Fishr: Invariant Gradient Variances for Out-of-Distribution Generalization

Alexandre Rame, Corentin Dancette, Matthieu Cord

Fishr: Invariant Gradient Variances for Out-of-Distribution Generalization

Alexandre Ramé [1] **Corentin Dancette** [1] **Matthieu Cord** [1 2]

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 at https://github.com/alexrame/fishr.

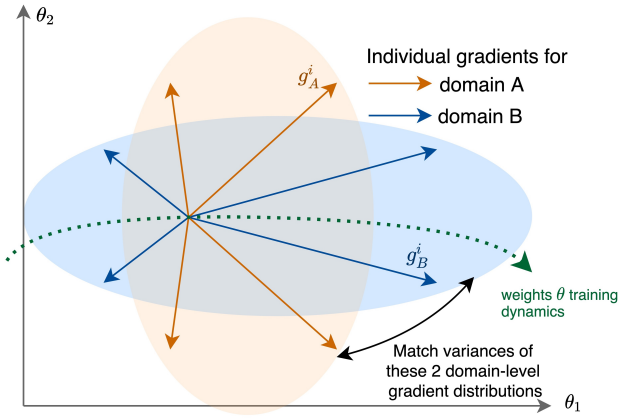

Figure 1: **Fishr principle**. Fishr considers the individual (per-sample) gradients of the loss in the network weights θ. Specifically, Fishr matches the domain-level gradient variances of the distributions across the two training domains: A ($\{\boldsymbol{g}_A^i\}_{i=1}^{n_A}$ in orange) and B ($\{\boldsymbol{g}_B^i\}_{i=1}^{n_B}$ 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.

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 (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 when more complex features are needed (Tenenbaum, 2018; Shah et al., 2020). For example, in the recent 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 (Blanchard et al., 2011; 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).

[1]Sorbonne Université, CNRS, LIP6, Paris, France [2]Valeo.ai. Correspondence to: Alexandre Ramé <alexandre.rame@sorbonne-universite.fr>.

Proceedings of the 39^{th} International Conference on Machine Learning, Baltimore, Maryland, USA, PMLR 162, 2022.

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. Additionally, individual gradients are expressive representations of the input (Fort et al., 2019; Charpiat et al., 2019). 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 variance and disagreements.

In this paper, we propose the Fishr regularization for out-of-distribution generalization in classification for computer vision — 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.

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 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 tackles the gradients only in the classifier, easily implemented with BackPACK (Dangel et al., 2020).

We summarize our contributions as follows:

- We introduce Fishr, a scalable regularization that brings closer the domain-level gradient variances.
- We theoretically justify that Fishr matches domain-level risks and Hessians, and consequently, reduces inconsistencies across domains.

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) when compared with state-of-the-art counterparts. Critically, Fishr is the only method to perform systematically better than ERM on all real datasets — PACS, VLCS, OfficeHome, TerraIncognita and DomainNet.

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 $w_\omega : f_\theta = w_\omega \circ \Phi_\phi$ 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, for instance pictures and drawings of the same objects — 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 **Empirical 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 negative log-likelihood loss. 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 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. Yet, recent works point out pitfalls of IRM (Guo et al., 2021; Kamath et al., 2021; Ahuja et al., 2019), 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). 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 $\mathcal{E} = \{A, B\}$.

A third and most recent line of work promotes **agreements between gradients** with respect to network weights. 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 previous gradient-based methods is the per-domain batch averaging of gradients: this removes more granular statistics, in particular 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 $e \in \mathcal{E}$. These gradient means capture the average learning direction but can not capture gradient disagreements (Sankararaman et al., 2020; Yin et al., 2018). With $\boldsymbol{G}_e = [\boldsymbol{g}_e^i]_{i=1}^{n_e}$ of size $n_e \times |\theta|$, we compute the **domain-level gradient variance** vectors of size $|\theta|$:

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

where the square indicates an element-wise product. 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 3.1. Gradients $\boldsymbol{g}_e^i$ can be computed on all network weights θ. Yet, to reduce the memory and training costs, they will often be computed only on a subset of θ, *e.g.*, only on classification weights ω. This approximation is discussed in Section 4.2.2 and Appendix D.3.2.

3.2. Theoretical analysis

We theoretically motivate our Fishr regularization by leveraging the **domain inconsistency score** introduced in AND-mask (Parascandolo et al., 2021). We first derive a generalized upper bound for this score. Then, we show that Fishr minimizes this upper bound by matching simultaneously **domain-level risks and Hessians**.

3.2.1. INCONSISTENCY FORMALISM

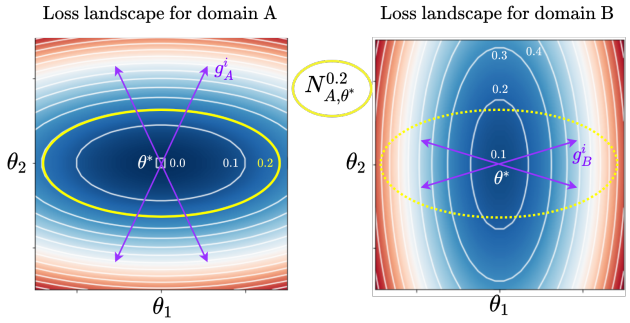

Figure 2: **Loss landscapes around inconsistent weights** θ^* at convergence. $N^{0.2}_{A,\theta^*}$ contains weights θ for which $\mathcal{R}_A(\theta)$ is low (≤ 0.2) but $\mathcal{R}_B(\theta)$ is high (≥ 0.9). This inconsistency is due to conflicting domain-level loss landscapes, specifically gaps between domain-level risks and curvatures at θ^*. This is visible in the disagreements across the variances of gradients $\{\boldsymbol{g}^i_A\}_{i=1}^{n_A}$ and $\{\boldsymbol{g}^i_B\}_{i=1}^{n_B}$.

Parascandolo et al. (2021) argues 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 insight with an inconsistency score:

$$\mathcal{I}^\epsilon(\theta^*) = \max_{(A,B)\in\mathcal{E}^2} \max_{\theta\in N^\epsilon_{A,\theta^*}} |\mathcal{R}_B(\theta) - \mathcal{R}_A(\theta^*)|, \quad (5)$$

where $\theta \in N^\epsilon_{A,\theta^*}$ if there exists 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 risks in B.

For $e \in \mathcal{E}$, the second-order Taylor expansion of $\mathcal{R}_e$ around $\theta^* = 0$ (with a change of variable) gives:

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

where the Hessian $\boldsymbol{H}_e = \nabla^2_\theta \mathcal{R}_e(\theta^*)$ approximates the local curvature of the loss landscape. Moreover, we assume simultaneous convergence, *i.e.*, θ^* is a local minima across all domains: $\nabla_\theta \mathcal{R}_e(\theta^*) = 0$. Thus, locally around θ^*:

$$\begin{aligned}
&\max_{\theta\in N^\epsilon_{A,\theta^*}} |\mathcal{R}_B(\theta) - \mathcal{R}_A(\theta^*)| \\
&\approx \max_{|\mathcal{R}_A(\theta)-\mathcal{R}_A(\theta^*)|\leq\epsilon} |\mathcal{R}_B(\theta) - \mathcal{R}_A(\theta^*)| \\
&\approx \max_{\frac{1}{2}|\theta^\top H_A\theta|\leq\epsilon} \left|\mathcal{R}_B(\theta^*) + \frac{1}{2}\theta^\top H_B\theta - \mathcal{R}_A(\theta^*)\right| \\
&\lessapprox |\mathcal{R}_B(\theta^*) - \mathcal{R}_A(\theta^*)| + \max_{\frac{1}{2}|\theta^\top H_A\theta|\leq\epsilon} \frac{1}{2}\left|\theta^\top H_B\theta\right|,
\end{aligned} \quad (6)$$

where we deduced the last line from the triangle inequality. Appendix A.1 formally demonstrates following equality.

Proposition 1. *Under the quadratic bowl Assumption A.1 with positive definite Hessians, for small ϵ (see Eq. 11):*

$$\begin{aligned}
\mathcal{I}^\epsilon(\theta^*) = \max_{(A,B)\in\mathcal{E}^2} \Big(&\mathcal{R}_B(\theta^*) - \mathcal{R}_A(\theta^*) \\
&+ \max_{\frac{1}{2}\theta^\top H_A\theta\leq\epsilon} \frac{1}{2}\theta^\top H_B\theta\Big).
\end{aligned} \quad (7)$$

The Hessian being positive definite is a standard hypothesis, notably used in Parascandolo et al. (2021), that is empirically reasonable (Sagun et al., 2018): "in only very few steps ... large negative eigenvalues disappear" (Ghorbani et al., 2019).

The first term in the RHS of Proposition 1 is the difference between domain-level risks, whose square is the criterion minimized in V-REx (Krueger et al., 2021). We will prove and show that Fishr forces this term to be small in Section 3.2.2. In contrast, Parascandolo et al. (2021) made the strong assumption: $\mathcal{R}_A(\theta^*) = \mathcal{R}_B(\theta^*) = 0$.

While Parascandolo et al. (2021) ignored this first term, we follow their diagonal approximation of the Hessians to analyze **the second term**. In that case, $\boldsymbol{H}_e = \operatorname{diag}(\lambda^e_1, \cdots, \lambda^e_h)$ with $\forall i \in \{1, \ldots, h\}, \lambda^e_i > 0$. Then:

$$\begin{aligned}
\max_{\frac{1}{2}\theta^\top H_A\theta\leq\epsilon} \frac{1}{2}\theta^\top H_B\theta &= \max_{\|\tilde{\theta}\|_2^2\leq\epsilon} \sum_i \tilde{\theta}_i^2 \lambda_i^B/\lambda_i^A \\
&= \epsilon \times \max_i \lambda_i^B/\lambda_i^A.
\end{aligned} \quad (8)$$

This is large when exists i such that λ^A_i is small but λ^B_i is large: indeed, a small weight perturbation in the direction of the associated eigenvector would change the loss slightly in the domain A but drastically in domain B. Thus, this second term decreases when $\boldsymbol{H}_A$ and $\boldsymbol{H}_B$ have similar eigenvalues. This result holds when Hessians are co-diagonalizable. In conclusion, this explains why forcing $\boldsymbol{H}_A = \boldsymbol{H}_B$ reduces inconsistencies in the loss landscape and thus improves generalization. AND-mask matches Hessians 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. As shown in Section 3.2.3, Fishr proposes a new method to align domain-level Hessians leveraging the close relations between the gradient variance, the Fisher Information and the Hessian.

3.2.2. FISHR MATCHES THE DOMAIN-LEVEL RISKS

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, 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).

The fact that the label and the loss appear in the formula of the gradients has another important consequence: matching gradient distributions also matches training risks, as motivated in V-REx (Krueger et al., 2021). We confirm this insight in Table 2: matching gradient variances with Fishr induces $|\mathcal{R}_A - \mathcal{R}_B|^2 \to 0$ when $\mathcal{E} = \{A, B\}$.

Intuitively, gradient amplitudes are directly weighted by the loss values: multiplying the loss by a constant will also multiply the gradients by the same constant. Thus roughly, if the domain-level empirical training risks are different, then the domain-level gradient norms should also differ.

Theoretically, we prove in Appendix A.2 that Fishr regularization component with reference to the classification bias is exactly the difference between domain-level mean squared errors. We recover the objective from V-REx (Krueger et al., 2021), with a different loss (squared error instead of negative log likelihood). More generally, we show in this Appendix that Fishr in the classifier w_ω acts as a feature-adaptive version of V-REx: the components in Fishr adaptively force the risks to be similar across domains.

3.2.3. FISHR MATCHES THE DOMAIN-LEVEL HESSIANS

The Hessian matrix $\boldsymbol{H} = \sum_{i=1}^{n} \nabla_\theta^2 \ell \left(f_\theta(\boldsymbol{x}^i), \boldsymbol{y}^i\right)$ is of key importance in deep learning. Yet, $\boldsymbol{H}$ cannot be computed efficiently in general. Recent methods (Izmailov et al., 2018; Parascandolo et al., 2021; Foret et al., 2021) tackled the Hessian indirectly by modifying the learning procedure. In contrast, we use the fact that the diagonal of $\boldsymbol{H}$ is approximated by the gradient variance $\mathrm{Var}(\boldsymbol{G})$; this is confirmed in Table 1. This result is derived below from 3 individual and standard approximation steps.

Table 1: **Cosine similarity between Hessian diagonals and gradient variances** $\cos\left(\mathrm{Diag}\left(\boldsymbol{H}_e\right), \mathrm{Var}(\boldsymbol{G}_e)\right)$, for an ERM at convergence on Colored MNIST with the two training domains $e \in \{90\%, 80\%\}$.

	$e = 90\%$	$e = 80\%$
On classifier weights w	0.9999980	0.9999905
On all network weights θ	0.9971040	0.9962264

The Hessian and the Fisher Information Matrix (FIM). The FIM $\boldsymbol{F} = \sum_{i=1}^{n} \mathbb{E}_{\hat{\boldsymbol{y}} \sim P_\theta(\cdot|\boldsymbol{x}^i)} \left[\nabla_\theta \log p_\theta(\hat{\boldsymbol{y}}|\boldsymbol{x}^i) \nabla_\theta \log p_\theta(\hat{\boldsymbol{y}}|\boldsymbol{x}^i)^\top\right]$ (Fisher, 1922; C.R., 1945) approximates the Hessian $\boldsymbol{H}$ with theoretically probably bounded errors under mild assumptions (Schraudolph, 2002).

The 'true' FIM and the 'empirical' FIM. 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 the 'true' FIM $\boldsymbol{F}$ with the 'empirical' FIM $\tilde{\boldsymbol{F}} = \boldsymbol{G}_e^\top \boldsymbol{G}_e = \sum_{i=1}^{n} \nabla_\theta \log p_\theta(\boldsymbol{y}^i|\boldsymbol{x}^i) \nabla_\theta \log p_\theta(\boldsymbol{y}^i|\boldsymbol{x}^i)^\top$ (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). They also have analogous properties: $\mathrm{Tr}(\tilde{\boldsymbol{F}}) \approx \mathrm{Tr}(\boldsymbol{F})$. 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).

The 'empirical' FIM and the gradient covariance. Critically, $\tilde{\boldsymbol{F}}$ is nothing else than the unnormalized uncentered covariance matrix when ℓ is the negative log-likelihood. Thus, the gradient covariance matrix $\boldsymbol{C} = \frac{1}{n-1}\left(\boldsymbol{G}^\top \boldsymbol{G} - \frac{1}{n}\left(\boldsymbol{1}^\top \boldsymbol{G}\right)^\top \left(\boldsymbol{1}^\top \boldsymbol{G}\right)\right)$ of size $|\theta| \times |\theta|$ and $\tilde{\boldsymbol{F}}$ are equivalent (up to the multiplicative constant n) at any first-order stationary point: $\boldsymbol{C} \asymp \tilde{\boldsymbol{F}}$. Overall, this suggests that $\boldsymbol{C}$ and $\boldsymbol{H}$ are closely related (Jastrzebski et al., 2018);.

Consequences for Fishr. Critically, Fishr considers the gradient variance $\mathrm{Var}(\boldsymbol{G})$, *i.e.*, the diagonal components of $\boldsymbol{C}$. In our multi-domain framework, we define the domain-level matrices with the subscript e. Table 2 empirically confirms that matching $\{\mathrm{Diag}(\boldsymbol{C}_e)\}_{e\in\mathcal{E}}$ — i.e., $\{\mathrm{Var}(\boldsymbol{G}_e)\}_{e\in\mathcal{E}}$ — with Fishr forces the domain-level Hessians $\{\mathrm{Diag}(\boldsymbol{H}_e)\}_{e\in\mathcal{E}}$ to be aligned at convergence (on the diagonal for computational reasons). Tackling the second moment of the first-order derivatives enables to regularize the second-order derivatives. Moreover, Appendix C.2.4 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 variances — perform similarly.

Remark 3.2. **Limitation of our approximation.** We acknowledge that approximating the 'true' FIM $\boldsymbol{F}$ by the 'empirical' FIM $\tilde{\boldsymbol{F}}$ is not fully justified theoretically (Martens, 2014; Kunstner et al., 2019). Indeed, this approximation is valid only under strong assumptions, in particular χ^2 convergence of predictions $P_\theta(\cdot|X)$ towards labels $P(Y|X)$ — as detailed in Proposition 1 from Thomas et al. (2020). In this paper, we trade off theoretical guarantees for efficiency.

Remark 3.3. **Diagonal approximation.** The empirical similarities between $\boldsymbol{C}$ and $\boldsymbol{H}$ motivate using **gradient variance**

rather than gradient covariance, which scales down the number of targeted components from $|\theta|^2$ to $|\theta|$. 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. We confirm empirically in Appendix C.2.3 that considering the off-diagonal parts of $\boldsymbol{C}$ performs no better that just matching the diagonals.

Table 2: **Invariance analysis** at convergence on Colored MNIST across the two training domains $\mathcal{E} = \{90\%, 80\%\}$. Compared to ERM, Fishr matches the gradient variance ($\mathrm{Diag}(\boldsymbol{C}_{90\%}) \approx \mathrm{Diag}(\boldsymbol{C}_{80\%})$) in all network weights θ. Most importantly, this enforces invariance in domain-level risks ($\mathcal{R}_{90\%} \approx \mathcal{R}_{80\%}$) and in domain-level Hessians ($\mathrm{Diag}(\boldsymbol{H}_{90\%}) \approx \mathrm{Diag}(\boldsymbol{H}_{80\%})$). The gradient variance, computable efficiently with a unique backpropagation, serves as a proxy for the Hessian. Details and more experiments in Section 4.1 (notably Fig. 3) and in Appendix C.2.1.

	ERM	Fishr
$\lVert\mathrm{Var}(\boldsymbol{G}_{90\%}) - \mathrm{Var}(\boldsymbol{G}_{80\%})\rVert_F^2$	1.6	4.1×10^{-5}
$\lvert\mathcal{R}_{90\%} - \mathcal{R}_{80\%}\rvert^2$	1.0×10^{-2}	3.8×10^{-6}
$\lVert\mathrm{Diag}\left(\boldsymbol{H}_{90\%} - \boldsymbol{H}_{80\%}\right)\rVert_F^2$	2.9×10^{-1}	2.7×10^{-4}

Conclusion. Fishr efficiently matches (1) domain-level empirical risks and (2) domain-level Hessians across the training domains, using gradient variances as a proxy. This will align domain-level loss landscapes, reduce domain inconsistencies and increase domain generalization. In particular, the domain-level Hessian matching illustrates that Fishr is more than just a generalization of gradient-mean approaches such as Fish (Shi et al., 2021).

Finally, we refer the readers to Appendix A.3 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

We prove Fishr effectiveness on Colored MNIST (Arjovsky et al., 2019) and then on the DomainBed benchmark (Gulrajani & Lopez-Paz, 2021). To facilitate reproducibility, the code is available at `https://github.com/alexrame/fishr`. Moreover, we show in Appendix B that Fishr is effective in the linear setting.

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 C.2.2). 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, which was optimized via a grid-search for IRM. More experimental details are provided in Appendix C.1.

Table 3 reports the accuracy averaged over 10 runs with standard deviation. Fishr$_\theta$ (*i.e.*, applying Fishr on all weights θ) obtains the best trade-off between train and test accuracies; notably in test, it reaches 71.2%, or 70.2% when digits are grayscale. Moreover, computing the gradients only in the classifier w_ω performs almost as well (69.5% in test 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. This last experiment shows that we can reduce domain shifts without

Table 3: **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

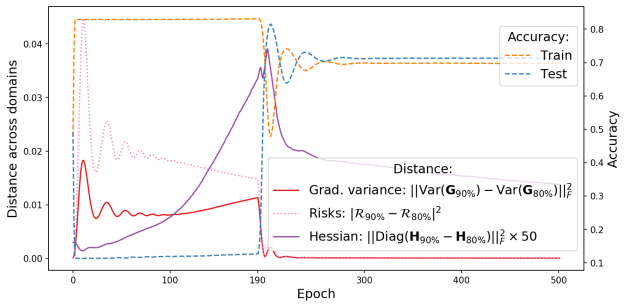

Figure 3: **Colored MNIST dynamics**. At epoch 190, λ strongly steps up: then, the Fishr$_\theta$ regularization matches the domain-level gradient variances (red) across domains $\mathcal{E} = \{90\%, 80\%\}$, and consequently, the training empirical risks (dotted pink) and Hessians (purple). This reduces train accuracy (orange) but increases test accuracy (blue) as the network learns to predict the digit's shape. As shown in Fig. 7, training dynamics are different for ERM.

explicitly forcing the predictors to be simultaneously optimal. These results highlight the effectiveness of gradient variance matching — even with standard hyperparameters — at different layers of the network.

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 empirical risks get closer once the Fishr$_\theta$ gradient variance matching loss is activated after step 190 ($|\mathcal{R}_{90\%} - \mathcal{R}_{80\%}| \to 0$), even though predicting accurately on the domain 90% is easier than on the domain 80%. This confirms insights from Section 3.2.2. Similarly, we observe that Fishr matches Hessians across the two training domains. This is confirmed by further experiments in Appendix C.2, and validates insights from Section 3.2.3. Overall, Fishr regularization reduces train accuracy, but sharply increases test accuracy. Yet, the main drawback of Colored MNIST is its insufficiency to ensure generalization for real-world datasets. Overall, it should be considered as a proof-of-concept.

4.2. DomainBed benchmark

4.2.1. Datasets and procedure

We conduct extensive experiments on **the DomainBed benchmark** (Gulrajani & Lopez-Paz, 2021). 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 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). To limit access to test domain, the framework enforces that all methods are trained with only 20 different configurations of hyperparameters and

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 predictions: $\forall i \in \mathcal{B}, \hat{\boldsymbol{y}}_e^i \leftarrow f_\theta(\boldsymbol{x}_e^i)$
    Compute empirical risks: $\mathcal{R}_e(\theta) \leftarrow \sum_{i \in \mathcal{B}} \ell\left(\hat{\boldsymbol{y}}_e^i, \boldsymbol{y}_e^i\right)$
  end for
  $\mathcal{L}(\theta) = \frac{1}{|\mathcal{E}|} \sum_{e \in \mathcal{E}} \mathcal{R}_e(\theta)$
  {# 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 gradient variances $\boldsymbol{v}_e$ (Eq. 2)
    Update $\boldsymbol{v}_e^{\text{mean}} = \boldsymbol{v}_e \leftarrow \gamma \boldsymbol{v}_e^{\text{mean}} + (1-\gamma)\boldsymbol{v}_e^{\text{iter}}$
  end for
  if iter $\geq i_{\text{warmup}}$ **then**
    $\mathcal{L}(\theta)$ += $\lambda \mathcal{L}_{\text{Fishr}}(\theta)$ (Eq. 3)
  end if
  {# Step 3: gradient descent in the whole network}
  Backpropagate gradients $\nabla_\theta \mathcal{L}(\theta)$ in the network f_θ with standard PyTorch
end for

for the same number of steps. Results are averaged over three trials. This experimental setup is further described in Appendix D.1. By imposing the datasets, the training procedure and controlling the hyperparameter search, DomainBed is arguably the fairer open-source benchmark to rigorously compare the different strategies for OOD generalization.

4.2.2. Implementation details

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. Fishr$_\theta$ and Fishr$_\omega$ performed similarly in previous Section 4.1. This is partly because the gradients in ω still depend on Φ_ϕ. Additionally, as highlighted in Appendix D.3.2, 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, if we used all weights $\theta = (\phi, \omega)$ to compute gradient variances, the invariance in w_ω may be

Table 4: **DomainBed benchmark**. 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
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
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
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
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
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
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
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
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
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
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
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
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
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
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
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
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
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
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**

overshadowed by Φ_ϕ due to $|\omega| \ll |\phi|$. Finally, it's worth noting that this **last-layer approximation** is consistent with the IRM condition (Arjovsky et al., 2019) and is common for unsupervised domain adaptation (Ganin et al., 2016).

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. We show that Fishr is robust to the choice of the sampling distribution for hyperparameter λ in Appendix D.3.3. *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., 2021). 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 controlling 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). The memory overhead is $(|\mathcal{E}| \times |\omega|)$. We study by **ablation** the importance of this warmup strategy and this γ in Appendices D.3.1 and D.3.2.

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 (Shi et al., 2021) leverages a meta-learning algorithm that is impractical as $|\mathcal{E}|$ times longer to train than ERM.

4.2.3. Results

Table 4 summarizes the results on DomainBed using the 'Test-domain' model selection: the validation set (to select the best hyperparameters) follows the same distribution as the test domain. Appendix D.2 reports results with the 'Training-domain' model selection while results are detailed per dataset in Appendix D.4.

ERM was carefully tuned in DomainBed and thus remains a strong baseline. Moreover, all previous methods are far from the best score on at least one dataset. Invariant predictors (IRM, V-REx) and gradient masking (AND-mask) approaches perform poorly on real datasets. Additionally, CORAL not only performs worse than ERM on TerraIncognita, but most importantly fails to detect correlation shifts on Colored MNIST: this is because feature-based approaches

do not take into account the label, as previously stated in Section 3.2.2.

Contrarily, Fishr is the **only method to efficiently tackle correlation and diversity shifts**, as defined in (Ye et al., 2021). Indeed, not only Fishr outperforms ERM on Colored MNIST (68.8% vs. 57.8%), but Fishr also systematically performs 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%). Appendix D.3.2 shows that Fishr performs even better when combined with gradient-mean matching. In summary, **Fishr consistently beats ERM** (despite the restricted hyperparameter search): this is the main point to validate the effectiveness of our method.

Additionally, Fishr performs best after averaging: Firshr reaches 70.8% vs. 69.2% for the second best CORAL. When ignoring the Colored MNIST task, averaging over the 6 other datasets leads to a similar ranking: 1.Fishr(avg=71.1), 2.CORAL(71.0), 3.Mixup(70.8) and 4.ERM(70.5). This arguably partial metric is confirmed by the more robust ranking information; Fishr's median ranking of second reflects that **Fishr is consistently among the best methods**. Overall, Fishr is the state-of-the-art approach, not only in average accuracy, but most importantly in average ranking.

5. Conclusion

In this paper, we addressed the task of out-of-distribution generalization for classification in computer vision. We derive a new and simple regularization — Fishr — that matches the gradient variances across domains as a proxy for matching domain-level risks and Hessians. We prove that this reduces inconsistencies across domains. Fishr reaches state-of-the-art performances on DomainBed when samples from the test domain are available for model selection. Our experiments — reproducible with our open-source implementation — suggest that Fishr would consistently improve a deep classifier for real-world usages when dealing with data from multiple domains. We hope to pave the way towards new gradient-based regularization to improve the generalization abilities of deep neural networks.

Acknowledgments

This work was granted access to the HPC resources of IDRIS under the allocation A0100612449 made by GENCI. We acknowledge the financial support by the ANR agency in the chair VISA-DEEP (ANR-20-CHIA-0022-01).

References

Ahmed, F., Bengio, Y., van Seijen, H., and Courville, A. Systematic generalisation with group invariant predictions. In *ICLR*, 2021. 3

Ahuja, K., Caballero, E., Zhang, D., Bengio, Y., Mitliagkas, I., and Rish, I. Invariance principle meets information bottleneck for out-of-distribution generalization. In *NeurIPS*, 2019. 3, 18

Arjovsky, M., Bottou, L., Gulrajani, I., and Lopez-Paz, D. Invariant risk minimization. *arXiv preprint*, 2019. 1, 2, 3, 6, 7, 8, 16, 19, 20, 23

Becker, S. and Le Cun, Y. Improving the convergence of back-propagation learning with second order methods. In *Connectionist models summer school*, 1988. 6

Beery, S., Van Horn, G., and Perona, P. Recognition in terra incognita. In *ECCV*, 2018. 7, 20

Blanchard, G., Lee, G., and Scott, C. Generalizing from several related classification tasks to a new unlabeled sample. In *NeurIPS*, 2011. 1

Blanchard, G., Deshmukh, A. A., Dogan, U., Lee, G., and Scott, C. Domain generalization by marginal transfer learning. *JMLR*, 2021. 8, 19, 21

Cha, J., Chun, S., Lee, K., Cho, H.-C., Park, S., Lee, Y., and Park, S. SWAD: Domain generalization by seeking flat minima. In *NeurIPS*, 2021. 20

Chang, S., Zhang, Y., Yu, M., and Jaakkola, T. Invariant rationalization. In *ICML*, 2020. 3

Charpiat, G., Girard, N., Felardos, L., and Tarabalka, Y. Input similarity from the neural network perspective. In *NeurIPS*, 2019. 2

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

Creager, E., Jacobsen, J.-H., and Zemel, R. Environment inference for invariant learning. In *ICML*, 2021. 2

D'Amour, A., Heller, K., Moldovan, D., Adlam, B., Alipanahi, B., Beutel, A., Chen, C., Deaton, J., Eisenstein, J., Hoffman, M. D., et al. Underspecification presents challenges for credibility in modern machine learning. *JMLR*, 2020. 21

Dangel, F., Kunstner, F., and Hennig, P. BackPACK: Packing more into backprop. In *ICLR*, 2020. 2, 8, 17

Dangel, F., Tatzel, L., and Hennig, P. Vivit: Curvature access through the generalized gauss-newton's low-rank structure. *arXiv preprint*, 2021. 5

DeGrave, A. J., Janizek, J. D., and Lee, S.-I. Ai for radiographic covid-19 detection selects shortcuts over signal. *Nature Machine Intelligence*, 2021. 1

Deutsch, D. The beginning of infinity: Explanations that transform the world. *Penguin UK*, 2011. 4

Ding, Z. and Fu, Y. Deep domain generalization with structured low-rank constraint. In *TIP*, 2017. 2

Dinh, L., Pascanu, R., Bengio, S., and Bengio, Y. Sharp minima can generalize for deep nets. In *ICML*, 2017. 17

Du, Y., Czarnecki, W. M., Jayakumar, S. M., Farajtabar, M., Pascanu, R., and Lakshminarayanan, B. Adapting auxiliary losses using gradient similarity. *arXiv preprint*, 2018. 3

Faghri, F., Duvenaud, D., Fleet, D. J., and Ba, J. A study of gradient variance in deep learning. *arXiv preprint*, 2020. 6

Fang, C., Xu, Y., and Rockmore, D. N. Unbiased metric learning: On the utilization of multiple datasets and web images for softening bias. In *ICCV*, 2013. 7, 20

Finn, C., Abbeel, P., and Levine, S. Model-agnostic meta-learning for fast adaptation of deep networks. In *ICML*, 2017. 3

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

Foret, P., Kleiner, A., Mobahi, H., and Neyshabur, B. Sharpness-aware minimization for efficiently improving generalization. In *ICLR*, 2021. 5

Fort, S., Nowak, P. K., Jastrzebski, S., and Narayanan, S. Stiffness: A new perspective on generalization in neural networks. *arXiv preprint*, 2019. 2

Frantar, E., Kurtic, E., and Alistarh, D. Efficient matrix-free approximations of second-order information, with applications to pruning and optimization. *arXiv preprint*, 2021. 5

Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., Marchand, M., and Lempitsky, V. Domain-adversarial training of neural networks. *JMLR*, 2016. 1, 2, 8, 19

Geirhos, R., Jacobsen, J.-H., Michaelis, C., Zemel, R., Brendel, W., Bethge, M., and Wichmann, F. A. Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2020. 1

Ghifary, M., Kleijn, W. B., Zhang, M., and Balduzzi, D. Domain generalization for object recognition with multi-task autoencoders. In *ICCV*, 2015. 7, 16, 20

Ghorbani, B., Krishnan, S., and Xiao, Y. An investigation into neural net optimization via hessian eigenvalue density. In *ICML*, 2019. 4

Gong, M., Zhang, K., Liu, T., Tao, D., Glymour, C., and Schölkopf, B. Domain adaptation with conditional transferable components. In *ICML*, 2016. 2

Gulrajani, I. and Lopez-Paz, D. In search of lost domain generalization. In *ICLR*, 2021. 1, 2, 3, 6, 7, 19

Guo, R., Zhang, P., Liu, H., and Kiciman, E. Out-of-distribution prediction with invariant risk minimization: The limitation and an effective fix. *arXiv preprint*, 2021. 3

Gur-Ari, G., Roberts, D. A., and Dyer, E. Gradient descent happens in a tiny subspace. *arXiv preprint*, 2018. 16

Heskes, T. On "natural" learning and pruning in multilayered perceptrons. *Neural Computation*, 2000. 18

Huang, Z., Wang, H., Xing, E. P., and Huang, D. Self-challenging improves cross-domain generalization. In *ECCV*, 2020. 20

Idnani, D. and Kao, J. C. Learning robust representations with score invariant learning. In *ICML UDL Workshop*, 2020. 3

Izmailov, P., Podoprikhin, D., Garipov, T., Vetrov, D., and Wilson, A. Averaging weights leads to wider optima and better generalization. In *UAI*, 2018. 5

Jacot, A., Gabriel, F., and Hongler, C. Neural tangent kernel: Convergence and generalization in neural networks. In *NeurIPS*, 2018. 6, 15

Jastrzebski, S., Kenton, Z., Arpit, D., Ballas, N., Fischer, A., Storkey, A., and Bengio, Y. Three factors influencing minima in SGD. In *ICANN*, 2018. 5, 14

Jastrzebski, S., Arpit, D., Astrand, O., Kerg, G. B., Wang, H., Xiong, C., Socher, R., Cho, K., and Geras, K. J. Catastrophic fisher explosion: Early phase fisher matrix impacts generalization. In *ICML*, 2021. 6

Johansson, F. D., Sontag, D., and Ranganath, R. Support and invertibility in domain-invariant representations. In *AISTATS*, 2019. 3, 5

Kamath, P., Tangella, A., Sutherland, D., and Srebro, N. Does invariant risk minimization capture invariance? In *AISTATS*, 2021. 3

Karakida, R., Akaho, S., and Amari, S.-i. Pathological spectra of the fisher information metric and its variants in deep neural networks. *arXiv preprint*, 2019. 16

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. *arXiv preprint*, 2014. 6, 17, 19

Kirkpatrick, J., Pascanu, R., Rabinowitz, N., Veness, J., Desjardins, G., Rusu, A. A., Milan, K., Quan, J., Ramalho, T., Grabska-Barwinska, A., et al. Overcoming catastrophic forgetting in neural networks. In *PNAS*, 2017. 6

Koh, P. W., Sagawa, S., Marklund, H., Xie, S. M., Zhang, M., Balsubramani, A., Hu, W., Yasunaga, M., Phillips, R. L., Gao, I., et al. Wilds: A benchmark of in-the-wild distribution shifts. *arXiv preprint*, 2020. 2, 3

Kopitkov, D. and Indelman, V. Neural spectrum alignment: Empirical study. *arXiv preprint*, 2019. 16

Koyama, M. and Yamaguchi, S. Out-of-distribution generalization with maximal invariant predictor. *arXiv preprint*, 2020. 3, 19, 20, 22

Krizhevsky, A., Sutskever, I., and Hinton, G. E. Imagenet classification with deep convolutional neural networks. In *NeurIPS*, 2012. 1

Krueger, D., Caballero, E., Jacobsen, J.-H., Zhang, A., Binas, J., Zhang, D., Priol, R. L., and Courville, A. Out-of-distribution generalization via risk extrapolation (rex). In *ICML*, 2021. 1, 3, 4, 5, 6, 8, 15, 20, 21

Kunstner, F., Hennig, P., and Balles, L. Limitations of the empirical fisher approximation for natural gradient descent. In *NeurIPS*, 2019. 5

Le Roux, N., Bengio, Y., and Fitzgibbon, A. Improving first and second-order methods by modeling uncertainty. *Optimization for Machine Learning*, 2011. 8, 21, 24

LeCun, Y., Denker, J., Solla, S., Howard, R., and Jackel, L. Optimal brain damage. In *NeurIPS*, 1990. 6

LeCun, Y., Cortes, C., and Burges, C. Mnist handwritten digit database, 2010. 16, 20

LeCun, Y., Bottou, L., Orr, G. B., and Müller, K.-R. Efficient backprop. In *Neural Networks*. 2012. 6

Li, D., Yang, Y., Song, Y.-Z., and Hospedales, T. M. Deeper, broader and artier domain generalization. In *ICCV*, 2017. 7, 20

Li, D., Yang, Y., Song, Y.-Z., and Hospedales, T. Learning to generalize: Meta-learning for domain generalization. In *AAAI*, 2018a. 19

Li, H., Pan, S. J., Wang, S., and Kot, A. C. Domain generalization with adversarial feature learning. In *CVPR*, 2018b. 2, 19

Li, X., Gu, Q., Zhou, Y., Chen, T., and Banerjee, A. Hessian based analysis of sgd for deep nets: Dynamics and generalization. In *SIAM*, 2020. 5, 17

Li, Y., Gong, M., Tian, X., Liu, T., and Tao, D. Domain generalization via conditional invariant representations. In *AAAI*, 2018c. 3, 19

Liu, L., Zhang, S., Kuang, Z., Zhou, A., Xue, J.-H., Wang, X., Chen, Y., Yang, W., Liao, Q., and Zhang, W. Group fisher pruning for practical network compression. In *ICML*, 2021. 5

Long, M., Wang, J., Ding, G., Sun, J., and Yu, P. S. Transfer joint matching for unsupervised domain adaptation. In *CVPR*, 2014. 3

Lopez-Paz, D. and Ranzato, M. A. Gradient episodic memory for continual learning. In *NeurIPS*, 2017. 3

Maddox, W. J., Tang, S., Moreno, P. G., Wilson, A. G., and Damianou, A. On transfer learning via linearized neural networks. In *NeurIPS workshop*, 2019. 16

Mancini, M., Bulo, S. R., Caputo, B., and Ricci, E. Best sources forward: domain generalization through source-specific nets. In *ICIP*, 2018. 2

Mansilla, L., Echeveste, R., Milone, D. H., and Ferrante, E. Domain generalization via gradient surgery. In *ICCV*, 2021. 3

Martens, J. New insights and perspectives on the natural gradient method. *arXiv preprint*, 2014. 5

Martens, J. and Grosse, R. Optimizing neural networks with kronecker-factored approximate curvature. In *ICML*, 2015. 18

Muandet, K., Balduzzi, D., and Schölkopf, B. Domain generalization via invariant feature representation. In *ICML*, 2013. 1, 3

Nam, H., Lee, H., Park, J., Yoon, W., and Yoo, D. Reducing domain gap by reducing style bias. In *CVPR*, 2021. 20

Nam, J., Cha, H., Ahn, S., Lee, J., and Shin, J. Learning from failure: De-biasing classifier from biased classifier. In *NeurIPS*, 2020. 8, 21

Parascandolo, G., Neitz, A., Orvieto, A., Gresele, L., and Schölkopf, B. Learning explanations that are hard to vary. In *ICLR*, 2021. 2, 3, 4, 5, 6, 20, 24

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., and Chintala, S. Pytorch: An imperative

style, high-performance deep learning library. In *NeurIPS*, 2019. 8

Pearl, J. *Causality*. Cambridge university press, 2009. 3

Peng, X., Bai, Q., Xia, X., Huang, Z., Saenko, K., and Wang, B. Moment matching for multi-source domain adaptation. In *ICCV*, 2019. 7, 20

Peters, J., Bühlmann, P., and Meinshausen, N. Causal inference by using invariant prediction: identification and confidence intervals. *Journal of the Royal Statistical Society*, 2016. 1, 3

Pezeshki, M., Kaba, S.-O., Bengio, Y., Courville, A., Precup, D., and Lajoie, G. Gradient starvation: A learning proclivity in neural networks. In *NeurIPS*, 2021. 8, 16

Rame, A., Kirchmeyer, M., Rahier, T., Rakotomamonjy, A., Gallinari, P., and Cord, M. Diverse weight averaging for out-of-distribution generalization. *arXiv preprint*, 2022. 20

Roberts, M., Driggs, D., Thorpe, M., Gilbey, J., Yeung, M., Ursprung, S., Aviles-Rivero, A. I., Etmann, C., McCague, C., Beer, L., 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. 1

Rojas-Carulla, M., Schölkopf, B., Turner, R., and Peters, J. Invariant models for causal transfer learning. *JMLR*, 2018. 3

Rosenfeld, E., Ravikumar, P. K., and Risteski, A. The risks of invariant risk minimization. In *ICLR*, 2021. 3

Sagawa, S., Koh, P. W., Hashimoto, T. B., and Liang, P. Distributionally robust neural networks. In *ICLR*, 2020a. 2, 19

Sagawa, S., Raghunathan, A., Koh, P. W., and Liang, P. An investigation of why overparameterization exacerbates spurious correlations. In *ICML*, 2020b. 2

Sagun, L., Evci, U., Guney, V. U., Dauphin, Y., and Bottou, L. Empirical analysis of the hessian of over-parametrized neural networks, 2018. 4

Sankararaman, K. A., De, S., Xu, Z., Huang, W. R., and Goldstein, T. The impact of neural network overparameterization on gradient confusion and stochastic gradient descent. In *ICML*, 2020. 2, 3

Schaul, T., Zhang, S., and LeCun, Y. No more pesky learning rates. In *ICML*, 2013. 14

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

Shah, H., Tamuly, K., Raghunathan, A., Jain, P., and Netrapalli, P. The pitfalls of simplicity bias in neural networks. In *NeurIPS*, 2020. 1

Shahtalebi, S., Gagnon-Audet, J.-C., Laleh, T., Faramarzi, M., Ahuja, K., and Rish, I. Sand-mask: An enhanced gradient masking strategy for the discovery of invariances in domain generalization. In *ICML UDL Workshop*, 2021. 3, 4, 19, 20, 24

Shi, Y., Seely, J., Torr, P. H., Siddharth, N., Hannun, A., Usunier, N., and Synnaeve, G. Gradient matching for domain generalization. *arXiv preprint*, 2021. 2, 3, 6, 8, 16, 20, 22, 24

Singh, S. P. and Alistarh, D. Woodfisher: Efficient second-order approximation for neural network compression. In *NeurIPS*, 2020. 5, 17

Sun, B. and Saenko, K. Deep coral: Correlation alignment for deep domain adaptation. In *ECCV*, 2016. 1, 3, 19

Sun, B., Feng, J., and Saenko, K. Return of frustratingly easy domain adaptation. In *AAAI*, 2016. 1, 3

Tenenbaum, J. Building machines that learn and think like people. In *AAMAS*, 2018. 1

Teney, D., Abbasnejad, E., and van den Hengel, A. Unshuffling data for improved generalization. *arXiv preprint*, 2020. 3

Teney, D., Abbasnejad, E., Lucey, S., and van den Hengel, A. Evading the simplicity bias: Training a diverse set of models discovers solutions with superior OOD generalization. *arXiv preprint*, 2021. 21

Theis, L., Korshunova, I., Tejani, A., and Huszár, F. Faster gaze prediction with dense networks and fisher pruning. *arXiv preprint*, 2018. 6

Thomas, V., Pedregosa, F., van Merriënboer, B., Manzagol, P.-A., Bengio, Y., and Roux, N. L. On the interplay between noise and curvature and its effect on optimization and generalization. In *AISTATS*, 2020. 5, 17

Turner, R., Eriksson, D., McCourt, M., Kiili, J., Laaksonen, E., Xu, Z., and Guyon, I. Bayesian optimization is superior to random search for machine learning hyperparameter tuning: Analysis of the black-box optimization challenge 2020. In *NeurIPS*, 2021. 24

Tzeng, E., Hoffman, J., Zhang, N., Saenko, K., and Darrell, T. Deep domain confusion: Maximizing for domain invariance. In *CoRR*, 2014. 3

Valle-Perez, G., Camargo, C. Q., and Louis, A. A. Deep learning generalizes because the parameter-function map is biased towards simple functions. In *ICLR*, 2019. 1

Vapnik, V. N. An overview of statistical learning theory. In *TNN*, 1999. 2, 19

Venkateswara, H., Eusebio, J., Chakraborty, S., and Panchanathan, S. Deep hashing network for unsupervised domain adaptation. In *CVPR*, 2017. 7, 20

Wald, Y., Feder, A., Greenfeld, D., and Shalit, U. On calibration and out-of-domain generalization. In *NeurIPS*, 2021. 21

Wang, Y., Li, H., and Kot, A. C. Heterogeneous domain generalization via domain mixup. In *ICASSP*, 2020. 2

Wu, Y., Inkpen, D., and El-Roby, A. Dual mixup regularized learning for adversarial domain adaptation. In *ECCV*, 2020. 2

Xie, S. M., Kumar, A., Jones, R., Khani, F., Ma, T., and Liang, P. In-n-out: Pre-training and self-training using auxiliary information for out-of-distribution robustness. In *ICLR*, 2021. 2

Yan, S., Song, H., Li, N., Zou, L., and Ren, L. Improve unsupervised domain adaptation with mixup training. *arXiv preprint*, 2020. 19

Yang, G. and Salman, H. A fine-grained spectral perspective on neural networks. *arXiv preprint*, 2019. 16

Ye, N., Li, K., Hong, L., Bai, H., Chen, Y., Zhou, F., and Li, Z. Ood-bench: Benchmarking and understanding out-of-distribution generalization datasets and algorithms. *arXiv preprint*, 2021. 1, 2, 7, 9

Yin, D., Pananjady, A., Lam, M., Papailiopoulos, D., Ramchandran, K., and Bartlett, P. Gradient diversity: a key ingredient for scalable distributed learning. In *AISTATS*, 2018. 2, 3

Yu, T., Kumar, S., Gupta, A., Levine, S., Hausman, K., and Finn, C. Gradient surgery for multi-task learning. In *NeurIPS*, 2020. 3

Zhang, M., Marklund, H., Dhawan, N., Gupta, A., Levine, S., and Finn, C. Adaptive risk minimization: A meta-learning approach for tackling group distribution shift. *arXiv preprint*, 2020. 3, 20, 21

Zhang, X., Cui, P., Xu, R., Zhou, L., He, Y., and Shen, Z. Deep stable learning for out-of-distribution generalization. In *CVPR*, 2021. 2, 21

Zhang, Y., Yu, W., and Turk, G. Learning novel policies for tasks. In *ICML*, 2019. 3

Zhao, H., Des Combes, R. T., Zhang, K., and Gordon, G. On learning invariant representations for domain adaptation. In *ICML*, 2019. 3, 5

These Appendices complement the main paper.

1. We first detail some theoretical points. Appendix A.1 demonstrates our Proposition 1. Appendix A.2 shows that Fishr acts as a feature-adaptive V-REx. Appendix A.3 motivates Fishr with intuitions from the Neural Tangent Kernel theory.
2. Appendix B proves the effectiveness of our approach for a linear toy dataset.
3. Appendix C enriches the Colored MNIST experiment in the IRM setup. In detail, we first describe the experimental setup in Appendix C.1. We then validate in Appendix C.2 some insights provided in the main paper; in particular, Appendix C.2.3 motivates the diagonal approximation of the gradient covariance.
4. Appendix D enriches the DomainBed experiments. After a description of the benchmark protocols in Appendix D.1, Appendix D.2 discusses the model selection strategy. Then Appendix D.3 provides additional experiments to analyze key components of Fishr. Specifically, D.3.1 analyzes the exponential moving average; D.3.2 compares gradient mean versus gradient variance matching and also motivates ignoring the gradients in the features extractor; D.3.3 discusses the methodology to select hyperparameter distributions. Finally, Appendix D.4 provides the per-dataset results.

A. Additional Theoretical Analysis

A.1. Demonstration of Proposition 1 from Section 3.2.1

Assumption A.1. We make the quadratic bowl assumption around the local minima θ^* on all domains : $\forall e \in \mathcal{E}$,

$$\mathcal{R}_e(\theta) = \mathcal{R}_e(\theta^*) + \frac{1}{2}(\theta - \theta^*)^\top H_e (\theta - \theta^*), \tag{9}$$

where H_e is positive definite of eigenvalues $\lambda_1^e \geq \cdots \geq \lambda_h^e > 0$.

Remark A.2. Assumption A.1 is milder on N^ϵ_{e,θ^*} for low ϵ. Indeed, when $\epsilon \to 0$, then $\max_{\theta \in N^\epsilon_{e,\theta^*}} \|\theta - \theta^*\|_2^2 \to 0$ and the quadratic approximation coincides with the second-order Taylor expansion around θ^*. Moreover, this approximation is common in optimization (Schaul et al., 2013; Jastrzebski et al., 2018).

Proposition 2. *(Reformulation of Proposition 1, illustrated in Fig. 4). Let $\epsilon > 0$, weights θ^*. $\forall (A, B) \in \mathcal{E}^2$, with N^ϵ_{A,θ^*} the largest path-connected region of weights space where the risk $\mathcal{R}_A$ remains in an ϵ interval around $\mathcal{R}_A(\theta^*)$, we note:*

$$\begin{aligned} \mathcal{I}^\epsilon(A, B) &= \max_{\theta \in N^\epsilon_{A,\theta^*}} \left| \mathcal{R}_B(\theta) - \mathcal{R}_A(\theta^*) \right|, \\ R(A, B) &= \mathcal{R}_B(\theta^*) - \mathcal{R}_A(\theta^*), \\ H^\epsilon(A, B) &= \max_{\frac{1}{2}(\theta - \theta^*)^\top H_A (\theta - \theta^*) \leq \epsilon} \frac{1}{2}(\theta - \theta^*)^\top H_B (\theta - \theta^*). \end{aligned} \tag{10}$$

If $\forall (A, B) \in \mathcal{E}^2$ such as $R(A, B) < 0$, we have:

$$\epsilon \leq -R(A, B) \times \frac{\lambda_h^A}{\lambda_1^B}, \tag{11}$$

then under previous Assumption A.1,

$$\max_{(A,B) \in \mathcal{E}^2} \mathcal{I}^\epsilon(A, B) = \max_{(A,B) \in \mathcal{E}^2} \left(R(A, B) + H^\epsilon(A, B) \right) \tag{12}$$

Proof We first prove that, under quadratic Assumption A.1, $\forall A \in \mathcal{E}$, $N^\epsilon_{A,\theta^*} = \{\theta \,|\, |\mathcal{R}_A(\theta) - \mathcal{R}_A(\theta^*)| \leq \epsilon\}$. Indeed, the former is always included in the latter by definition. Reciprocally, be given θ in the latter, $\{\lambda\theta^* + (1 - \lambda)\theta | \lambda \in [0, 1]\}$ linearly connects θ^* to θ in parameter space with the risk $\mathcal{R}_A$ remaining in an ϵ interval around $\mathcal{R}_A(\theta^*)$ because $\forall \mu \in [0, 1]$ we have $|\mathcal{R}_A(\mu\theta^* + (1 - \mu)\theta) - \mathcal{R}_A(\theta^*)| = (1 - \mu)^2 |\mathcal{R}_A(\theta) - \mathcal{R}_A(\theta^*)| \leq (1 - \mu)^2 \epsilon \leq \epsilon$.

Therefore $\forall (A, B) \in \mathcal{E}^2$:

$$\mathcal{I}^\epsilon(A, B) = \max_{|\mathcal{R}_A(\theta) - \mathcal{R}_A(\theta^*)| \leq \epsilon} |\mathcal{R}_B(\theta) - \mathcal{R}_A(\theta^*)| = \max_{\frac{1}{2}(\theta - \theta^*)^\top H_A (\theta - \theta^*) \leq \epsilon} \left| R(A, B) + \frac{1}{2}(\theta - \theta^*)^\top H_B (\theta - \theta^*) \right| \tag{13}$$

As the Hessians are positive, $H^\epsilon(A, B) > 0$. We now need to split the analysis based on the sign of $R(A, B)$.

Case $R(A,B) \geq 0$ Both $R(A,B)$ and $H^\epsilon(A,B)$ are non-negative. Removing the absolute value from the RHS of Eq. 13 gives: $\mathcal{I}^\epsilon(A,B) = R(A,B) + H^\epsilon(A,B)$. Taking the maximum over $(A,B) \in \mathcal{E}^2$ where $R(A,B) \geq 0$ gives:

$$\begin{aligned} &\max_{(A,B)\in\mathcal{E}^2 | R(A,B)\geq 0} \mathcal{I}^\epsilon(A,B) \\ &= \max_{(A,B)\in\mathcal{E}^2 | R(A,B)\geq 0} \left(R(A,B) + H^\epsilon(A,B)\right). \end{aligned} \tag{14}$$

Case $R(A,B) < 0$ Leveraging λ_1^B the largest eigenvalue from H_B and λ_h^A the lowest eigenvalue from H_A, we upper bound:

$$H^\epsilon(A,B) \leq \max_{\frac{\lambda_h^A}{2}\|\theta-\theta^*\|_2^2 \leq \epsilon} \frac{\lambda_1^B}{2}\|\theta-\theta^*\|_2^2 = \epsilon \times \frac{\lambda_1^B}{\lambda_h^A}. \tag{15}$$

Then Eq. 11 gives $H^\epsilon(A,B) < -R(A,B)$. Thus the number inside the absolute value from the RHS of Eq. 13 is negative. This leads to: $\mathcal{I}^\epsilon(A,B) = -R(A,B) - H^\epsilon(A,B) < -R(A,B) = R(B,A) < \mathcal{I}^\epsilon(B,A)$. Thus the max over $\mathcal{E}^2$ of function $(A,B) \to \mathcal{I}^\epsilon(A,B)$ can not be achieved for (A,B) with $R(A,B) < 0$. We obtain:

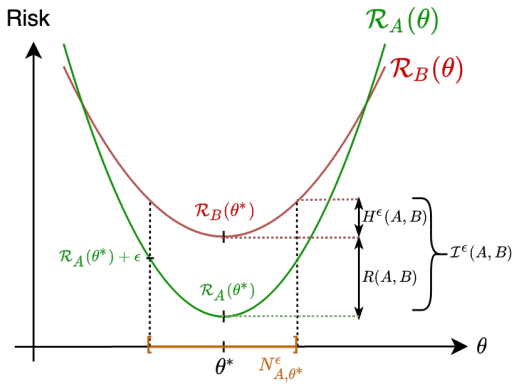

Figure 4: **Inconsistency** $\mathcal{I}^\epsilon(A,B)$ **between domains** A **and** B, decomposed into $R(A,B)$ depending on domain-level risks and $H^\epsilon(A,B)$ depending on domain-level curvatures at θ^*.

$$\max_{(A,B)\in\mathcal{E}^2} \mathcal{I}^\epsilon(A,B) = \max_{(A,B)\in\mathcal{E}^2 | R(A,B)\geq 0} \mathcal{I}^\epsilon(A,B) \tag{16}$$

Similarly, $R(A,B) + H^\epsilon(A,B) \leq 0 < \mathcal{R}(B,A) + H^\epsilon(B,A)$. Thus the max over $\mathcal{E}^2$ of function $(A,B) \to (R(A,B) + H^\epsilon(A,B))$ can not be achieved for (A,B) with $R(A,B) < 0$. We obtain:

$$\max_{(A,B)\in\mathcal{E}^2} \left(R(A,B) + H^\epsilon(A,B)\right) = \max_{(A,B)\in\mathcal{E}^2 | R(A,B)\geq 0} \left(R(A,B) + H^\epsilon(A,B)\right) \tag{17}$$

Conclusion Combining Eq. 14, Eq. 16 and Eq. 17, we conclude the proof.

A.2. Fishr as a feature-adaptive version of V-REx

We delve into the theoretical analysis of the Fishr regularization in the classifier w_ω, that leverages p features extracted from 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: $\boldsymbol{v}_e^b = \frac{1}{n_e}\sum_{i=1}^{n_e}(\hat{y}_e^i - y_e^i)^2$, which is exactly the mean squared error (MSE) between predictions and targets in domain e. Thus, matching gradient variances in b will match risks across domains. This is the objective from V-REx (Krueger et al., 2021), where the squared error has replaced the negative log likelihood.

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 $z_e^i[k]$: matching gradient variances directly matches these weighted squared errors, with k different weighting schemes, that depend on the features distribution. This describes Fishr as a **feature-adaptive version of V-REx** (Krueger et al., 2021). An intuitive example is when features are binary ($z_e^i \in \{0,1\}$); in that case, Fishr matches domain-level risks on groups of samples having a shared feature.

More exactly in Fishr, we match centered gradient variances, equivalent to the uncentered variance gradient matching at convergence under the assumption $\boldsymbol{g}_e \approx 0$. Experiments in Table 5 and in Appendix C.2.4 confirm that centering or not the variances perform similarly.

A.3. 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., 2021).

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 3.2.3), this further motivates the need to match gradient variances during the SGD trajectory — and not only at convergence as in Section 3.2.

B. Experiments on a Linear Example

We experimentally prove that Fishr is effective in the linear setting. To do so, we consider the binary classification dataset introduced in the Section 3.2 from Fish (Shi et al., 2021). Each example is composed of 4 static 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)

C. Colored MNIST in the IRM Setup

C.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 is almost unchanged from `https://github.com/facebookresearch/InvariantRiskMinimization`.

C.2. Empirical validation of some key insights

C.2.1. Hessian matching

Based on empirical works (Li et al., 2020; Singh & Alistarh, 2020; Thomas et al., 2020), we argue in Section 3.2.3 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 2 and during training in Fig. 3. We leveraged the *DiagHessian* method from BackPACK to compute Hessian diagonals, in all network weights θ. Notably, Hessians are impractical in a training objective as computing "Hessian is an order of magnitude more computationally intensive" (see Fig. 9 in Dangel et al. (2020)). This Appendix further analyzes the Hessian trajectory during training.

Fig. 5 illustrates the dynamics for Fishr$_\theta$: following the scheduling previously described in Appendix C.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).

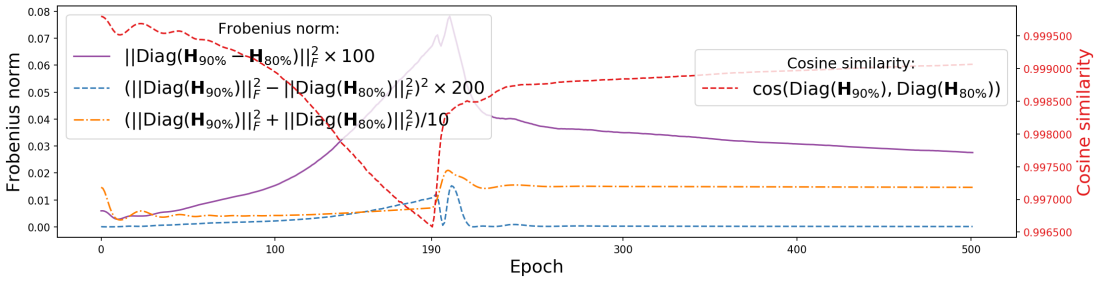

Figure 5: **Hessian dynamics on Colored MNIST with Fishr**: at epoch 190, λ steps up. Then domain-level Hessians are matched across domains (purple). More precisely, they take similar directions — high cosine similarity (red) — and similar norms (blue). The Hessians' norms (orange) remain quite high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

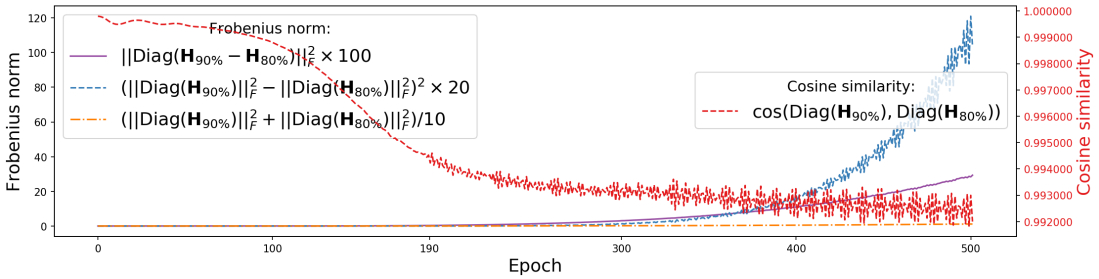

Figure 6: **Hessian dynamics on Colored MNIST with ERM**: $\lambda = 0$ along training. The Frobenius distance between domain-level Hessians (purple) keeps increasing: so does the distance between their norms (blue). Their cosine similarity (red) steadily decreases. The loss landscapes are flat at convergence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

This is also visible in Fig. 7, which is equivalent to Fig. 3, but for ERM (without the Fishr regularization). The distance between domain-level gradient variances (red) keeps increasing across domains $\mathcal{E} = \{90\%, 80\%\}$: so does the distance across Hessians (purple). The distance across risks (pink) decreases, but slower than with Fishr regularization. Overall, the network still predicts the digit's color while only slightly using the digit's shape. That's why the test accuracy (blue) remains low.

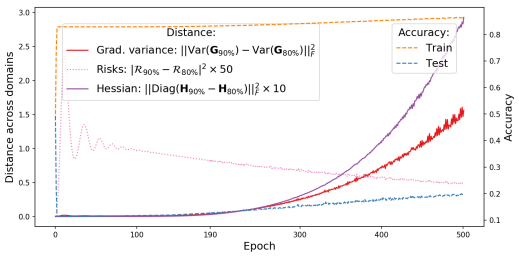

Figure 7: **Colored MNIST dynamics with ERM**.

C.2.2. 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 C.1, we recover that IRM (82.2%) fails when the invariant feature is fully predictive (Ahuja et al., 2019): indeed, it performs worse than ERM (91.8%). In contrast, V-REx and Fishr$_\omega$ perform better (95.3%): in conclusion, 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

C.2.3. 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 hyperparameters were 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$_\omega$)	$\text{Var}(\boldsymbol{G}_e)$	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$_\theta$)	$\text{Var}(\boldsymbol{G}_e)$	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$_\phi$)	$\text{Var}(\boldsymbol{G}_e)$	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

C.2.4. Centered or uncentered variance ?

In Section 3.2.3, 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). This was empirically confirmed in DomainBed: for example, Fishr with either centered or uncentered variances reach 67.8. Still, it's worth noting that explicitly matching simultaneously the gradient centered variances along with the gradient means performs best in Appendix D.3.2.

D. DomainBed

D.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). Scores for IGA (Koyama & Yamaguchi, 2020) are not yet available: yet, for the sake of completeness, we analyze IGA in Appendix D.3.2. 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 D.3.3. 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 (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.

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$(0, 5000)$

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)

We omitted the recent weight averaging approaches (Cha et al., 2021; Rame et al., 2022) whose contribution is complementary to others, that uses a custom hyperparameter search and does not report scores with the 'Test-domain' model selection.

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 C.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.

The convolutional neural network architecture used for the MNIST experiments is the one introduced in DomainBed: note that this is not the same MLP (described in Appendix C.1) as in our proof of concept in Section 4.1. All real datasets leverage a 'ResNet-50' pretrained on ImageNet, with a dropout layer before the newly added dense layer and fine-tuned with frozen batch normalization layers.

D.2. 'Training-domain' model selection

In the main paper, we focus on the 'Test-domain' model selection, where the validation set follows the same distribution as the test domain. This is important to adapt the degree of model invariance according to the test domain. For Fishr, if the domain-dependant correlations are useful in test, the selected λ would be small and Fishr would behave like ERM; in contrast, if the domain-dependant correlations are detrimental in test, the selected λ would be large, and Fishr would improve over ERM by enforcing invariance.

Table 9: **DomainBed with '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
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
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
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
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
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
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**
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
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
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
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
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
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
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
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
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
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
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
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

In Table 9, we use the 'Training-domain' model selection: the validation set is formed by randomly collecting 20% of each training domain. 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. These additional results were not included in the main paper due to space constraints and also because this 'Training-domain' model selection has three clear limitations.

First, learning causal mechanisms can be useless in this 'Training-domain' setup. Indeed, when the correlations are more predictive in training than the causal features, the variant model may be selected over the invariant one. This explains the poor results for all methods in 'Training-domain' Colored MNIST, where the color information is more predictive than the shape information in training. The best model on this task is ARM (Zhang et al., 2020) that uses test time adaptation - thus in a sense uses information from the test-domain - and whose contribution is mostly complementary to ours.

Second, the 'Training-domain' 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.

Third, the 'Test-domain' model selection is more realistic for real applications. Indeed, one user would easily label some samples to validate the efficiency of its algorithm. It's not realistic to believe that the users would simply deploy their new algorithm without at least checking that the performances are correct. We recall that the 'Test-domain' setup in DomainBed benchmark is quite restricting, allowing only one evaluation per choice of hyperparameters, without early-stopping.

That's why Teney et al. (2021) even states that "OOD performance cannot, by definition, be performed with a validation set from the same distribution as the training data". Both opinions being reasonable and arguable, we included 'Training-domain' results for the sake of completeness, where Fishr remains stronger than ERM. Yet, our state-of-the-art results on the 'Test-domain' setup from Table 4 alone are sufficient to prove the usefulness of our approach for real-world applications.

D.3. Fishr component analysis on DomainBed

D.3.1. 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

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

Model selection	Algorithm	ema	+90%	+80%	10%	Avg
Test-domain	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-domain	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
Test-domain	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-domain	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**

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 'Test-domain'). 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 'Test-domain' 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 3 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.

D.3.2. 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 4.2.1. 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 'Test-domain'). Moreover, on OfficeHome in Table 13, IGA hinders learning (56.9% vs. 66.4% for ERM in 'Test-domain'). 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 variance matching**. These experiments provide further insights regarding Fishr main components: specifically, enforcing invariance (1) only in the classifier's

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

Model selection	Algorithm	Gradients in	Warmup	ema	+90%	+80%	10%	Avg
Test-domain	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-domain	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 variance) vs. IGA (gradient mean)** on DomainBed's OfficeHome.

Model selection	Algorithm	Gradients in	Warmup	ema	A	C	P	R	Avg
Test-domain	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-domain	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**

weights ω (2) after a warmup period and (3) with an exponential moving average.

First, Fishr only considers gradient variances 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 'Test-domain' 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 all individual gradients for a ResNet-50 in the GPU memory was not possible on our hardware.

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 'Test-domain' 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 reduces IGA 'Test-domain' scores on OfficeHome.

Third, the estimation of gradient variances was improved with an **exponential moving average** (see Section 4.2.1 and Appendix D.3.1). 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 'Test-domain' 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 4.2.1 which showed that tackling gradient variance 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 variances (the second moment) — performs best. Future works may further analyze the complementary of these gradient-based methods.

D.3.3. HYPERPARAMETER DISTRIBUTIONS

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

Model selection	λ distribution	CMNIST	RMNIST	VLCS	PACS	OfficeHome	TerraInc	DomainNet	Avg
Test-domain	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-domain	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**

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.

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 'Test-domain' model selection and among the best approaches with the 'Training-domain' 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.

D.4. Full DomainBed results

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

D.4.1. COLORED MNIST

Colored MNIST. Model selection: 'Test-domain' validation set					
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

D.4.2. ROTATED MNIST

Rotated MNIST. Model selection: 'Test-domain' validation set								
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

D.4.3. VLCS

VLCS. Model selection: 'Test-domain' validation set

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

D.4.4. PACS

PACS. Model selection: 'Test-domain' validation set						
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

D.4.5. OFFICEHOME

OfficeHome. Model selection: 'Test-domain' validation set						
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

D.4.6. TERRAINCOGNITA

TerraIncognita. Model selection: 'Test-domain' validation set						
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

D.4.7. DomainNet

DomainNet. Model selection: 'Test-domain' validation set								
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