta - \mathcal{R}_A(\theta^*)\right| \\ &\lessapprox \left|\mathcal{R}_B(\theta^*) - \mathcal{R}_A(\theta^*)\right| + \max_{\left|\frac{1}{2}\theta^\top H_A \theta\right| \leqslant \epsilon} \left|\frac{1}{2}\theta^\top H_B \theta\right|, \end{aligned} \tag{5}$$

where the last inequality is deduced from the triangle inequality.

The first term $|\mathcal{R}_B(\theta^*) - \mathcal{R}_A(\theta^*)|$ was simply assumed small at convergence in AND-mask (Parascandolo et al., 2021). We further justify this approximation for Fishr by reminding that the empirical risks difference across domains is the V-REx (Krueger et al., 2021) criterion: thus, as argued in Section 3.2 and as shown in Appendix B.2.1, Fishr forces this first term to be low at convergence.

Following Parascandolo et al. (2021), the second term is more easily understood when Hessians are diagonal: $\boldsymbol{H}_e = \operatorname{diag}\left(\lambda_1^e, \cdots, \lambda_n^e\right)$ with $\forall i \in \{1, \ldots, |\theta|\}, \lambda_i^e > 0$. In this case, $\max_{\left|\frac{1}{2}\theta^\top H_A \theta\right| \leqslant \epsilon} \left|\frac{1}{2}\theta^\top H_B \theta\right| = \max_{\|\tilde{\theta}\|^2 \leqslant \epsilon} \sum_i \tilde{\theta}_i^2 \lambda_i^B / \lambda_i^A = \epsilon \cdot \max_i \lambda_i^B / \lambda_i^A$ is large when $\exists i$ such that λ_i^A is small but λ_i^B is large. Thus, $\mathcal{I}^{\epsilon}\left(\theta^*\right) \lessapprox \epsilon \cdot \max \max_i \left\{\lambda_i^B / \lambda_i^A, \lambda_i^A / \lambda_i^B\right\}$ decreases when $\boldsymbol{H}_A$ and $\boldsymbol{H}_B$ have similar eigenvalues.

In conclusion, Fishr reduces inconsistency by matching (1) domain-level empirical risks and (2) domain-level Hessians across the training domains.

A.2 NEURAL TANGENT KERNEL PERSPECTIVE

In this section we motivate the matching of gradient covariances with new arguments from the Neural Tangent Kernel (NTK) (Jacot et al., 2018) theory. As a reminder, the NTK $\boldsymbol{K} \in \mathbb{R}^{n \times n}$ is the gramian matrix with entries $\boldsymbol{K}[i,j] = \nabla_\theta f_\theta(x^i)^\top \cdot \nabla_\theta f_\theta(x^j)$ that measure the gradients similarity at two different input points x^i and x^j. This kernel dictates the training dynamics of the DNN and remains fixed in the infinite width limit. Most importantly, as stated in Yang & Salman (2019), "the simplicity bias of a wide neural network can be read off quickly from the spectrum of $\boldsymbol{K}$: if the largest eigenvalue $[\lambda^{\max}]$ of $\boldsymbol{K}$ accounts for most of $\mathrm{Tr}(\boldsymbol{K})$, then a typical random network looks like a function from the top eigenspace of $\boldsymbol{K}$": this holds for ReLu networks. In summary, gradient descent mostly happens in a tiny subspace (Gur-Ari et al., 2018) whose directions are defined by the main eigenvectors from $\boldsymbol{K}$. Moreover, the learning speed is dictated by $\lambda^{\max}$, which can be used to estimate a condition for a learning rate η to converge: $\eta < 2/\lambda^{\max}$ (Karakida et al., 2019).

In a multi-domain framework, having similar spectral decompositions across $\{\boldsymbol{K}_e\}_{e\in\mathcal{E}}$ during the optimization process would improve OOD generalization for two reasons:

1. Having similar top eigenvectors across $\{\boldsymbol{K}_e\}_{e\in\mathcal{E}}$ would delete detrimental domain-dependent shortcuts and favor the learning of a common mechanism. Indeed, truly informative features should remain consistent across domains.
2. Having similar top eigenvalues across $\{\boldsymbol{K}_e\}_{e\in\mathcal{E}}$ would improve the optimization schema for simultaneous training at the same speed. Indeed, it would facilitate the finding of a learning rate for simultaneous convergence on all domains. It's worth noting that if we quickly overfit on a first domain using spurious explanations, invariances will then be hard to learn due to the gradient starvation phenomena (Pezeshki et al., 2020).

Directly matching $\boldsymbol{K}_e$ would require assuming that each domain coincides and contains the same samples; for example, with different pose angles (Ghifary et al., 2015). To avoid such a strong assumption, we leverage the fact that the 'true' Fisher Information Matrix $\boldsymbol{F}$ and the NTK $\boldsymbol{K}$ share the same non-zero eigenvalues since $\boldsymbol{F}$ is dual to $\boldsymbol{K}$ (see Appendix C.1 in Maddox et al. (2019), notably for classification tasks). Moreover, their eigenvectors are strongly related (see Appendix C in Kopitkov & Indelman (2019)). Thus, having similar $\{\boldsymbol{F}_e\}_{e\in\mathcal{E}}$ encourages $\{\boldsymbol{K}_e\}_{e\in\mathcal{E}}$ to have similar spectral decomposition. Based on the close relations between $\boldsymbol{C}$ and $\boldsymbol{F}$ (see Section 3.3.1), this further motivates the need to match gradient variances during the SGD trajectory — and not only at convergence as in Section 3.3.2.

A.3 ANALYSIS IN THE LINEAR SETTING

A.3.1 EMPIRICAL PROOF ON A LINEAR EXAMPLE

First, we experimentally prove that Fishr is effective in the linear setting. To do so, we consider the binary classification dataset introduced in Section 3.2 from Fish (Shi et al., 2021). Each example is composed of 4 features (f_1, f_2, f_3, f_4). While f_1 is invariant across the two train domains and the test domain, the three other features are spurious: their correlations with the label vary in each domain. The model is a linear logistic regression, with trainable weights W and bias b. As f_2 and f_3 have higher correlations with the label than f_1 in training, ERM relies mostly on f_2 and f_3. This is indicated in the first line of Table 5 by the large values (3.3) for weights associated to f_2 and f_3; this induces low test accuracy (57%). On the contrary, Fishr forces the linear model to rely mostly on the invariant feature f_1, as indicated by the lower values (1.2) for weights associated to f_2 and f_3; in accuracy, Fishr performs similarly in test and train (93%).

Method	Matched statistics	Train acc.	Test acc.	W	b
ERM	N/A	97 %	57 %	[2.8,3.3,3.3,0.0]	-2.7
Fish	Gradient means	93 %	93 %	[0.4,0.2,0.2,0.0]	-0.4
Fishr	Centered gradient variances	93 %	93 %	[2.0,1.2,1.2,0.0]	-0.6
Fishr	Uncentered gradient variances	93 %	93 %	[1.9,0.9,0.9,0.0]	-0.6

Table 5: **Performances comparison on the linear dataset** from (Shi et al., 2021)

A.3.2 THEORETICAL ANALYSIS IN THE LINEAR CLASSIFIER

Second, we delve into the theoretical analysis of the Fishr regularization in a linear binary classifier, that leverages p features: these features are either fixed through learning (the linear setting), or predicted from a trainable features extractor ϕ. We note $z_e^i \in \mathbb{R}^p$ the features for the i-th example from the domain e, $\hat{y}_e^i \in [0, 1]$ the predictions after sigmoid and $y_e^i \in \{0, 1\}$ the one-hot encoded target. The linear layer W is parametrized by weights $\{w_k\}_{k=1}^p$ and bias b.

The gradient of the loss for this sample with respect to the **bias** b is $\nabla_b \ell(y_e^i, \hat{y}_e^i) = (\hat{y}_e^i - y_e^i)$. Thus, the uncentered gradient variance in b for domain e is: $\boldsymbol{v}_e^b = \frac{1}{n_e} \sum_{i=1}^{n_e} (\hat{y}_e^i - y_e^i)^2 = \text{MSE}_e$, which is exactly the mean squared error (MSE) between predictions and targets in domain e. Thus, matching gradient variances in b will match the risks across domains. We recover the objective from V-REx (Krueger et al., 2021), with a different loss (squared error instead of negative log likelihood). This remains true both when features are fixed through learning and when they are trainable.

We can also look at the gradients with respect to the **weight** w_k: $\nabla_{w_k} \ell(y_e^i, \hat{y}_e^i) = (\hat{y}_e^i - y_e^i) z_e^i[k]$. Thus, the uncentered gradient variance in w_k for domain e is: $\boldsymbol{v}_e^{w_k} = \frac{1}{n_e} \sum_{i=1}^{n_e} \left((\hat{y}_e^i - y_e^i) z_e^i[k] \right)^2$. This is the squared error, weighted for each sample (z_e^i, y_e^i) by the square of the k-th feature $z_e^i[k]^2$: matching v^{w_k} will match this weighted squared error, with k different weighting schemes. In the linear setting, these weightings are constant through learning and depend on the static features distribution: Fishr matches weighted risks across domains. An interesting example is when features are binary ($z_e^i \in \{0, 1\}$); then, Fishr applies domain-level risks matching on groups of samples having a shared feature. In contrast when features are trainable, ϕ will also adapt to match those terms.

More exactly in Fishr, we match centered gradient variances, whose formula in b for domain e is:

$$\boldsymbol{v}^b = \frac{1}{n_e - 1} \sum_{i=1}^{n_e} (\hat{y}_e^i - y_e^i)^2 - \frac{n_e}{n_e - 1} \overline{\nabla_b \ell(y_e, \hat{y}_e)}^2 = \frac{n_e}{n_e - 1} \left(\text{MSE}_e - \overline{\nabla_b \ell(y_e, \hat{y}_e)}^2 \right), \quad (6)$$

where $\overline{\nabla_b \ell(y_e, \hat{y}_e)}$ is the mean gradient in b for domain e. Under the mild assumption $\overline{\nabla_b \ell(y_e, \hat{y}_e)} \approx 0$ at convergence, we recover the uncentered variance gradient matching. Similar assumptions can be made for the other weights w_k. This further explains why centering or not the variances perform similarly in Table 5 and in Appendix B.2.5. These gradient means can also be explicitly matched for improved performances, as shown in Appendix C.2.2.

B COLORED MNIST IN THE IRM SETUP

B.1 DESCRIPTION OF THE COLORED MNIST EXPERIMENT

Colored MNIST is a binary digit classification dataset introduced in IRM (Arjovsky et al., 2019). Compared to the traditional MNIST (LeCun et al., 2010), it has 2 main differences. *First*, 0-4 and 5-9 digits are each collapsed into a single class, with a 25% chance of label flipping. *Second*, digits are either colored red or green, with a strong correlation between label and color in training. However, this correlation is reversed at test time. Specifically, in training, the model has access to two domains $\mathcal{E} = \{90\%, 80\%\}$: in the first domain, green digits have a 90% chance of being in 5-9; in the second, this chance goes down to 80%. In test, green digits have a 10% chance of being in 5-9. Due to this modification in correlation, a model should ideally ignore the color information and only rely on the digits' shape: this would obtain a 75% test accuracy.

In the experimental setup from IRM, the network is a 3 layers MLP with ReLu activation, optimized with Adam (Kingma & Ba, 2014). IRM selected the following hyperparameters by random search over 50 trials: hidden dimension of 390, l_2 regularizer weight of 0.00110794568, learning rate of 0.0004898536566546834, penalty anneal iters (or warmup iter) of 190, penalty weight (λ) of 91257.18613115903, 501 epochs and batch size 25,000 (half of the dataset size). We strictly keep the same hyperparameters values in our proof of concept in Section 4.1. The code in the anonymous repository `https://anonymous.4open.science/r/fishr-anonymous-EBB6/coloredmnist/train_coloredmnist.py` is almost unchanged from `https://github.com/facebookresearch/InvariantRiskMinimization`.

B.2 EMPIRICAL VALIDATION OF SOME KEY INSIGHTS

B.2.1 EMPIRICAL TRAINING RISK MATCHING

We argue in Section 3.2 that gradient amplitudes are directly related to the loss values. Indeed, the constant multiplier rule states that multiplying the loss by a constant will also multiply the gradients by the same constant. Thus, forcing gradients to be similar should bring the domain-level empirical training risks closer. This was proved in Appendix A.3. Fig. 4 empirically verifies this insight on Colored MNIST: $|\mathcal{R}_{90\%} - \mathcal{R}_{80\%}| \rightarrow 0$ after epoch 190 for a network Fishr$_\omega$-trained, even though predicting accurately on the domain 90% is easier than on the domain 80%.

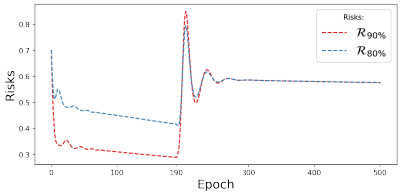

Figure 4: **Risks dynamics on Colored MNIST with Fishr$_\omega$**. At epoch 190, λ steps us and then empirical risks $\mathcal{R}_{90\%}$ and $\mathcal{R}_{80\%}$ get closer.

B.2.2 HESSIAN MATCHING

Based on empirical works (Li et al., 2020; Singh & Alistarh, 2020; Thomas et al., 2020), we argue in Section 3.3.1 that gradient covariance $\boldsymbol{C}$ can be used as a proxy to regularize the Hessian $\boldsymbol{H}$ — even though the proper approximation bounds are out of scope of this paper. This was empirically validated at convergence in Table 1 and during training in Fig. 3. We leveraged the *DiagHessian* method from BackPACK to compute Hessian diagonals, in classification weights ω — because of memory issues when computing Hessians for all weights θ. As a side note, Hessians remain impractical in a training objective as computing "Hessian is an order of magnitude more computationally intensive" than individual gradients (see Fig. 9 in Dangel et al. (2020)).

This appendix further analyzes the Hessian during training. Fig. 5 illustrates the dynamics for Fishr$_\omega$: following the scheduling previously described in Appendix B.1, λ jumping to a high value at epoch 190 activates the regularization. After this epoch, the domain-level Hessians are not only close in Frobenius distance, but also have similar norms and directions. On the contrary, when using only ERM in Fig. 6, the distance between domain-level Hessians keeps increasing with the number of epochs. As a side note, flatter loss landscapes in ERM — as reflected by the Hessian norms in orange — do not correlate with improved generalization (Dinh et al., 2017).

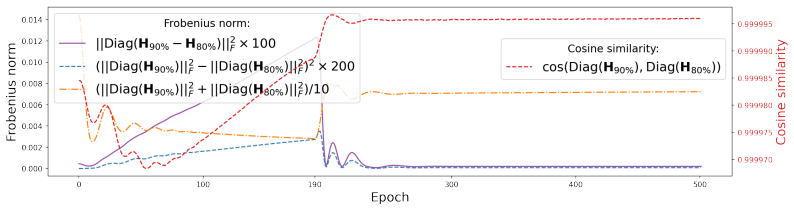

Figure 5: **Dynamics on Colored MNIST with Fishr$_\omega$ strategy**: at epoch 190, λ steps up. Then domain-level Hessians are matched across domains (purple). More precisely, they take similar directions — high cosine similarity (red) — and similar norms (blue). The Hessians' norms (orange) remain quite high thus the domain-level loss landscapes are rather sharp.

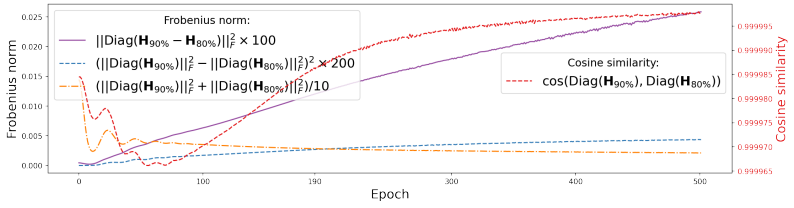

Figure 6: **Dynamics on Colored MNIST with ERM strategy**: $\lambda = 0$ along training. The Frobenius distance between domain-level Hessians (purple) keeps increasing: so does the distance between their norms (blue). Their cosine similarity (red) steadily increases, but slower than with Fishr. The domain-level loss landscapes are flat at convergence (low Hessian norms in orange).

B.2.3 COLORED MNIST WITHOUT LABEL FLIPPING

To further validate that Fishr can tackle distribution shifts, we investigate Colored MNIST but without the 25% label flipping. In Table 6, the label is then fully predictable from the digit shape. Using hyperparameters defined previously in Appendix B.1, IRM (82.2%) performs worse than ERM (91.8%) while V-REx and Fishr$_\omega$ perform better (95.3%): Fishr works even without label noise.

Table 6: **Colored MNIST experiments without label flipping**.

Method	Train acc.	Test acc.	Gray test acc.
ERM	99.0 ± 0.0	91.8 ± 0.2	95.0 ± 0.4
IRM	96.4 ± 0.2	82.2 ± 0.1	92.6 ± 0.2
V-REx	97.1 ± 0.2	95.3 ± 0.4	94.1 ± 0.4
Fishr$_\theta$	97.9 ± 0.2	93.6 ± 0.4	94.8 ± 0.4
Fishr$_\omega$	97.0 ± 0.2	95.3 ± 0.4	94.1 ± 0.4
Fishr$_\phi$	97.9 ± 0.1	93.5 ± 0.3	94.8 ± 0.4

B.2.4 GRADIENT VARIANCE OR COVARIANCE ?

We have justified ignoring the off-diagonal parts of the covariance to reduce the memory overhead. For the sake of completeness, the second line in Table 7 shows results with the full covariance matrix. This experiment is possible only when considering gradient in the classifier w_ω for memory reasons. Overall, results are similar (or slightly worse) as when using only the diagonal: the slight difference may be explained by the approaches' different suitability to the hyperparameters (that were optimized for IRM). In conclusion, this preliminary experiment suggests that targeting the diagonal components is the most critical. We hope future works will further investigate this diagonal approximation or provide new methods to reduce the computational costs, such as K-FAC approximations (Heskes, 2000; Martens & Grosse, 2015).

Table 7: **Colored MNIST experiments** with different statistics matched. All methods use hyperparameters optimized for IRM.

Method			25% label flipping			No label flipping		
Gradients in	Name	Matched statistics	Train acc.	Test acc.	Gray test acc.	Train acc.	Test acc.	Gray test acc.
ω	Centered variance (= Fishr$_\omega$)	$\text{Diag}(\boldsymbol{C}_e)$	71.0 ± 0.9	69.5 ± 1.0	70.2 ± 1.1	97.0 ± 0.2	95.3 ± 0.4	94.1 ± 0.4
	Centered covariance	$\boldsymbol{C}_e$	70.7 ± 1.0	69.1 ± 1.1	69.9 ± 1.1	97.0 ± 0.2	95.3 ± 0.4	94.0 ± 0.4
	Uncentered variance	$\text{Diag}(\frac{1}{n_e}\tilde{\boldsymbol{F}}_e)$	71.3 ± 0.9	69.5 ± 1.0	70.3 ± 1.0	97.0 ± 0.2	95.3 ± 0.4	94.1 ± 0.4
θ	Centered variance (= Fishr$_\theta$)	$\text{Diag}(\boldsymbol{C}_e)$	69.6 ± 0.9	71.2 ± 1.1	70.2 ± 0.7	97.9 ± 0.1	93.5 ± 0.3	94.7 ± 0.4
	Centered covariance	$\boldsymbol{C}_e$	Not	possible	for	computational	(memory)	reasons
	Uncentered variance	$\text{Diag}(\frac{1}{n_e}\tilde{\boldsymbol{F}}_e)$	71.0 ± 0.8	70.0 ± 1.1	70.1 ± 0.9	97.9 ± 0.0	93.5 ± 0.3	94.8 ± 0.4
ϕ	Centered variance (= Fishr$_\phi$)	$\text{Diag}(\boldsymbol{C}_e)$	65.6 ± 1.3	73.8 ± 1.0	70.0 ± 0.9	97.9 ± 0.1	93.5 ± 0.3	94.8 ± 0.4
	Centered covariance	$\boldsymbol{C}_e$	Not	possible	for	computational	(memory)	reasons
	Uncentered variance	$\text{Diag}(\frac{1}{n_e}\tilde{\boldsymbol{F}}_e)$	71.5 ± 0.8	69.1 ± 1.1	70.0 ± 1.0	97.9 ± 0.1	93.5 ± 0.3	94.8 ± 0.4

B.2.5 CENTERED OR UNCENTERED VARIANCE ?

In Section 3.3.1, we argue that the gradient centered covariance $\boldsymbol{C}$ and the empirical Fisher Information Matrix (or uncentered covariance) $\tilde{\boldsymbol{F}}$ are highly related and equivalent when the DNN is at convergence and the gradient means are zero. So, we could have tackled the diagonals of the domain-level $\{\tilde{\boldsymbol{F}}_e\}_{e\in\mathcal{E}}$ across domains, *i.e.*, without centering the variances. Empirically, comparing the first and third lines in Table 7 shows that centering or not the variance are almost equivalent. This holds true when applying Fishr on all weights θ (as lines fourth and six are also very similar).

C DOMAINBED

C.1 DESCRIPTION OF THE DOMAINBED BENCHMARK

We now further detail our experiments on the DomainBed benchmark. Scores from most baselines are taken from the DomainBed (Gulrajani & Lopez-Paz, 2021) paper. Scores for AND-mask and SAND-mask are taken from the SAND-mask paper (Shahtalebi et al., 2021). For Fish (Shi et al., 2021), averaged 'Training' scores are taken from the arXiv paper and averaged 'Oracle' scores are from direct messages with the authors: however, the per-dataset results are not available. Scores for IGA (Koyama & Yamaguchi, 2020) are not yet available: yet, for the sake of completeness, we analyze IGA in Appendix C.2.2. Missing scores will be included when available.

The same procedure was applied for all methods: for each domain, a random hyperparameter search of 20 trials over a joint distribution, described in Table 8, is performed. We discuss the choice of these distributions in Appendix C.2.3. The learning rate, the batch size (except for ARM), the weight decay and the dropout distributions are shared across all methods - all trained with Adam

Table 8: **Hyperparameters**, their default values and distributions for random search.

Condition	Parameter	Default value	Random distribution
VLCS / PACS / OfficeHome / TerraIncognita / DomainNet	learning rate	0.00005	$10^{\text{Uniform}(-5,-3.5)}$
	batch size	32	$2^{\text{Uniform}(3,5.5)}$ if not DomainNet else $2^{\text{Uniform}(3,5)}$
	weight decay	0	$10^{\text{Uniform}(-6,-2)}$
	dropout	0	RandomChoice $([0, 0.1, 0.5])$
Rotated MNIST / Colored MNIST	learning rate	0.001	$10^{\text{Uniform}(-4.5,-3.5)}$
	batch size	64	$2^{\text{Uniform}(3,9)}$
	weight decay	0	0
All	steps	5000	5000
Fishr	regularization strength λ	1000	$10^{\text{Uniform}(1,4)}$
	ema γ	0.95	Uniform$(0.9, 0.99)$
	warmup iterations	1500	Uniform$(0, 5000)$

(Kingma & Ba, 2014). Specific hyperparameter distributions for concurrent methods can be found in the original work of Gulrajani & Lopez-Paz (2021). The data from each domain is split into 80% (used as training and testing) and 20% (used as validation for hyperparameter selection) splits. This random process is repeated with 3 different seeds: the reported numbers are the means and the standard errors over these 3 seeds. We focus on the two 'Oracle' and 'Training' model selection methods and have not run the 'Leave-one-domain-out Cross-validation' for computational reasons.

We clarify a subtle point (omitted in the Algorithm 1) concerning the hyperparameter γ that controls: $\bar{\boldsymbol{v}}_e^t = \gamma \bar{\boldsymbol{v}}_e^{t-1} + (1-\gamma)\boldsymbol{v}_e^t$ at step t. We remind that $\bar{\boldsymbol{v}}_e^{t-1}$ from previous step $t-1$ is 'detached' from the computational graph. Thus when $\mathcal{L}$ from Eq. 4 is differentiated during SGD, the gradients going through $\boldsymbol{v}_e^t$ are multiplied by $(1-\gamma)$. To compensate this and decorrelate the impact of γ and of λ (that controls the regularization strength), we match $\frac{1}{1-\gamma}\bar{\boldsymbol{v}}_e^t$. Finally, with this $(1-\gamma)$ **correction**, the gradients' strength backpropagated in the network is independent of γ.

Here we list all **concurrent approaches**.

- ERM: Empirical Risk Minimization (Vapnik, 1999)
- IRM: Invariant Risk Minimization (Arjovsky et al., 2019)
- GroupDRO: Group Distributionally Robust Optimization (Sagawa et al., 2020a)
- Mixup: Interdomain Mixup (Yan et al., 2020)
- MLDG: Meta Learning Domain Generalization (Li et al., 2018a)
- CORAL: Deep CORAL (Sun & Saenko, 2016)
- MMD: Maximum Mean Discrepancy (Li et al., 2018b)
- DANN: Domain Adversarial Neural Network (Ganin et al., 2016)
- CDANN: Conditional Domain Adversarial Neural Network (Li et al., 2018c)
- MTL: Marginal Transfer Learning (Blanchard et al., 2021)
- SagNet: Style Agnostic Networks (Nam et al., 2021)
- ARM: Adaptive Risk Minimization (Zhang et al., 2020)
- V-REx: Variance Risk Extrapolation (Krueger et al., 2021)
- RSC: Representation Self-Challenging (Huang et al., 2020)
- AND-mask: Learning Explanations that are Hard to Vary (Parascandolo et al., 2021)
- SAND-mask: An Enhanced Gradient Masking Strategy for the Discovery of Invariances in Domain Generalization (Shahtalebi et al., 2021)
- IGA: Out-of-distribution generalization with maximal invariant predictor (Koyama & Yamaguchi, 2020)
- Fish: Gradient Matching for Domain Generalization (Shi et al., 2021)

DomainBed includes seven multi-domain computer vision classification **datasets**:

1. Colored MNIST (Arjovsky et al., 2019) is a variant of the MNIST handwritten digit classification dataset (LeCun et al., 2010). As described previously in Appendix B.1, domain $d \in \{90\%, 80\%, 10\%\}$ contains a disjoint set of digits colored: the correlation strengths between color and label vary across domains. The dataset contains 70,000 examples of dimension $(2, 28, 28)$ and 2 classes. Most importantly, the network, the hyperparameters, the image shapes, etc. are **not** the same as in the IRM setup from Section 4.1.
2. Rotated MNIST (Ghifary et al., 2015) is a variant of MNIST where domain $d \in \{0, 15, 30, 45, 60, 75\}$ contains digits rotated by d degrees, with 70,000 examples of dimension $(1, 28, 28)$ and 10 classes.
3. VLCS (Fang et al., 2013) includes photographic domains $d \in \{$Caltech101, LabelMe, SUN09, VOC2007$\}$, with 10,729 examples of dimension $(3, 224, 224)$ and 5 classes.
4. PACS (Li et al., 2017) includes domains $d \in \{$art, cartoons, photos, sketches$\}$, with 9,991 examples of dimension $(3, 224, 224)$ and 7 classes.
5. OfficeHome (Venkateswara et al., 2017) includes domains $d \in \{$art, clipart, product, real$\}$, with 15,588 examples of dimension $(3, 224, 224)$ and 65 classes.
6. TerraIncognita (Beery et al., 2018) contains photographs of wild animals taken by camera traps at locations $d \in \{$L100, L38, L43, L46$\}$, with 24,788 examples of dimension $(3, 224, 224)$ and 10 classes.
7. DomainNet (Peng et al., 2019) has six domains $d \in \{$clipart, infograph, painting, quickdraw, real, sketch$\}$, with 586,575 examples of size $(3, 224, 224)$ and 345 classes.

Neural network **architectures** used for each dataset are shown in Table 9a. Table 9b describes the convolutional neural network architecture used for MNIST experiments: note that this is not the same MLP (described in Appendix B.1) as in our proof of concept in Section 4.1. The 'ResNet-50' network is pretrained on ImageNet, has a dropout layer before the newly added dense layer and is fine-tuned on the new datasets with frozen batch normalization layers.

Table 9: Summary of the architectures used in DomainBed.

(a) Neural network architectures used for each dataset.

Dataset	**Architecture**
Colored MNIST / Rotated MNIST	MNIST ConvNet
VLCS / PACS / OfficeHome / TerraIncognita / DomainNet	ResNet-50

(b) MNIST ConvNet architecture.

#	**Layer**
1	Conv2D (in $= d$, out $= 64$)
2	ReLU
3	GroupNorm (groups $= 8$)
4	Conv2D (in $= 64$, out $= 128$, stride $= 2$)
5	ReLU
6	GroupNorm (8 groups)
7	Conv2D (in=128, out=128)
8	ReLU
9	GroupNorm (8 groups)
10	Conv2D (in=128, out=128)
11	ReLU
12	GroupNorm (8 groups)
13	Global average-pooling

C.2 FISHR COMPONENT ANALYSIS ON DOMAINBED

C.2.1 FOCUS ON THE EXPONENTIAL MOVING AVERAGE

Following Le Roux et al. (2011), we use an exponential moving average (ema) parameterized by γ for computing gradient variances in DomainBed: the closer γ is to 1, the longer a batch will impact the variance from later steps. We now further analyze the impact of this strategy, which is

not specific to Fishr and was used previously in other works (Nam et al., 2020; Blanchard et al., 2021; Zhang et al., 2021) for OOD generalization. Notably, this ema strategy could be applied to better estimate domain-level empirical risks in V-REx (Krueger et al., 2021). For a fair comparison, we introduce a new approach — V-REx with ema — that penalizes $|\bar{\mathcal{R}}_A^t - \bar{\mathcal{R}}_B^t|^2$ at step t where $\bar{\mathcal{R}}_e^t = \gamma \bar{\mathcal{R}}_e^{t-1} + (1-\gamma)\mathcal{R}_e^t$ when $\mathcal{E} = \{A, B\}$.

Thus, we compare V-REx and Fishr, with $\gamma = 0$ (✗) or with $\gamma \sim \text{Uniform}(0.9, 0.99)$ (✓, as described in Table 8). On the synthetic Colored MNIST in Table 10, the ema is critical for Fishr — notably when training on $\mathcal{E} = \{90\%, 80\%\}$ and the dataset 10% is in test (from ✗34.0% to ✓58.9% in 'Oracle'). V-REx also benefits from ema. On the 'real' dataset OfficeHome in Table 11, the ema is less beneficial (from ✗67.5% to ✓68.2% in 'Oracle' for Fishr). Notably, it worsens V-REX. Overall, Fishr — with and without ema — outperforms V-REx on OfficeHome.

We speculate that ema mainly helps when the batch size is not sufficiently large to detect 'slight' correlation shifts in the training datasets: *e.g.*, when batch size $\sim 2^{\text{Uniform}(3,9)}$ and training datasets $\mathcal{E} = \{90\%, 80\%\}$ in Colored MNIST. We remind that when the batch size was 25,000 in the Colored MNIST setup from IRM, Fishr reached 69.5% (without ema) in Table 2 from Section 4.1. On the contrary, when the shift is more prominent as in OfficeHome, the ema may be less necessary. Most importantly, Fishr — with and without ema — improves over ERM on these datasets.

Table 10: **Importance of the exponential moving average (ema)** on DomainBed's Colored MNIST.

Model selection	Algorithm	ema	+90%	+80%	10%	Avg
Oracle	ERM	N/A	71.8 ± 0.4	72.9 ± 0.1	28.7 ± 0.5	57.8
	V-REx	✗	72.8 ± 0.3	73.0 ± 0.3	55.2 ± 4.0	67.0
		✓	73.0 ± 0.2	73.0 ± 0.3	**59.9** ± 2.6	68.6
	Fishr	✗	72.7 ± 0.3	72.8 ± 0.1	34.0 ± 4.5	59.8
		✓	**74.1** ± 0.6	**73.3** ± 0.1	58.9 ± 3.7	**68.8**
Training	ERM	N/A	71.7 ± 0.1	72.9 ± 0.2	10.0 ± 0.1	51.5
	V-REx	✗	72.4 ± 0.3	72.9 ± 0.4	**10.2** ± 0.0	51.8
		✓	**72.6** ± 0.5	73.3 ± 0.1	9.8 ± 0.1	51.9
	Fishr	✗	71.1 ± 0.6	**73.6** ± 0.1	10.1 ± 0.2	51.6
		✓	72.3 ± 0.9	73.5 ± 0.2	10.1 ± 0.2	**52.0**

Table 11: **Importance of the exponential moving average (ema)** on DomainBed's OfficeHome.

Model selection	Algorithm	ema	A	C	P	R	Avg
Oracle	ERM	N/A	61.7 ± 0.7	53.4 ± 0.3	74.1 ± 0.4	76.2 ± 0.6	66.4
	V-REx	✗	59.6 ± 1.0	53.3 ± 0.3	73.2 ± 0.5	76.6 ± 0.4	65.7
		✓	59.0 ± 0.7	52.8 ± 0.8	74.6 ± 0.4	75.5 ± 0.3	65.5
	Fishr	✗	**63.6** ± 0.4	53.2 ± 0.5	75.4 ± 0.5	77.8 ± 0.3	67.5
		✓	63.4 ± 0.8	**54.2** ± 0.3	**76.4** ± 0.3	**78.5** ± 0.2	**68.2**
Training	ERM	N/A	61.3 ± 0.7	52.4 ± 0.3	75.8 ± 0.1	76.6 ± 0.3	66.5
	V-REx	✗	60.7 ± 0.9	53.0 ± 0.9	75.3 ± 0.1	76.6 ± 0.5	66.4
		✓	59.2 ± 1.0	51.7 ± 0.5	75.2 ± 0.2	76.6 ± 0.3	65.7
	Fishr	✗	62.2 ± 1.0	53.5 ± 0.2	**76.6** ± 0.2	77.8 ± 0.4	67.5
		✓	**62.4** ± 0.5	**54.4** ± 0.4	76.2 ± 0.5	**78.3** ± 0.1	**67.8**

C.2.2 More general component analysis by comparing gradient variance versus gradient mean matching

As a reminder from the Section 2, IGA (Koyama & Yamaguchi, 2020) is an unpublished gradient-based approach that matches gradient means across domains, *i.e.*, minimizes $||\boldsymbol{g}_A - \boldsymbol{g}_B||_2^2$ when $\mathcal{E} = \{A, B\}$ and where $\boldsymbol{g}_e = \frac{1}{n_e} \sum_{i=1}^{n_e} \nabla_\theta \ell\left(f_\theta(\boldsymbol{x}_e), \boldsymbol{y}_e\right)$. Scores for IGA are not available publicly and thus were not included in Section 4.2. Moreover, IGA is very costly and impractical: IGA is approximately $(|\mathcal{E}| + 1)$ times longer to train than ERM. Yet, we ran the DomainBed implementation of IGA on one 'synthetic' and one 'real' dataset. Table 12 shows that the IGA has little effect on Colored MNIST (58.0% vs. 57.8% for ERM in 'Oracle'). Moreover, on OfficeHome in Table

13, IGA hinders learning (56.9% vs. 66.4% for ERM in 'Oracle'). In brief, the seminal "IGA [...] could completely fail when generalizing to unseen domains", as stated in Fish (Shi et al., 2021).

In the rest of this section, we include IGA in Fishr codebase so that both methods leverage the same implementation choices: this enables **fairer comparisons between gradient mean matching and gradient covariance matching**. These experiments provide further insights regarding Fishr main components: specifically, enforcing invariance (1) only in the classifier's weights ω (2) after a warmup period and (3) with an exponential moving average.

First, Fishr only considers gradient covariances in the classifier's weights ω. Similarly, we try to apply IGA's gradient mean matching but only in w_ω rather than in f_θ. This new method works significantly better (67.2% when $\boldsymbol{g}_e = \frac{1}{n_e}\sum_{i=1}^{n_e} \nabla_\omega \ell\left(f_\theta(\boldsymbol{x}_e), \boldsymbol{y}_e\right)$ vs. 56.9% when $\boldsymbol{g}_e = \frac{1}{n_e}\sum_{i=1}^{n_e} \nabla_\theta \ell\left(f_\theta(\boldsymbol{x}_e), \boldsymbol{y}_e\right)$ for 'Oracle' OfficeHome in Table 13) while reducing the computational overhead. This further motivates the **invariance in the classifier rather than in the low-level layers** (which need to adapt to shifts in pixels for instance). We have done this analysis on IGA and not on Fishr because keeping individual gradients from the whole network f_θ in the GPU memory was not possible with ResNet-50 on our hardware.

Table 12: **Fishr (gradient covariance) vs. IGA (gradient mean)** on DomainBed's Colored MNIST.

Model selection	Algorithm	Gradients in	Warmup	ema	+90%	+80%	10%	Avg
Oracle	ERM	N/A	N/A	N/A	71.8 ± 0.4	72.9 ± 0.1	28.7 ± 0.5	57.8
	IGA	$\theta = \omega \oplus \phi$	✗	✗	71.8 ± 0.5	73.0 ± 0.3	29.2 ± 0.5	58.0
		ω	✗	✗	72.4 ± 0.1	**73.3** ± 0.2	29.3 ± 0.6	58.3
		ω	✓	✗	72.5 ± 0.2	**73.3** ± 0.1	31.8 ± 0.7	59.2
		ω	✓	✓	72.6 ± 0.3	72.9 ± 0.2	50.0 ± 1.2	65.2
	Fishr	ω	✗	✗	73.0 ± 0.3	73.2 ± 0.1	29.5 ± 1.1	58.6
			✓	✗	72.7 ± 0.3	72.8 ± 0.1	34.0 ± 4.5	59.8
			✓	✓	**74.1** ± 0.6	**73.3** ± 0.1	58.9 ± 3.7	68.8
	Fishr + IGA	ω	✓	✓	73.3 ± 0.0	72.6 ± 0.5	**66.3** ± 2.9	**70.7**
Training	ERM	N/A	N/A	N/A	71.7 ± 0.1	72.9 ± 0.2	10.0 ± 0.1	51.5
	IGA	$\theta = \omega \oplus \phi$	✗	✗	71.8 ± 0.3	73.2 ± 0.2	9.8 ± 0.0	51.6
		ω	✗	✗	71.8 ± 0.1	73.2 ± 0.2	**10.1** ± 0.0	51.7
		ω	✓	✗	71.8 ± 0.2	73.1 ± 0.2	**10.1** ± 0.0	51.7
		ω	✓	✓	**72.5** ± 0.4	73.3 ± 0.2	**10.1** ± 0.1	**52.0**
	Fishr	ω	✗	✗	71.6 ± 0.1	73.2 ± 0.1	9.9 ± 0.0	51.6
			✓	✗	71.1 ± 0.6	**73.6** ± 0.1	**10.1** ± 0.2	51.6
			✓	✓	72.3 ± 0.9	73.5 ± 0.2	**10.1** ± 0.2	**52.0**
	Fishr + IGA	ω	✓	✓	72.4 ± 0.4	73.1 ± 0.1	**10.1** ± 0.1	51.8

Table 13: **Fishr (gradient covariance) vs. IGA (gradient mean)** on DomainBed's OfficeHome.

Model selection	Algorithm	Gradients in	Warmup	ema	A	C	P	R	Avg
Oracle	ERM	N/A	N/A	N/A	61.7 ± 0.7	53.4 ± 0.3	74.1 ± 0.4	76.2 ± 0.6	66.4
	IGA	$\theta = \omega \oplus \phi$	✗	✗	50.1 ± 2.5	49.6 ± 1.6	59.5 ± 6.7	68.5 ± 1.2	56.9
		ω	✗	✗	62.3 ± 0.3	53.9 ± 0.2	75.2 ± 0.4	77.4 ± 0.1	67.2
		ω	✓	✗	61.9 ± 0.4	52.6 ± 0.6	76.0 ± 0.8	77.5 ± 0.3	67.0
		ω	✓	✓	62.3 ± 1.0	53.4 ± 0.3	76.0 ± 0.7	77.0 ± 0.1	67.2
	Fishr	ω	✗	✗	61.8 ± 0.9	53.8 ± 0.4	**76.6** ± 0.6	77.7 ± 0.2	67.5
			✓	✗	**63.6** ± 0.4	53.2 ± 0.5	75.4 ± 0.5	77.8 ± 0.3	67.5
			✓	✓	63.4 ± 0.8	54.2 ± 0.3	76.4 ± 0.3	**78.5** ± 0.2	68.2
	Fishr + IGA	ω	✓	✓	**63.6** ± 1.0	**54.6** ± 0.5	**76.6** ± 0.2	78.4 ± 0.4	**68.3**
Training	ERM	N/A	N/A	N/A	61.3 ± 0.7	52.4 ± 0.3	75.8 ± 0.1	76.6 ± 0.3	66.5
	IGA	$\theta = \omega \oplus \phi$	✗	✗	51.7 ± 1.3	49.3 ± 1.5	58.6 ± 7.1	69.0 ± 1.1	57.1
		ω	✗	✗	61.9 ± 0.0	53.6 ± 0.9	75.7 ± 0.5	76.0 ± 0.1	66.8
		ω	✓	✗	61.2 ± 0.1	52.2 ± 0.5	76.1 ± 0.2	77.2 ± 0.3	66.7
		ω	✓	✓	61.7 ± 0.5	52.4 ± 0.7	75.9 ± 0.4	77.1 ± 0.2	66.8
	Fishr	ω	✗	✗	**63.8** ± 0.6	52.5 ± 0.5	**76.7** ± 0.6	77.1 ± 1.0	67.5
			✓	✗	62.2 ± 1.0	53.5 ± 0.2	76.6 ± 0.2	77.8 ± 0.4	67.5
			✓	✓	62.4 ± 0.5	**54.4** ± 0.4	76.2 ± 0.5	**78.3** ± 0.1	67.8
	Fishr + IGA	ω	✓	✓	63.3 ± 1.0	54.1 ± 0.3	76.5 ± 0.4	78.2 ± 0.6	**68.0**

Second, Fishr uses a double-stage scheduling inherited from IRM (Arjovsky et al., 2019): the DNN first learns predictive features with standard ERM ($\lambda = 0$) until a given epoch, at which λ

takes its true (high) value to then force domain invariance. **This warmup strategy** slightly increases 'Oracle' results on Colored MNIST (from 58.6% to 59.8% for Fishr, from 58.3% to 59.2% for IGA) but does not seem critical: in particular, it slightly reduces IGA 'Oracle' scores on OfficeHome.

Third, the estimation of gradient variances was improved with an **exponential moving average** (see Section 4.2 and Appendix C.2.1). We now use this strategy with domain-level gradient means for IGA in ω: $\bar{\boldsymbol{g}}_e^t = \gamma \bar{\boldsymbol{g}}_e^{t-1} + (1-\gamma)\boldsymbol{g}_e^t$. This improves IGA (from 67.0% to 67.2% in 'Oracle' on OfficeHome): yet, these scores remain consistently worse than Fishr's (from 67.5% to 68.2%).

In conclusion, this complements the experiments in Section 4.2 which showed that tackling gradient covariance does better than tackling gradient mean: indeed, Fishr performed better than Fish (Shi et al., 2021), AND-mask (Parascandolo et al., 2021) and SAND-mask (Shahtalebi et al., 2021). As a final note, Fishr + IGA — *i.e.*, matching simultaneously gradient means (the first moment) and covariances (the second moment) — performs best. Future works may further analyze the complementary of these gradient-based methods.

C.2.3 Hyperparameter distributions

This section is a preliminary introduction to a meta-discussion, not about the methodology to select the best hyperparameters, but about the methodology to select the hyperparameter distributions in DomainBed. This question has not been discussed in previous works (as far as we know).

After few initial iterations on the main idea of the paper, we had to select the distributions to sample our three hyperparameters from, as described in Table 8. *First*, to select the ema γ distribution, we knew that the authors from Le Roux et al. (2011) have not noticed "any significant difference in validation errors" for different values higher than 0.9. Moreover γ should remain strictly lower than 1. Thus, sampling from $\text{Uniform}(0.9, 0.99)$ seemed appropriate. *Second*, sampling the number of warmup iterations uniformly along training from $\text{Uniform}(0, 5000)$ seemed the most natural and neutral choice. *Lastly*, the choice of the λ distribution was more complex. As a reminder, a low λ inactivates the regularization while an extremely high λ may destabilize the training.

Table 14: **Impact of the λ distribution** from Table 8.

Model selection	λ distribution	CMNIST	RMNIST	VLCS	PACS	OfficeHome	TerraInc	DomainNet	Avg
Oracle	Constant(0) (= ERM)	57.8 ± 0.2	97.8 ± 0.1	77.6 ± 0.3	86.7 ± 0.3	66.4 ± 0.5	53.0 ± 0.3	41.3 ± 0.1	68.7
	$10^{\text{Uniform}(1,4)}$	**68.8** ± 1.4	97.8 ± 0.1	78.2 ± 0.2	86.9 ± 0.2	**68.2** ± 0.2	**53.6** ± 0.4	41.8 ± 0.1	**70.8**
	$10^{\text{Uniform}(1,5)}$	68.7 ± 1.3	97.8 ± 0.0	**78.7** ± 0.3	**87.5** ± 0.1	68.0 ± 0.4	52.2 ± 0.5	**42.0** ± 0.1	70.7
Training	Constant(0) (= ERM)	51.5 ± 0.1	**98.0** ± 0.0	77.5 ± 0.4	85.5 ± 0.2	66.5 ± 0.3	46.1 ± 1.8	40.9 ± 0.1	66.6
	$10^{\text{Uniform}(1,4)}$	**52.0** ± 0.2	97.8 ± 0.0	77.8 ± 0.1	85.5 ± 0.4	**67.8** ± 0.1	**47.4** ± 1.6	41.7 ± 0.0	**67.1**
	$10^{\text{Uniform}(1,5)}$	51.8 ± 0.3	97.9 ± 0.0	**77.9** ± 0.1	85.5 ± 0.6	67.4 ± 0.3	47.2 ± 1.0	**41.8** ± 0.1	**67.1**

In Table 14, we investigate two distributions: $\lambda \sim 10^{\text{Uniform}(1,4)}$ (eventually chosen for Fishr) and $\lambda \sim 10^{\text{Uniform}(1,5)}$. *First*, we observe that results are mostly similar: it confirms that Fishr is consistently better than ERM (where $\lambda = 0$), and in average is the best approach with the 'Oracle' model selection and among the best approaches with the 'Training' model selection. *Second*, the existence of consistent differences in results suggests that the best hyperparameter distribution depends on the dataset at hand and that the performance gap depends on the selection method.

While out of the scope of this paper, we believe these results were important for transparency (along with publishing our code), and may motivate the need for new protocols — for example with bayesian hyperparameter search (Turner et al., 2021) — that future benchmarks may introduce.

C.3 Full DomainBed results

Tables below detail results for each dataset with 'Oracle' and 'Training' model selection methods. We format **first** and second best accuracies. Note that the per-dataset results for Fish (Shi et al., 2021) are not available.

C.3.1 COLORED MNIST

Colored MNIST. Model selection: test-domain validation set (oracle)					
Algorithm	**+90%**	**+80%**	**10%**	**Avg**	**Ranking**
ERM	71.8 ± 0.4	72.9 ± 0.1	28.7 ± 0.5	57.8	16
IRM	72.0 ± 0.1	72.5 ± 0.3	58.5 ± 3.3	67.7	2
GroupDRO	73.5 ± 0.3	73.0 ± 0.3	36.8 ± 2.8	61.1	8
Mixup	72.5 ± 0.2	73.9 ± 0.4	28.6 ± 0.2	58.4	13
MLDG	71.9 ± 0.3	73.5 ± 0.2	29.1 ± 0.9	58.2	14
CORAL	71.1 ± 0.2	73.4 ± 0.2	31.1 ± 1.6	58.6	10
MMD	69.0 ± 2.3	70.4 ± 1.6	50.6 ± 0.2	63.3	4
DANN	72.4 ± 0.5	73.9 ± 0.5	24.9 ± 2.7	57.0	18
CDANN	71.8 ± 0.5	72.9 ± 0.1	33.8 ± 6.4	59.5	9
MTL	71.2 ± 0.2	73.5 ± 0.2	28.0 ± 0.6	57.6	17
SagNet	72.1 ± 0.3	73.2 ± 0.3	29.4 ± 0.5	58.2	14
ARM	**84.9** ± 0.9	**76.8** ± 0.6	27.9 ± 2.1	63.2	5
V-REx	72.8 ± 0.3	73.0 ± 0.3	55.2 ± 4.0	67.0	3
RSC	72.0 ± 0.1	73.2 ± 0.1	30.2 ± 1.6	58.5	12
AND-mask	71.9 ± 0.6	73.6 ± 0.5	30.2 ± 1.4	58.6	10
SAND-mask	79.9 ± 3.8	75.9 ± 1.6	31.6 ± 1.1	62.3	6
Fish				61.8	7
Fishr	74.1 ± 0.6	73.3 ± 0.1	**58.9** ± 3.7	**68.8**	1

Colored MNIST. Model selection: training-domain validation set					
Algorithm	**+90%**	**+80%**	**10%**	**Avg**	**Ranking**
ERM	71.7 ± 0.1	72.9 ± 0.2	10.0 ± 0.1	51.5	12
IRM	72.5 ± 0.1	73.3 ± 0.5	10.2 ± 0.3	52.0	4
GroupDRO	73.1 ± 0.3	73.2 ± 0.2	10.0 ± 0.2	52.1	2
Mixup	72.7 ± 0.4	73.4 ± 0.1	10.1 ± 0.1	52.1	2
MLDG	71.5 ± 0.2	73.1 ± 0.2	9.8 ± 0.1	51.5	12
CORAL	71.6 ± 0.3	73.1 ± 0.1	9.9 ± 0.1	51.5	12
MMD	71.4 ± 0.3	73.1 ± 0.2	9.9 ± 0.3	51.5	12
DANN	71.4 ± 0.9	73.1 ± 0.1	10.0 ± 0.0	51.5	12
CDANN	72.0 ± 0.2	73.0 ± 0.2	10.2 ± 0.1	51.7	8
MTL	70.9 ± 0.2	72.8 ± 0.3	**10.5** ± 0.1	51.4	17
SagNet	71.8 ± 0.2	73.0 ± 0.2	10.3 ± 0.0	51.7	8
ARM	**82.0** ± 0.5	**76.5** ± 0.3	10.2 ± 0.0	**56.2**	1
V-REx	72.4 ± 0.3	72.9 ± 0.4	10.2 ± 0.0	51.8	6
RSC	71.9 ± 0.3	73.1 ± 0.2	10.0 ± 0.2	51.7	8
AND-mask	70.7 ± 0.5	73.3 ± 0.2	10.0 ± 0.1	51.3	18
SAND-mask	72.0 ± 0.5	73.2 ± 0.4	10.3 ± 0.2	51.8	6
Fish				51.6	11
Fishr	72.3 ± 0.9	73.5 ± 0.2	10.1 ± 0.2	52.0	4

C.3.2 ROTATED MNIST

Rotated MNIST. Model selection: test-domain validation set (oracle)								
Algorithm	**0**	**15**	**30**	**45**	**60**	**75**	**Avg**	**Ranking**
ERM	95.3 ± 0.2	98.7 ± 0.1	98.9 ± 0.1	98.7 ± 0.2	98.9 ± 0.0	96.2 ± 0.2	97.8	12
IRM	94.9 ± 0.6	98.7 ± 0.2	98.6 ± 0.1	98.6 ± 0.2	98.7 ± 0.1	95.2 ± 0.3	97.5	16
GroupDRO	95.9 ± 0.1	**99.0** ± 0.1	98.9 ± 0.1	98.8 ± 0.1	98.6 ± 0.1	96.3 ± 0.4	97.9	5
Mixup	95.8 ± 0.3	98.7 ± 0.0	**99.0** ± 0.1	98.8 ± 0.1	98.8 ± 0.1	96.6 ± 0.2	98.0	2
MLDG	95.7 ± 0.2	98.9 ± 0.1	98.8 ± 0.1	**98.9** ± 0.1	98.6 ± 0.1	95.8 ± 0.4	97.8	12
CORAL	**96.2** ± 0.2	98.8 ± 0.1	98.8 ± 0.1	98.8 ± 0.1	98.9 ± 0.1	96.4 ± 0.2	98.0	2
MMD	96.1 ± 0.2	98.9 ± 0.0	**99.0** ± 0.0	98.8 ± 0.0	98.9 ± 0.0	96.4 ± 0.2	98.0	2
DANN	95.9 ± 0.1	98.9 ± 0.1	98.6 ± 0.2	98.7 ± 0.1	98.9 ± 0.0	96.3 ± 0.3	97.9	5
CDANN	95.9 ± 0.2	98.8 ± 0.0	98.7 ± 0.1	**98.9** ± 0.1	98.8 ± 0.1	96.1 ± 0.3	97.9	5
MTL	96.1 ± 0.2	98.9 ± 0.0	**99.0** ± 0.0	98.7 ± 0.1	99.0 ± 0.0	95.8 ± 0.3	97.9	5
SagNet	95.9 ± 0.1	**99.0** ± 0.1	98.9 ± 0.1	98.6 ± 0.1	98.8 ± 0.1	96.3 ± 0.1	97.9	5
ARM	95.9 ± 0.4	**99.0** ± 0.1	98.8 ± 0.1	**98.9** ± 0.1	**99.1** ± 0.1	**96.7** ± 0.2	**98.1**	1
V-REx	95.5 ± 0.2	**99.0** ± 0.0	98.7 ± 0.2	98.8 ± 0.1	98.8 ± 0.0	96.4 ± 0.0	97.9	5
RSC	95.4 ± 0.1	98.6 ± 0.1	98.6 ± 0.1	**98.9** ± 0.0	98.8 ± 0.1	95.4 ± 0.3	97.6	15
AND-mask	94.9 ± 0.1	98.8 ± 0.1	98.8 ± 0.1	98.7 ± 0.2	98.6 ± 0.2	95.5 ± 0.2	97.5	16
SAND-mask	94.7 ± 0.2	98.5 ± 0.2	98.6 ± 0.1	98.6 ± 0.1	98.5 ± 0.1	95.2 ± 0.1	97.4	18
Fish							97.9	11
Fishr	95.8 ± 0.1	98.3 ± 0.1	98.8 ± 0.1	98.6 ± 0.3	98.7 ± 0.1	96.5 ± 0.1	97.8	12

Rotated MNIST. Model selection: training-domain validation set								
Algorithm	**0**	**15**	**30**	**45**	**60**	**75**	**Avg**	**Ranking**
ERM	95.9 ± 0.1	98.9 ± 0.0	98.8 ± 0.0	98.9 ± 0.0	98.9 ± 0.0	96.4 ± 0.0	98.0	2
IRM	95.5 ± 0.1	98.8 ± 0.2	98.7 ± 0.1	98.6 ± 0.1	98.7 ± 0.0	95.9 ± 0.2	97.7	15
GroupDRO	95.6 ± 0.1	98.9 ± 0.1	98.9 ± 0.1	99.0 ± 0.0	98.9 ± 0.0	**96.5** ± 0.2	98.0	2
Mixup	95.8 ± 0.3	98.9 ± 0.0	98.9 ± 0.0	98.9 ± 0.0	98.8 ± 0.1	**96.5** ± 0.3	98.0	2
MLDG	95.8 ± 0.1	98.9 ± 0.1	99.0 ± 0.0	98.9 ± 0.1	99.0 ± 0.0	95.8 ± 0.3	97.9	8
CORAL	95.8 ± 0.3	98.8 ± 0.0	98.9 ± 0.0	99.0 ± 0.0	98.9 ± 0.1	96.4 ± 0.2	98.0	2
MMD	95.6 ± 0.1	98.9 ± 0.1	99.0 ± 0.0	99.0 ± 0.0	98.9 ± 0.0	96.0 ± 0.2	97.9	8
DANN	95.0 ± 0.5	98.9 ± 0.1	99.0 ± 0.0	99.0 ± 0.1	98.9 ± 0.0	96.3 ± 0.2	97.8	13
CDANN	95.7 ± 0.2	98.8 ± 0.0	98.9 ± 0.1	98.9 ± 0.1	98.9 ± 0.1	96.1 ± 0.3	97.9	8
MTL	95.6 ± 0.1	99.0 ± 0.1	99.0 ± 0.0	98.9 ± 0.1	99.0 ± 0.1	95.8 ± 0.2	97.9	8
SagNet	95.9 ± 0.3	98.9 ± 0.1	99.0 ± 0.1	**99.1** ± 0.0	99.0 ± 0.1	96.3 ± 0.1	98.0	2
ARM	**96.7** ± 0.2	**99.1** ± 0.0	99.0 ± 0.0	99.0 ± 0.1	**99.1** ± 0.1	**96.5** ± 0.4	**98.2**	1
V-REx	95.9 ± 0.2	99.0 ± 0.1	98.9 ± 0.1	98.9 ± 0.1	98.7 ± 0.1	96.2 ± 0.2	97.9	8
RSC	94.8 ± 0.5	98.7 ± 0.1	98.8 ± 0.1	98.8 ± 0.0	98.9 ± 0.1	95.9 ± 0.2	97.6	16
AND-mask	94.8 ± 0.2	98.8 ± 0.1	98.9 ± 0.0	98.7 ± 0.0	98.7 ± 0.1	95.5 ± 0.4	97.6	16
SAND-mask	94.5 ± 0.4	98.6 ± 0.1	98.8 ± 0.1	98.7 ± 0.1	98.6 ± 0.0	95.5 ± 0.2	97.4	18
Fish							98.0	2
Fishr	95.0 ± 0.3	98.5 ± 0.0	**99.2** ± 0.1	98.9 ± 0.0	98.9 ± 0.1	96.5 ± 0.0	97.8	13

C.3.3 VLCS

VLCS. Model selection: test-domain validation set (oracle)						
Algorithm	**C**	**L**	**S**	**V**	**Avg**	**Ranking**
ERM	97.6 ± 0.3	67.9 ± 0.7	70.9 ± 0.2	74.0 ± 0.6	77.6	12
IRM	97.3 ± 0.2	66.7 ± 0.1	71.0 ± 2.3	72.8 ± 0.4	76.9	16
GroupDRO	97.7 ± 0.2	65.9 ± 0.2	72.8 ± 0.8	73.4 ± 1.3	77.4	15
Mixup	97.8 ± 0.4	67.2 ± 0.4	71.5 ± 0.2	75.7 ± 0.6	78.1	4
MLDG	97.1 ± 0.5	66.6 ± 0.5	71.5 ± 0.1	75.0 ± 0.9	77.5	14
CORAL	97.3 ± 0.2	67.5 ± 0.6	71.6 ± 0.6	74.5 ± 0.0	77.7	10
MMD	98.8 ± 0.0	66.4 ± 0.4	70.8 ± 0.5	75.6 ± 0.4	77.9	6
DANN	**99.0** ± 0.2	66.3 ± 1.2	73.4 ± 1.4	**80.1** ± 0.5	79.7	2
CDANN	98.2 ± 0.1	**68.8** ± 0.5	**74.3** ± 0.6	78.1 ± 0.5	**79.9**	1
MTL	97.9 ± 0.7	66.1 ± 0.7	72.0 ± 0.4	74.9 ± 1.1	77.7	10
SagNet	97.4 ± 0.3	66.4 ± 0.4	71.6 ± 0.1	75.0 ± 0.8	77.6	12
ARM	97.6 ± 0.6	66.5 ± 0.3	72.7 ± 0.6	74.4 ± 0.7	77.8	7
V-REx	98.4 ± 0.2	66.4 ± 0.7	72.8 ± 0.1	75.0 ± 1.4	78.1	4
RSC	98.0 ± 0.4	67.2 ± 0.3	70.3 ± 1.3	75.6 ± 0.4	77.8	7
AND-mask	98.3 ± 0.3	64.5 ± 0.2	69.3 ± 1.3	73.4 ± 1.3	76.4	17
SAND-mask	97.6 ± 0.3	64.5 ± 0.6	69.7 ± 0.6	73.0 ± 1.2	76.2	18
Fish					77.8	7
Fishr	97.6 ± 0.7	67.3 ± 0.5	72.2 ± 0.9	75.7 ± 0.3	78.2	3

VLCS. Model selection: training-domain validation set						
Algorithm	**C**	**L**	**S**	**V**	**Avg**	**Ranking**
ERM	97.7 ± 0.4	64.3 ± 0.9	73.4 ± 0.5	74.6 ± 1.3	77.5	10
IRM	98.6 ± 0.1	64.9 ± 0.9	73.4 ± 0.6	77.3 ± 0.9	78.5	3
GroupDRO	97.3 ± 0.3	63.4 ± 0.9	69.5 ± 0.8	76.7 ± 0.7	76.7	18
Mixup	98.3 ± 0.6	64.8 ± 1.0	72.1 ± 0.5	74.3 ± 0.8	77.4	13
MLDG	97.4 ± 0.2	**65.2** ± 0.7	71.0 ± 1.4	75.3 ± 1.0	77.2	15
CORAL	98.3 ± 0.1	66.1 ± 1.2	73.4 ± 0.3	**77.5** ± 1.2	**78.8**	1
MMD	97.7 ± 0.1	64.0 ± 1.1	72.8 ± 0.2	75.3 ± 3.3	77.5	10
DANN	**99.0** ± 0.3	65.1 ± 1.4	73.1 ± 0.3	77.2 ± 0.6	78.6	2
CDANN	97.1 ± 0.3	65.1 ± 1.2	70.7 ± 0.8	77.1 ± 1.5	77.5	10
MTL	97.8 ± 0.4	64.3 ± 0.3	71.5 ± 0.7	75.3 ± 1.7	77.2	15
SagNet	97.9 ± 0.4	64.5 ± 0.5	71.4 ± 1.3	**77.5** ± 0.5	77.8	6
ARM	98.7 ± 0.2	63.6 ± 0.7	71.3 ± 1.2	76.7 ± 0.6	77.6	9
V-REx	98.4 ± 0.3	64.4 ± 1.4	**74.1** ± 0.4	76.2 ± 1.3	78.3	4
RSC	97.9 ± 0.1	62.5 ± 0.7	72.3 ± 1.2	75.6 ± 0.8	77.1	17
AND-mask	97.8 ± 0.4	64.3 ± 1.2	73.5 ± 0.7	76.8 ± 2.6	78.1	5
SAND-mask	98.5 ± 0.3	63.6 ± 0.9	70.4 ± 0.8	77.1 ± 0.8	77.4	13
Fish					77.8	6
Fishr	98.9 ± 0.3	64.0 ± 0.5	71.5 ± 0.2	76.8 ± 0.7	77.8	6

C.3.4 PACS

PACS. Model selection: test-domain validation set (oracle)						
Algorithm	**A**	**C**	**P**	**S**	**Avg**	**Ranking**
ERM	86.5 ± 1.0	81.3 ± 0.6	96.2 ± 0.3	**82.7** ± 1.1	86.7	8
IRM	84.2 ± 0.9	79.7 ± 1.5	95.9 ± 0.4	78.3 ± 2.1	84.5	18
GroupDRO	87.5 ± 0.5	**82.9** ± 0.6	97.1 ± 0.3	81.1 ± 1.2	87.1	3
Mixup	87.5 ± 0.4	81.6 ± 0.7	97.4 ± 0.2	80.8 ± 0.9	86.8	6
MLDG	87.0 ± 1.2	82.5 ± 0.9	96.7 ± 0.3	81.2 ± 0.6	86.8	6
CORAL	86.6 ± 0.8	81.8 ± 0.9	97.1 ± 0.5	**82.7** ± 0.6	87.1	3
MMD	**88.1** ± 0.8	82.6 ± 0.7	97.1 ± 0.5	81.2 ± 1.2	**87.2**	1
DANN	87.0 ± 0.4	80.3 ± 0.6	96.8 ± 0.3	76.9 ± 1.1	85.2	17
CDANN	87.7 ± 0.6	80.7 ± 1.2	97.3 ± 0.4	77.6 ± 1.5	85.8	14
MTL	87.0 ± 0.2	82.7 ± 0.8	96.5 ± 0.7	80.5 ± 0.8	86.7	8
SagNet	87.4 ± 0.5	81.2 ± 1.2	96.3 ± 0.8	80.7 ± 1.1	86.4	10
ARM	85.0 ± 1.2	81.4 ± 0.2	95.9 ± 0.3	80.9 ± 0.5	85.8	14
V-REx	87.8 ± 1.2	81.8 ± 0.7	97.4 ± 0.2	82.1 ± 0.7	**87.2**	1
RSC	86.0 ± 0.7	81.8 ± 0.9	96.8 ± 0.7	80.4 ± 0.5	86.2	12
AND-mask	86.4 ± 1.1	80.8 ± 0.9	97.1 ± 0.2	81.3 ± 1.1	86.4	10
SAND-mask	86.1 ± 0.6	80.3 ± 1.0	97.1 ± 0.3	80.0 ± 1.3	85.9	13
Fish					85.8	14
Fishr	87.9 ± 0.6	80.8 ± 0.5	**97.9** ± 0.4	81.1 ± 0.8	86.9	5

PACS. Model selection: training-domain validation set						
Algorithm	**A**	**C**	**P**	**S**	**Avg**	**Ranking**
ERM	84.7 ± 0.4	**80.8** ± 0.6	97.2 ± 0.3	79.3 ± 1.0	85.5	3
IRM	84.8 ± 1.3	76.4 ± 1.1	96.7 ± 0.6	76.1 ± 1.0	83.5	17
GroupDRO	83.5 ± 0.9	79.1 ± 0.6	96.7 ± 0.3	78.3 ± 2.0	84.4	14
Mixup	86.1 ± 0.5	78.9 ± 0.8	97.6 ± 0.1	75.8 ± 1.8	84.6	10
MLDG	85.5 ± 1.4	80.1 ± 1.7	97.4 ± 0.3	76.6 ± 1.1	84.9	8
CORAL	88.3 ± 0.2	80.0 ± 0.5	97.5 ± 0.3	78.8 ± 1.3	86.2	2
MMD	86.1 ± 1.4	79.4 ± 0.9	96.6 ± 0.2	76.5 ± 0.5	84.6	10
DANN	86.4 ± 0.8	77.4 ± 0.8	97.3 ± 0.4	73.5 ± 2.3	83.6	16
CDANN	84.6 ± 1.8	75.5 ± 0.9	96.8 ± 0.3	73.5 ± 0.6	82.6	18
MTL	87.5 ± 0.8	77.1 ± 0.5	96.4 ± 0.8	77.3 ± 1.8	84.6	10
SagNet	87.4 ± 1.0	80.7 ± 0.6	97.1 ± 0.1	**80.0** ± 0.4	**86.3**	1
ARM	86.8 ± 0.6	76.8 ± 0.5	97.4 ± 0.3	79.3 ± 1.2	85.1	7
V-REx	86.0 ± 1.6	79.1 ± 0.6	96.9 ± 0.5	77.7 ± 1.7	84.9	8
RSC	85.4 ± 0.8	79.7 ± 1.8	**97.6** ± 0.3	78.2 ± 1.2	85.2	6
AND-mask	85.3 ± 1.4	79.2 ± 2.0	96.9 ± 0.4	76.2 ± 1.4	84.4	14
SAND-mask	85.8 ± 1.7	79.2 ± 0.8	96.3 ± 0.2	76.9 ± 2.0	84.6	10
Fish					85.5	3
Fishr	**88.4** ± 0.2	78.7 ± 0.7	97.0 ± 0.1	77.8 ± 2.0	85.5	3

C.3.5 OFFICEHOME

OfficeHome. Model selection: test-domain validation set (oracle)						
Algorithm	**A**	**C**	**P**	**R**	**Avg**	**Ranking**
ERM	61.7 ± 0.7	53.4 ± 0.3	74.1 ± 0.4	76.2 ± 0.6	66.4	8
IRM	56.4 ± 3.2	51.2 ± 2.3	71.7 ± 2.7	72.7 ± 2.7	63.0	18
GroupDRO	60.5 ± 1.6	53.1 ± 0.3	75.5 ± 0.3	75.9 ± 0.7	66.2	3
Mixup	63.5 ± 0.2	**54.6** ± 0.4	76.0 ± 0.3	78.0 ± 0.7	68.0	6
MLDG	60.5 ± 0.7	54.2 ± 0.5	75.0 ± 0.2	76.7 ± 0.5	66.6	6
CORAL	**64.8** ± 0.8	54.1 ± 0.9	**76.5** ± 0.4	78.2 ± 0.4	**68.4**	3
MMD	60.4 ± 1.0	53.4 ± 0.5	74.9 ± 0.1	76.1 ± 0.7	66.2	1
DANN	60.6 ± 1.4	51.8 ± 0.7	73.4 ± 0.5	75.5 ± 0.9	65.3	17
CDANN	57.9 ± 0.2	52.1 ± 1.2	74.9 ± 0.7	76.2 ± 0.2	65.3	14
MTL	60.7 ± 0.8	53.5 ± 1.3	75.2 ± 0.6	76.6 ± 0.6	66.5	8
SagNet	62.7 ± 0.5	53.6 ± 0.5	76.0 ± 0.3	77.8 ± 0.1	67.5	10
ARM	58.8 ± 0.5	51.8 ± 0.7	74.0 ± 0.1	74.4 ± 0.2	64.8	14
V-REx	59.6 ± 1.0	53.3 ± 0.3	73.2 ± 0.5	76.6 ± 0.4	65.7	1
RSC	61.7 ± 0.8	53.0 ± 0.9	74.8 ± 0.8	76.3 ± 0.5	66.5	12
AND-mask	60.3 ± 0.5	52.3 ± 0.6	75.1 ± 0.2	76.6 ± 0.3	66.1	10
SAND-mask	59.9 ± 0.7	53.6 ± 0.8	74.3 ± 0.4	75.8 ± 0.5	65.9	13
Fish					66.0	12
Fishr	63.4 ± 0.8	54.2 ± 0.3	76.4 ± 0.3	**78.5** ± 0.2	68.2	5

OfficeHome. Model selection: training-domain validation set						
Algorithm	**A**	**C**	**P**	**R**	**Avg**	**Ranking**
ERM	61.3 ± 0.7	52.4 ± 0.3	75.8 ± 0.1	76.6 ± 0.3	66.5	7
IRM	58.9 ± 2.3	52.2 ± 1.6	72.1 ± 2.9	74.0 ± 2.5	64.3	18
GroupDRO	60.4 ± 0.7	52.7 ± 1.0	75.0 ± 0.7	76.0 ± 0.7	66.0	11
Mixup	62.4 ± 0.8	**54.8** ± 0.6	**76.9** ± 0.3	78.3 ± 0.2	68.1	3
MLDG	61.5 ± 0.9	53.2 ± 0.6	75.0 ± 1.2	77.5 ± 0.4	66.8	6
CORAL	**65.3** ± 0.4	54.4 ± 0.5	76.5 ± 0.1	**78.4** ± 0.5	**68.7**	1
MMD	60.4 ± 0.2	53.3 ± 0.3	74.3 ± 0.1	77.4 ± 0.6	66.3	10
DANN	59.9 ± 1.3	53.0 ± 0.3	73.6 ± 0.7	76.9 ± 0.5	65.9	12
CDANN	61.5 ± 1.4	50.4 ± 2.4	74.4 ± 0.9	76.6 ± 0.8	65.8	13
MTL	61.5 ± 0.7	52.4 ± 0.6	74.9 ± 0.4	76.8 ± 0.4	66.4	8
SagNet	63.4 ± 0.2	**54.8** ± 0.4	75.8 ± 0.4	78.3 ± 0.3	68.1	3
ARM	58.9 ± 0.8	51.0 ± 0.5	74.1 ± 0.1	75.2 ± 0.3	64.8	17
V-REx	60.7 ± 0.9	53.0 ± 0.9	75.3 ± 0.1	76.6 ± 0.5	66.4	8
RSC	60.7 ± 1.4	51.4 ± 0.3	74.8 ± 1.1	75.1 ± 1.3	65.5	16
ANDMask	59.5 ± 1.2	51.7 ± 0.2	73.9 ± 0.4	77.1 ± 0.2	65.6	15
SAND-mask	60.3 ± 0.5	53.3 ± 0.7	73.5 ± 0.7	76.2 ± 0.3	65.8	13
Fish					68.6	2
Fishr	62.4 ± 0.5	54.4 ± 0.4	76.2 ± 0.5	78.3 ± 0.1	67.8	5

C.3.6 TERRAINCOGNITA

TerraIncognita. Model selection: test-domain validation set (oracle)						
Algorithm	**L100**	**L38**	**L43**	**L46**	**Avg**	**Ranking**
ERM	59.4 ± 0.9	49.3 ± 0.6	**60.1** ± 1.1	43.2 ± 0.5	53.0	3
IRM	56.5 ± 2.5	49.8 ± 1.5	57.1 ± 2.2	38.6 ± 1.0	50.5	16
GroupDRO	60.4 ± 1.5	48.3 ± 0.4	58.6 ± 0.8	42.2 ± 0.8	52.4	6
Mixup	67.6 ± 1.8	**51.0** ± 1.3	59.0 ± 0.0	40.0 ± 1.1	**54.4**	1
MLDG	59.2 ± 0.1	49.0 ± 0.9	58.4 ± 0.9	41.4 ± 1.0	52.0	9
CORAL	60.4 ± 0.9	47.2 ± 0.5	59.3 ± 0.4	44.4 ± 0.4	52.8	4
MMD	**60.6** ± 1.1	45.9 ± 0.3	57.8 ± 0.5	43.8 ± 1.2	52.0	9
DANN	55.2 ± 1.9	47.0 ± 0.7	57.2 ± 0.9	42.9 ± 0.9	50.6	15
CDANN	56.3 ± 2.0	47.1 ± 0.9	57.2 ± 1.1	42.4 ± 0.8	50.8	13
MTL	58.4 ± 2.1	48.4 ± 0.8	58.9 ± 0.6	43.0 ± 1.3	52.2	7
SagNet	56.4 ± 1.9	50.5 ± 2.3	59.1 ± 0.5	44.1 ± 0.6	52.5	5
ARM	60.1 ± 1.5	48.3 ± 1.6	55.3 ± 0.6	40.9 ± 1.1	51.2	12
V-REx	56.8 ± 1.7	46.5 ± 0.5	58.4 ± 0.3	43.8 ± 0.3	51.4	11
RSC	59.9 ± 1.4	46.7 ± 0.4	57.8 ± 0.5	44.3 ± 0.6	52.1	8
AND-mask	54.7 ± 1.8	48.4 ± 0.5	55.1 ± 0.5	41.3 ± 0.6	49.8	18
SAND-mask	56.2 ± 1.8	46.3 ± 0.3	55.8 ± 0.4	42.6 ± 1.2	50.2	17
Fish					50.8	13
Fishr	60.4 ± 0.9	50.3 ± 0.3	58.8 ± 0.5	**44.9** ± 0.5	53.6	2

TerraIncognita. Model selection: training-domain validation set						
Algorithm	**L100**	**L38**	**L43**	**L46**	**Avg**	**Ranking**
ERM	49.8 ± 4.4	42.1 ± 1.4	56.9 ± 1.8	35.7 ± 3.9	46.1	10
IRM	54.6 ± 1.3	39.8 ± 1.9	56.2 ± 1.8	39.6 ± 0.8	47.6	4
GroupDRO	41.2 ± 0.7	38.6 ± 2.1	56.7 ± 0.9	36.4 ± 2.1	43.2	16
Mixup	**59.6** ± 2.0	42.2 ± 1.4	55.9 ± 0.8	33.9 ± 1.4	47.9	2
MLDG	54.2 ± 3.0	**44.3** ± 1.1	55.6 ± 0.3	36.9 ± 2.2	47.7	3
CORAL	51.6 ± 2.4	42.2 ± 1.0	57.0 ± 1.0	39.8 ± 2.9	47.6	4
MMD	41.9 ± 3.0	34.8 ± 1.0	57.0 ± 1.9	35.2 ± 1.8	42.2	18
DANN	51.1 ± 3.5	40.6 ± 0.6	57.4 ± 0.5	37.7 ± 1.8	46.7	7
CDANN	47.0 ± 1.9	41.3 ± 4.8	54.9 ± 1.7	39.8 ± 2.3	45.8	11
MTL	49.3 ± 1.2	39.6 ± 6.3	55.6 ± 1.1	37.8 ± 0.8	45.6	12
SagNet	53.0 ± 2.9	43.0 ± 2.5	**57.9** ± 0.6	40.4 ± 1.3	**48.6**	1
ARM	49.3 ± 0.7	38.3 ± 2.4	55.8 ± 0.8	38.7 ± 1.3	45.5	13
V-REx	48.2 ± 4.3	41.7 ± 1.3	56.8 ± 0.8	38.7 ± 3.1	46.4	9
RSC	50.2 ± 2.2	39.2 ± 1.4	56.3 ± 1.4	**40.8** ± 0.6	46.6	8
AND-mask	50.0 ± 2.9	40.2 ± 0.8	53.3 ± 0.7	34.8 ± 1.9	44.6	15
SAND-mask	45.7 ± 2.9	31.6 ± 4.7	55.1 ± 1.0	39.0 ± 1.8	42.9	17
Fish					45.1	14
Fishr	50.2 ± 3.9	43.9 ± 0.8	55.7 ± 2.2	39.8 ± 1.0	47.4	6

C.3.7 DOMAINNET

DomainNet. Model selection: test-domain validation set (oracle)								
Algorithm	**clip**	**info**	**paint**	**quick**	**real**	**sketch**	**Avg**	**Ranking**
ERM	58.6 ± 0.3	19.2 ± 0.2	47.0 ± 0.3	13.2 ± 0.2	59.9 ± 0.3	49.8 ± 0.4	41.3	5
IRM	40.4 ± 6.6	12.1 ± 2.7	31.4 ± 5.7	9.8 ± 1.2	37.7 ± 9.0	36.7 ± 5.3	28.0	17
GroupDRO	47.2 ± 0.5	17.5 ± 0.4	34.2 ± 0.3	9.2 ± 0.4	51.9 ± 0.5	40.1 ± 0.6	33.4	14
Mixup	55.6 ± 0.1	18.7 ± 0.4	45.1 ± 0.5	12.8 ± 0.3	57.6 ± 0.5	48.2 ± 0.4	39.6	8
MLDG	**59.3** ± 0.1	19.6 ± 0.2	46.8 ± 0.2	13.4 ± 0.2	60.1 ± 0.4	50.4 ± 0.3	41.6	4
CORAL	59.2 ± 0.1	19.9 ± 0.2	47.4 ± 0.2	**14.0** ± 0.4	59.8 ± 0.2	50.4 ± 0.4	41.8	2
MMD	32.2 ± 13.3	11.2 ± 4.5	26.8 ± 11.3	8.8 ± 2.2	32.7 ± 13.8	29.0 ± 11.8	23.5	18
DANN	53.1 ± 0.2	18.3 ± 0.1	44.2 ± 0.7	11.9 ± 0.1	55.5 ± 0.4	46.8 ± 0.6	38.3	11
CDANN	54.6 ± 0.4	17.3 ± 0.1	44.2 ± 0.7	12.8 ± 0.2	56.2 ± 0.4	45.9 ± 0.5	38.5	10
MTL	58.0 ± 0.4	19.2 ± 0.2	46.2 ± 0.1	12.7 ± 0.2	59.9 ± 0.1	49.0 ± 0.0	40.8	6
SagNet	57.7 ± 0.3	19.1 ± 0.1	46.3 ± 0.5	13.5 ± 0.4	58.9 ± 0.4	49.5 ± 0.2	40.8	6
ARM	49.6 ± 0.4	16.5 ± 0.3	41.5 ± 0.8	10.8 ± 0.1	53.5 ± 0.3	43.9 ± 0.4	36.0	13
V-REx	43.3 ± 4.5	14.1 ± 1.8	32.5 ± 5.0	9.8 ± 1.1	43.5 ± 5.6	37.7 ± 4.5	30.1	16
RSC	55.0 ± 1.2	18.3 ± 0.5	44.4 ± 0.6	12.5 ± 0.1	55.7 ± 0.7	47.8 ± 0.9	38.9	9
AND-mask	52.3 ± 0.8	17.3 ± 0.5	43.7 ± 1.1	12.3 ± 0.4	55.8 ± 0.4	46.1 ± 0.8	37.9	12
SAND-mask	43.8 ± 1.3	15.2 ± 0.2	38.2 ± 0.6	9.0 ± 0.2	47.1 ± 1.1	39.9 ± 0.6	32.2	15
Fish							**43.4**	1
Fishr	58.3 ± 0.5	**20.2** ± 0.2	**47.9** ± 0.2	13.6 ± 0.3	**60.5** ± 0.3	**50.5** ± 0.3	41.8	2

DomainNet. Model selection: training-domain validation set								
Algorithm	**clip**	**info**	**paint**	**quick**	**real**	**sketch**	**Avg**	**Ranking**
ERM	58.1 ± 0.3	18.8 ± 0.3	46.7 ± 0.3	12.2 ± 0.4	59.6 ± 0.1	49.8 ± 0.4	40.9	5
IRM	48.5 ± 2.8	15.0 ± 1.5	38.3 ± 4.3	10.9 ± 0.5	48.2 ± 5.2	42.3 ± 3.1	33.9	14
GroupDRO	47.2 ± 0.5	17.5 ± 0.4	33.8 ± 0.5	9.3 ± 0.3	51.6 ± 0.4	40.1 ± 0.6	33.3	16
Mixup	55.7 ± 0.3	18.5 ± 0.5	44.3 ± 0.5	12.5 ± 0.4	55.8 ± 0.3	48.2 ± 0.5	39.2	8
MLDG	59.1 ± 0.2	19.1 ± 0.3	45.8 ± 0.7	**13.4** ± 0.3	59.6 ± 0.2	50.2 ± 0.4	41.2	4
CORAL	**59.2** ± 0.1	19.7 ± 0.2	46.6 ± 0.3	**13.4** ± 0.4	59.8 ± 0.2	50.1 ± 0.6	41.5	3
MMD	32.1 ± 13.3	11.0 ± 4.6	26.8 ± 11.3	8.7 ± 2.1	32.7 ± 13.8	28.9 ± 11.9	23.4	18
DANN	53.1 ± 0.2	18.3 ± 0.1	44.2 ± 0.7	11.8 ± 0.1	55.5 ± 0.4	46.8 ± 0.6	38.3	10
CDANN	54.6 ± 0.4	17.3 ± 0.1	43.7 ± 0.9	12.1 ± 0.7	56.2 ± 0.4	45.9 ± 0.5	38.3	10
MTL	57.9 ± 0.5	18.5 ± 0.4	46.0 ± 0.1	12.5 ± 0.1	59.5 ± 0.3	49.2 ± 0.1	40.6	6
SagNet	57.7 ± 0.3	19.0 ± 0.2	45.3 ± 0.3	12.7 ± 0.5	58.1 ± 0.5	48.8 ± 0.2	40.3	7
ARM	49.7 ± 0.3	16.3 ± 0.5	40.9 ± 1.1	9.4 ± 0.1	53.4 ± 0.4	43.5 ± 0.4	35.5	13
V-REx	47.3 ± 3.5	16.0 ± 1.5	35.8 ± 4.6	10.9 ± 0.3	49.6 ± 4.9	42.0 ± 3.0	33.6	15
RSC	55.0 ± 1.2	18.3 ± 0.5	44.4 ± 0.6	12.2 ± 0.2	55.7 ± 0.7	47.8 ± 0.9	38.9	9
AND-mask	52.3 ± 0.8	16.6 ± 0.3	41.6 ± 1.1	11.3 ± 0.1	55.8 ± 0.4	45.4 ± 0.9	37.2	12
SAND-mask	43.8 ± 1.3	14.8 ± 0.3	38.2 ± 0.6	9.0 ± 0.3	47.0 ± 1.1	39.9 ± 0.6	32.1	17
Fish							**42.7**	1
Fishr	58.2 ± 0.5	**20.2** ± 0.2	**47.7** ± 0.3	12.7 ± 0.2	**60.3** ± 0.2	**50.8** ± 0.1	41.7	2