Understanding Hessian Alignment for Domain Generalization

Sobhan Hemati*   Guojun Zhang*   Amir Estiri   Xi Chen
Huawei Noah's Ark Lab
{sobhan.hemati, guojun.zhang, amir.hossein.estiri1, xi.chen4}@huawei.com

Abstract

Out-of-distribution (OOD) generalization is a critical ability for deep learning models in many real-world scenarios including healthcare and autonomous vehicles. Recently, different techniques have been proposed to improve OOD generalization. Among these methods, gradient-based regularizers have shown promising performance compared with other competitors. Despite this success, our understanding of the role of Hessian and gradient alignment in domain generalization is still limited. To address this shortcoming, we analyze the role of the classifier's head Hessian matrix and gradient in domain generalization using recent OOD theory of transferability. Theoretically, we show that spectral norm between the classifier's head Hessian matrices across domains is an upper bound of the transfer measure, a notion of distance between target and source domains. Furthermore, we analyze all the attributes that get aligned when we encourage similarity between Hessians and gradients. Our analysis explains the success of many regularizers like CORAL, IRM, V-REx, Fish, IGA, and Fishr as they regularize part of the classifier's head Hessian and/or gradient. Finally, we propose two simple yet effective methods to match the classifier's head Hessians and gradients in an efficient way, based on the Hessian Gradient Product (HGP) and Hutchinson's method (Hutchinson), and without directly calculating Hessians. We validate the OOD generalization ability of proposed methods in different scenarios, including transferability, severe correlation shift, label shift and diversity shift. Our results show that Hessian alignment methods achieve promising performance on various OOD benchmarks. The code is available here.

1. Introduction

A typical assumption in the design of current supervised deep learning algorithms is identical distributions of test and training data. Unfortunately, in real-world problems, this assumption is not always true (Koyama and Yamaguchi,

*Equal contribution.

2021) and the distribution gap between test and training data degrades the performance of deep learning models (Wang et al., 2022). In practice, not only is there a distribution shift between test and training data, but also the training data does not follow the i.i.d. assumption but contains data from multiple domains. Histopathology medical datasets are a concrete example where each hospital is using different types of staining procedures and/or scanners (Chang et al., 2021). In such a scenario, a deep learning model often fails to generalize its knowledge to unseen domains. Arjovsky et al. (2019) argue that the main reason behind poor OOD generalization is the tendency of deep learning models to capture simple spurious correlations instead of real causal information. In order to improve OOD generalization, the learning algorithm has to learn from invariant mechanisms.

The problem setup in domain generalization (DG) assumes the training set contains data from multiple sources (domains) where the task remains the same across different data sources (Muandet et al., 2013). Furthermore, we assume there is a causal mechanism that remains invariant across different domains and hopefully for OOD data. Although many learning algorithms have been proposed to capture invariance, a recent work by Gulrajani and Lopez-Paz (2020) shows that given a standard experimental setting, the classic Empirical Risk Minimization (ERM) algorithm (Vapnik, 1999), which minimizes the average of losses across different training domains, outperforms many recently proposed domain generalization methods. This suggests that many current algorithms may not be successful in capturing the invariance in data.

Out of the effective DG algorithms, one recent and promising line of research is gradient-based methods which try to capture invariance in gradient space (Parascandolo et al., 2020; Koyama and Yamaguchi, 2021; Shi et al., 2021; Rame et al., 2022). One fast and simple explanation for this success can be derived from the seminal work by Jaakkola and Haussler (1998). According to this work, given a generative model, the gradient with respect to each parameter quantifies the role of that parameter in the generation of a data point. With this intuition, encouraging the similarity between gradients across environments can be translated as
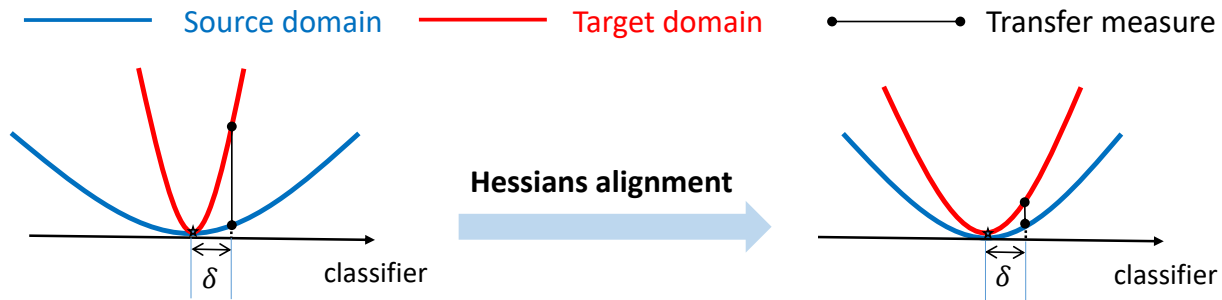
Figure 1: Visualization of Hessian alignment for domain generalization. δ denotes the small deviation from an optimum. Hessian alignment matches the curvature and thus improves the transfer measure between source and target domains.

enforcing the network parameters to contribute the same for different environments which captures a notion of invariance. Beyond gradient matching, the Invariant Learning Consistency (ILC) measure (Parascandolo et al., 2020) motivates matching Hessians across different environments after convergence, given that we found the optimal solution for all environments and that the Hessian matrix is diagonal. Although the efforts in Parascandolo et al. (2020); Koyama and Yamaguchi (2021); Rame et al. (2022) improve our understanding of the role of Hessians and gradients to some extent, the ILC measure is built based on heuristics and is only valid under restricted assumptions like diagonal Hessian matrices. Moreover, it is not exactly clear what attributes are getting aligned when we match gradients or even Hessians. Finally, the proposed methods to align gradients or Hessians (e.g., ANDmask, Parascandolo et al. 2020) seem to be suboptimal as discussed by follow-up works (Shahtalebi et al., 2021; Rame et al., 2022).

To address these limitations, in this paper, we study the role of Hessians and gradients in domain generalization. Since Hessians for the whole neural networks are hard to compute, we focus on Hessians of the classifier head, which contains more information than one usually expects. We summarize our contributions as follows:

- To justify Hessian matching, we utilize a recent concept from statistical learning theory, called transfer measure (Zhang et al., 2021), which describes transferability between target and source domains. Unlike ILC, transfer measure avoids the restrictive assumptions of diagonal Hessians. We theoretically show that the distance between the classifier's head Hessians is an upper bound of the transfer measure.

- Furthermore, we show that Hessians and gradient alignment can be treated as feature matching. Our analysis of feature matching compares other DG algorithms like CORAL (Sun and Saenko, 2016) and V-REx (Krueger

et al., 2021) in a unified framework. Especially, the success of CORAL can be attributed to approximate Hessian alignment using our analysis.

- Last but not least, to match Hessians efficiently, we propose two simple yet effective methods, based on different estimation of the Hessian. To our knowledge, these methods are the first DG algorithms based on Hessian estimation that align Hessians and gradients simultaneously.

Figure 1 provides an intuitive explanation why Hessian alignment can reduce domain shift through minimizing transferability. With Hessian matching, the transfer measure, or the excess risk gap between source and target domains, is minimized. This makes any near-optimal source classifier to be also near-optimal on the target, and thus improves the transferability and OOD generalization.

2. Problem Setting and Related Work

Consider a deep neural network that consists of a feature extractor g and a classifier head $h_{\boldsymbol{\theta}}$ with parameters $\boldsymbol{\theta}$, and suppose the training data contains n different source domains $\mathcal{E}^{source} = \{\mathcal{S}_1, \ldots, \mathcal{S}_n\}$. The main goal in DG is to design a learning algorithm where the trained model can generalize to an unseen target domain $\mathcal{T}$. Denoting $\mathcal{H}$ as the hypothesis class, the classification loss of a classifier $h_{\boldsymbol{\theta}} \in \mathcal{H}$ on a domain $\mathcal{D}$ is:

$$\mathcal{L}_{\mathcal{D}}(\boldsymbol{\theta}) = \mathbb{E}_{(\boldsymbol{x}, \boldsymbol{y}) \sim \mathcal{D}}[\ell(\hat{\boldsymbol{y}}, \boldsymbol{y}; \boldsymbol{\theta})], \qquad (1)$$

where $\boldsymbol{x}$ and $\boldsymbol{y}$ are the input and the associated one-hot label, $\hat{\boldsymbol{y}}$ is the logit (classifier output) and $\hat{\boldsymbol{y}} = h_{\boldsymbol{\theta}}(g(\boldsymbol{x}))$ where $g(\boldsymbol{x})$ is the output of a feature extractor, $h_{\boldsymbol{\theta}}$ is the classifier and $\ell(\hat{\boldsymbol{y}}, \boldsymbol{y}; \boldsymbol{\theta})$ is the cross entropy between $\hat{\boldsymbol{y}}$ and $\boldsymbol{y}$. The classic baseline for OOD generalization is Empirical Risk Minimization (ERM, Vapnik, 1999) which minimizes the average of losses across different training environments

namely $\frac{1}{n}\sum_e \mathcal{L}_{\mathcal{S}_e}$. However, it has been shown that this approach might fail to generalize to out-of-distribution data when there is an easy-to-learn spurious correlation in data (Arjovsky et al., 2019). To tackle this, many algorithms are proposed to avoid domain-specific representations and capture invariance of data from different perspectives. We categorize the most related into four main classes:

Data augmentation. Recently many papers have shown that data augmentation can improve OOD generalization. Gulrajani and Lopez-Paz (2020) showed that the classic ERM with data augmentation can outperform many DG algorithms. Zhang et al. (2018) proposed *mixup*, a data augmentation strategy designed for domain generalization where samples across multiple source domains get mixed to generate new data for training. Ilse et al. (2021) derived an algorithm that select proper data augmentation.

Robust optimization techniques have been used to achieve better OOD generalization (Sagawa et al., 2020; Hu et al., 2018). These techniques minimize the worst-case loss over a set of source domains, and can be helpful when we know the distance between training and test environments.

Learning domain invariant features. Invariant representation learning was initialized by seminal work of Ben-David et al. (2010). A wide range of notions of invariance have been proposed since then. Tzeng et al. (2014) proposed feature matching across domains while Sun and Saenko (2016) considered encouraging the similarity between feature covariance matrices. DANN (Ganin et al., 2016) proposed matching feature distributions with an adversarial network, which is followed by MMD (Li et al., 2018b) and MDAN (Zhao et al., 2018). Zhao et al. (2019) discussed the fundamental tradeoff between feature matching and optimal joint risk. Another line work started from Invariant Risk Minimization (IRM) (Arjovsky et al., 2019), which proposed to encourage the classifier head to be optimal for different domains simultaneously. This idea led to a new notion of invariance as followed by e.g., Risk Extrapolation (V-REx) (Krueger et al., 2021) where authors proposed similarity between losses across training domains.

Gradient matching based algorithms. Recently, gradient alignment has been used for domain generalization (Parascandolo et al., 2020; Shahtalebi et al., 2021; Koyama and Yamaguchi, 2021; Mansilla et al., 2021; Shi et al., 2021; Rame et al., 2022) with good performance. In this regard, Shi et al. (2021) proposed Fish to maximize the inner product of average gradients for different domains. In a similar approach, Koyama and Yamaguchi (2021) proposed the Inter Gradient Alignment (IGA) algorithm that minimizes the variance of average gradients over environments. Parascandolo et al. (2020) developed ANDmask algorithm where the authors proposed ILC, which measures the consistency of local minima for different environments. Parascandolo et al. (2020) proposed ANDmask to reduce the speed of

convergence in directions where landscapes have different curvatures. Inspired by ILC measure (Parascandolo et al., 2020), Rame et al. (2022) proposed Fishr as a better alternative compared with ANDmask to reduce inconsistency. In Fishr, the variance of domain level gradients is minimized.

3. Theory of Hessian Matching

In this section, we study the role of Hessians and gradients in domain generalization from two views: transferability and invariant representation learning.

3.1. Hessian alignment minimizes transfer measure

We motivate our work through the lens of transferability and transfer measure introduced in Zhang et al. (2021). By their definition, domain $\mathcal{S}$ is transferable to domain $\mathcal{T}$ on a hypothesis class $\mathcal{H}$, if given any near-optimal classifier $h_{\boldsymbol{\theta}}$ on $\mathcal{S}$, it also achieves near-optimal performance on $\mathcal{T}$. More precisely, we have:

Definition 1 (**transferability, Zhang et al. 2021**). *$\mathcal{S}$ is $(\delta_{\mathcal{S}}, \delta_{\mathcal{T}})_{\mathcal{H}}$-transferable to $\mathcal{T}$ if for $\delta_{\mathcal{S}} > 0$, there exists $\delta_{\mathcal{T}} > 0$ such that $\mathrm{argmin}(\mathcal{L}_{\mathcal{S}}, \delta_{\mathcal{S}})_{\mathcal{H}} \subseteq \mathrm{argmin}(\mathcal{L}_{\mathcal{T}}, \delta_{\mathcal{T}})_{\mathcal{H}}$, where:*

$$\mathrm{argmin}(\mathcal{L}_{\mathcal{D}}, \delta_{\mathcal{D}})_{\mathcal{H}} := \{h_{\boldsymbol{\theta}} \in \mathcal{H} : \mathcal{L}_{\mathcal{D}}(\boldsymbol{\theta}) \\ \leq \inf_{h_{\boldsymbol{\theta}} \in \mathcal{H}} \mathcal{L}_{\mathcal{D}}(\boldsymbol{\theta}) + \delta_{\mathcal{D}}\}, \mathcal{D} \in \{\mathcal{S}, \mathcal{T}\}.$$

The set $\mathrm{argmin}(\mathcal{L}_{\mathcal{D}}, \delta_{\mathcal{D}})_{\mathcal{H}}$ is called a *δ-minimal set* (Koltchinskii, 2010) of $\mathcal{L}_{\mathcal{D}}$, and represents the near-optimal set of classifiers. With the δ-minimal set, we can define the transfer measure:

Definition 2 (**transfer measure, Zhang et al. 2021**). *Given $\Gamma = \mathrm{argmin}(\mathcal{L}_{\mathcal{S}}, \delta_{\mathcal{S}})_{\mathcal{H}}$, $\mathcal{L}_{\mathcal{S}}^* := \inf_{h_{\boldsymbol{\theta}} \in \Gamma} \mathcal{L}_{\mathcal{S}}(\boldsymbol{\theta})$ and $\mathcal{L}_{\mathcal{T}}^* := \inf_{h_{\boldsymbol{\theta}} \in \Gamma} \mathcal{L}_{\mathcal{T}}(\boldsymbol{\theta})$ we define the transfer measure $\mathrm{T}_{\Gamma}(\mathcal{S}\|\mathcal{T})$ as:*

$$\mathrm{T}_{\Gamma}(\mathcal{S}\|\mathcal{T}) := \sup_{h_{\boldsymbol{\theta}} \in \Gamma} \mathcal{L}_{\mathcal{T}}(\boldsymbol{\theta}) - \mathcal{L}_{\mathcal{T}}^* - (\mathcal{L}_{\mathcal{S}}(\boldsymbol{\theta}) - \mathcal{L}_{\mathcal{S}}^*). \quad (2)$$

The transfer measure in Eq. 2 measures the upper bound of the difference between source domain excess risk ($\mathcal{L}_{\mathcal{S}}(\boldsymbol{\theta}) - \mathcal{L}_{\mathcal{S}}^*$) and the target domain excess risk ($\mathcal{L}_{\mathcal{T}}(\boldsymbol{\theta}) - \mathcal{L}_{\mathcal{T}}^*$), on a neighborhood Γ. Usually, we choose Γ to be a neighborhood of our learned near-optimal source classifier.

Now, we state our main theorem which, under mild assumptions, implies that Hessian alignment effectively reduces the transfer measure and improving transferability.

Theorem 3 (**Hessian alignment**). *Suppose both the source and target domain losses $\mathcal{L}_{\mathcal{S}}$ and $\mathcal{L}_{\mathcal{T}}$ are μ-strongly convex with respect to the classifier head and twice differentiable with the same minimizer, i.e., $\mathrm{argmin}\,\mathcal{L}_{\mathcal{S}} = \mathrm{argmin}\,\mathcal{L}_{\mathcal{T}}$. Then with $\delta = 2\delta_{\mathcal{S}}/\mu$ and $\Gamma = \mathrm{argmin}(\mathcal{L}_{\mathcal{S}}, \delta_{\mathcal{S}})_{\mathcal{H}}$ we have:*

$$\mathrm{T}_{\Gamma}(\mathcal{S}\|\mathcal{T}) \leq \frac{1}{2}\delta^2 \|\boldsymbol{H}_{\mathcal{T}} - \boldsymbol{H}_{\mathcal{S}}\|_2 + o(\delta^2). \quad (3)$$

Proof. Let $\boldsymbol{\theta}^*$ represent the optimal classifier for both source and target domains. Given that $\mathcal{L}_\mathcal{S}(\boldsymbol{\theta})$ is strongly convex, we can write $\inf_{h_{\boldsymbol{\theta}} \in \mathcal{H}} \mathcal{L}_\mathcal{S}(\boldsymbol{\theta}) = \mathcal{L}_\mathcal{S}(\boldsymbol{\theta}^*)$ and subsequently Γ can be written as

$$\mathrm{argmin}(\mathcal{L}_\mathcal{S}, \delta_\mathcal{S})_\mathcal{H} := \{h_{\boldsymbol{\theta}} \in \mathcal{H} : \mathcal{L}_\mathcal{S}(\boldsymbol{\theta}) - \mathcal{L}_\mathcal{S}(\boldsymbol{\theta}^*) \leq \delta_\mathcal{S}\}.$$

Now, we define set $\mathcal{F}_1$ as

$$\mathcal{F}_1 = \{\boldsymbol{\theta} : \mathcal{L}_\mathcal{S}(\boldsymbol{\theta}) - \mathcal{L}_\mathcal{S}(\boldsymbol{\theta}^*) \leq \delta_\mathcal{S}\}. \tag{4}$$

On the other hand, from the μ-strong convexity of $\mathcal{L}_\mathcal{S}(\boldsymbol{\theta})$, we can write

$$\mathcal{L}_\mathcal{S}(\boldsymbol{\theta}) \geq \mathcal{L}_\mathcal{S}(\boldsymbol{\theta}^*) + \nabla_{\boldsymbol{\theta}} \mathcal{L}_\mathcal{S}^\top(\boldsymbol{\theta}^*)(\boldsymbol{\theta} - \boldsymbol{\theta}^*) + \frac{\mu}{2}\|\boldsymbol{\theta} - \boldsymbol{\theta}^*\|_2^2. \tag{5}$$

Given the optimality of $\boldsymbol{\theta}^*$ (and thus $\nabla_{\boldsymbol{\theta}} \mathcal{L}_\mathcal{S}(\boldsymbol{\theta}^*) = \mathbf{0}$), from Eqs. 4 and 5 we obtain that for any $\boldsymbol{\theta} \in \mathcal{F}_1$,

$$\frac{\mu}{2}\|\boldsymbol{\theta} - \boldsymbol{\theta}^*\|_2^2 \leq \mathcal{L}_\mathcal{S}(\boldsymbol{\theta}) - \mathcal{L}_\mathcal{S}(\boldsymbol{\theta}^*) \leq \delta_\mathcal{S}. \tag{6}$$

If we define set $\mathcal{F}_2$ as

$$\mathcal{F}_2 = \{\boldsymbol{\theta} : \frac{\mu}{2}\|\boldsymbol{\theta} - \boldsymbol{\theta}^*\|_2^2 \leq \delta_\mathcal{S}\}. \tag{7}$$

Then (6) implies $\mathcal{F}_1 \subset \mathcal{F}_2$ and as a result, an upper bound of the transfer measure $\mathrm{T}_\Gamma(\mathcal{S}\|\mathcal{T})$ can be written as

$$\sup_{\|\boldsymbol{\theta}-\boldsymbol{\theta}^*\|_2 \leq \delta} \mathcal{L}_\mathcal{T}(\boldsymbol{\theta}) - \mathcal{L}_\mathcal{T}(\boldsymbol{\theta}^*) - (\mathcal{L}_\mathcal{S}(\boldsymbol{\theta}) - \mathcal{L}_\mathcal{S}(\boldsymbol{\theta}^*)), \tag{8}$$

with $\delta = 2\delta_\mathcal{S}/\mu$. Now, we may write the second-order Taylor expansion $\mathcal{L}_\mathcal{S}$ around $\boldsymbol{\theta}^*$ as:

$$\mathcal{L}_\mathcal{S}(\boldsymbol{\theta}) = \mathcal{L}_\mathcal{S}(\boldsymbol{\theta}^*) + (\boldsymbol{\theta} - \boldsymbol{\theta}^*)^\top \nabla_{\boldsymbol{\theta}} \mathcal{L}_\mathcal{S}(\boldsymbol{\theta}^*) +$$
$$+ \frac{1}{2}(\boldsymbol{\theta} - \boldsymbol{\theta}^*)^\top \boldsymbol{H}_\mathcal{S}(\boldsymbol{\theta} - \boldsymbol{\theta}^*) + o(\|\boldsymbol{\theta} - \boldsymbol{\theta}^*\|^2), \tag{9}$$

where given $\nabla_{\boldsymbol{\theta}} \mathcal{L}_\mathcal{S}(\boldsymbol{\theta}^*) = \mathbf{0}$, the source domain excess risk is

$$\mathcal{L}_\mathcal{S}(\boldsymbol{\theta}) - \mathcal{L}_\mathcal{S}(\boldsymbol{\theta}^*) = \frac{1}{2}\Delta\boldsymbol{\theta}^\top \boldsymbol{H}_\mathcal{S}\Delta\boldsymbol{\theta} + o(\|\boldsymbol{\theta} - \boldsymbol{\theta}^*\|^2), \tag{10}$$

with $\Delta\boldsymbol{\theta} = \boldsymbol{\theta} - \boldsymbol{\theta}^*$. Following a similar path for the target domain and considering that $\boldsymbol{\theta}^*$ is the optimal solution for both source and target domains, the transfer measure $\mathrm{T}_\Gamma(\mathcal{S}\|\mathcal{T})$ is upper bounded by

$$\sup_{\|\boldsymbol{\theta}-\boldsymbol{\theta}^*\|_2 \leq \delta} \frac{1}{2}(\boldsymbol{\theta} - \boldsymbol{\theta}^*)^\top (\boldsymbol{H}_\mathcal{T} - \boldsymbol{H}_\mathcal{S})(\boldsymbol{\theta} - \boldsymbol{\theta}^*) + o(\delta^2), \tag{11}$$

which, from the definition of spectral norm, results in

$$\mathrm{T}_\Gamma(\mathcal{S}\|\mathcal{T}) \leq \frac{1}{2}\delta^2\|\boldsymbol{H}_\mathcal{T} - \boldsymbol{H}_\mathcal{S}\|_2 + o(\delta^2), \tag{12}$$

and the proof is complete. □

Note that convexity and differentiability are easily satisfied if we use the standard linear classifier head and cross entropy loss with softmax. Since in practice we do not have access to the target domain, we minimize the distance between available source domains. For simplicity, we replace the spectral norm with the Frobenius norm (note that Frobenius norm is an upper bound, i.e., $\|\cdot\|_2 \leq \|\cdot\|_F$). As opposed to pairwise regularizers which grow quadratically with the number of environments, we minimize the variance of Hessian distances across domains, namely

$$\frac{1}{n}\sum_{e=1}^n \|\boldsymbol{H}_{\mathcal{S}_e} - \overline{\boldsymbol{H}_\mathcal{S}}\|_2^2 \text{ where } \overline{\boldsymbol{H}_\mathcal{S}} = \frac{1}{n}\sum_{e=1}^n \boldsymbol{H}_{\mathcal{S}_e}$$

and $\boldsymbol{H}_{\mathcal{S}_e}$ is the Hessian matrix for the e-th source domain. In this way, computation is linear with respect to the number of environments.

3.2. Hessian and gradient alignment is feature matching

In §3.1 we assumed that the minimizers of source and target are the same. In order to align them, gradient alignment is often necessary. To get more insight into the attributes that get matched while the Hessians and gradients get aligned, in Proposition 4, we study the Hessian and gradient structures with respect to the classifier head parameters.

Proposition 4 (Alignment attributes in Hessian and gradient). *Let $\hat{y}_p$ and y_p be the network prediction and true target with the p-th class, z_i be the i-th feature before the classifier. Denote the classifier's parameters as $w_{k,i}$, the element in row k and column i of the classifier weight matrix, and b_k as the bias term for the k-th neuron. Matching the gradients and Hessians w.r.t. the classifier head across domains aligns the following:*

$$\frac{\partial \ell}{\partial b_p} = (\hat{y}_p - y_p), \ (Error) \tag{13}$$

$$\frac{\partial \mathcal{L}}{\partial w_{p,q}} = (\hat{y}_p - y_p)z_q, \ (Error \times Feature) \tag{14}$$

$$\frac{\partial^2 \ell}{\partial b_u \partial b_v} = \hat{y}_u(\delta_{u,v} - \hat{y}_v), \ (Logit) \tag{15}$$

$$\frac{\partial^2 \ell}{\partial w_{p,q} \partial b_u} = z_q \hat{y}_p(\delta_{p,u} - \hat{y}_u), \ (Logit \times Feature) \tag{16}$$

$$\frac{\partial^2 \ell}{\partial w_{p,q} \partial w_{u,v}} = \hat{y}_p z_q z_v(\delta_{p,u} - \hat{y}_u), \ (Logit \times Covariance) \tag{17}$$

where $\delta_{p,u}$ is the Kronecker delta which is 1 if $p = u$ and 0 otherwise and u, v, p, q, k, i are dummy indices.

Here the error means the difference between the true one-hot label and our prediction. Eqs. 13–17 show that

Table 1: The alignment attributes for different Domain Generalization algorithms where Loss is $(y_p - \hat{y}_p)^2$, Feature z_q, Covariance $z_q z_v$, Error $y_p - \hat{y}_p$, Error $\times$ Feature $(\hat{y}_p - y_p)z_q$, Logit $\hat{y}_u(\delta_{u,v} - \hat{y}_v)$, Logit $\times$ Feature $z_q \hat{y}_p(\delta_{p,u} - \hat{y}_u)$, and Logit $\times$ Covariance is $\hat{y}_p z_q z_v(\delta_{p,u} - \hat{y}_u)$. Gradient alignment method matches Error and Error $\times$ Feature while matching Hessians would align Logit, Logit $\times$ Feature, Logit $\times$ Covariance. [1]To be precise, the Loss that get aligned in V-Rex is negative log likelihood instead of squared error. [2]To be precise, Fishr aligns Loss $\times$ Feature.

Alignment attribute	Loss	Feature	Covariance	Error	Error × Feature	Logit	Logit × Feature	Logit × Covariance
V-Rex	✓[1]	✗	✗	✗	✗	✗	✗	✗
CORAL	✗	✓	✓	✗	✗	✗	✗	✗
IGA	✗	✗	✗	✓	✓	✗	✗	✗
Fish	✗	✗	✗	✓	✓	✗	✗	✗
Fishr	✓	✗	✗	✗	✓[2]	✗	✗	✗
Hessian Alignment	✗	✗	✗	✓	✓	✓	✓	✓

matching Hessians and gradients with respect to classifier parameters will align the errors, features weighted by errors, logits, features weighted by logits, and covariance weighted by logits across different domains simultaneously. Although gradient alignment can be helpful before convergence, when the training is close to convergence, the gradients go to zero and aligning them is not helpful anymore. On the other hand, the attributes extracted from the Hessian matrix can remain non-zero both before and after convergence and aligning them can boost OOD performance.

Proposition 4 shows that matching gradients and Hessians can be seen as a generalization of other works like V-Rex (Krueger et al., 2021), CORAL (Sun and Saenko, 2016), Fish (Shi et al., 2021), IGA (Koyama and Yamaguchi, 2021), and Fishr (Rame et al., 2022), as these algorithms only partially realize the matching of loss/error, feature or covariance. We summarize alignments in different methods in Table 1 and reveal their connection to Hessian and gradient alignment. One interesting observation is that similar to Hessian alignment, CORAL also matches the feature and its covariance, which opens the venue to understand the success of CORAL with Hessian matching. In the supplementary, we also present similar results for regression tasks.

4. Efficient Hessian matching

So far, we have shown that aligning Hessians across domains can reduce the transfer measure (or in other words, increase transferability) and match the representations at different levels i.e., logits, features, errors and covariances. However, given the computational complexity of calculating Hessian matrices, it is quite challenging to directly minimize $\|\boldsymbol{H}_{\mathcal{S}_1} - \boldsymbol{H}_{\mathcal{S}_2}\|_F$. To align Hessian matrices efficiently, we propose two approaches, aligning Hessian-gradient products and matching Hessian diagonals using Hutchinson's trace estimator (Bekas et al., 2007).

4.1. Hessian Gradient Product (HGP)

For the Hessian-gradient product matching we propose the following loss:

$$\mathcal{L}_{\mathrm{HGP}} = \frac{1}{n}\sum_{e=1}^{n}\mathcal{L}_{\mathcal{S}_e} + \alpha\|\boldsymbol{H}_{\mathcal{S}_e}\nabla_{\boldsymbol{\theta}}\mathcal{L}_{\mathcal{S}_e} - \overline{\boldsymbol{H}_{\mathcal{S}}\nabla_{\boldsymbol{\theta}}\mathcal{L}_{\mathcal{S}}}\|_2^2 +$$
$$+ \beta\|\nabla_{\boldsymbol{\theta}}\mathcal{L}_{\mathcal{S}_e} - \overline{\nabla_{\boldsymbol{\theta}}\mathcal{L}_{\mathcal{S}}}\|_2^2, \quad (18)$$

where α and β are regularization parameters,

$$\overline{\nabla_{\boldsymbol{\theta}}\mathcal{L}_{\mathcal{S}}} = \frac{1}{n}\sum_{e=1}^{n}\nabla_{\boldsymbol{\theta}}\mathcal{L}_{\mathcal{S}_e}, \text{ and} \quad (19)$$

$$\overline{\boldsymbol{H}_{\mathcal{S}}\nabla_{\boldsymbol{\theta}}\mathcal{L}_{\mathcal{S}}} = \frac{1}{n}\sum_{e=1}^{n}\boldsymbol{H}_{\mathcal{S}_e}\nabla_{\boldsymbol{\theta}}\mathcal{L}_{\mathcal{S}_e}. \quad (20)$$

Given that $\nabla_{\boldsymbol{\theta}}\mathcal{L}_{\mathcal{S}_e}$ and $\overline{\nabla_{\boldsymbol{\theta}}\mathcal{L}_{\mathcal{S}}}$ are close enough, minimizing eq. 18 reduces the distance between $\boldsymbol{H}_{\mathcal{S}_e}$ and $\overline{\boldsymbol{H}_{\mathcal{S}}}$. We can efficiently compute the HGP term without directly calculating Hessians, using the following expression

$$\boldsymbol{H}_{\mathcal{S}_e}\nabla_{\boldsymbol{\theta}}\mathcal{L}_{\mathcal{S}_e} = \|\nabla_{\boldsymbol{\theta}}\mathcal{L}_{\mathcal{S}_e}\| \cdot \nabla_{\boldsymbol{\theta}}\|\nabla_{\boldsymbol{\theta}}\mathcal{L}_{\mathcal{S}_e}\|, \quad (21)$$

which follows from $\nabla_{\boldsymbol{x}}\|\boldsymbol{x}\|_2 = \boldsymbol{x}/\|\boldsymbol{x}\|_2$ with $\boldsymbol{x} \in \mathbb{R}^d$ and chain rule. For domain e, to calculate the exact value of the Hessian-gradient product, we only need two rounds of back-propagation: one for $\nabla_{\boldsymbol{\theta}}\mathcal{L}_{\mathcal{S}_e}$ and another for $\nabla_{\boldsymbol{\theta}}\|\nabla_{\boldsymbol{\theta}}\mathcal{L}_{\mathcal{S}_e}\|$.

4.2. Hutchinson's method

In order to estimate Hessians more accurately, we propose another approach to estimate the Hessian diagonal using Hutchinson's method (Bekas et al., 2007):

$$\boldsymbol{D}_{\mathcal{S}_e} = \mathrm{diag}(\boldsymbol{H}_{\mathcal{S}_e}) = \mathbb{E}_{\boldsymbol{r}}[\boldsymbol{r} \odot (\boldsymbol{H}_{\mathcal{S}_e} \cdot \boldsymbol{r})], \quad (22)$$

where $\mathbb{E}_{\boldsymbol{r}}$ is expectation of random variable $\boldsymbol{r}$ sampled from Rademacher distribution and $\odot$ is Hadamard product. In

practice, the expectation is replaced with the mean over a set of samples. Note that $\boldsymbol{H}_{\mathcal{S}_e} \cdot \boldsymbol{r}$ can be calculated efficiently by calculating $\nabla_{\boldsymbol{\theta}} \left(\nabla_{\boldsymbol{\theta}} \mathcal{L}_{\mathcal{S}_e}^{\top} \boldsymbol{r} \right)$. Given that we have estimated the Hessian diagonal, the $\mathcal{L}_{\text{Hutchinson}}$ loss is

$$\mathcal{L}_{\text{Hutchinson}} = \frac{1}{n} \sum_{e=1}^{n} \mathcal{L}_{\mathcal{S}_e} + \alpha \| \boldsymbol{D}_{\mathcal{S}_e} - \overline{\boldsymbol{D}_{\mathcal{S}}} \|_2^2 +$$
$$\beta \| \nabla_{\boldsymbol{\theta}} \mathcal{L}_{\mathcal{S}_e} - \overline{\nabla_{\boldsymbol{\theta}} \mathcal{L}_{\mathcal{S}}} \|_2^2. \quad (23)$$

Compared to HGP, Hutchinson's method is more expensive to compute, as it requires an estimation with sampling. In practice, we use 100 samples for each domain. The trade-off is that Hutchinson's method gives us a more accurate matching algorithm for Hessian, since after training, the Hessian matrix is often close to diagonal (Skorski, 2021), whereas the gradient (and thus the Hessian-gradient product) becomes nearly zero.

5. Experiments

To comprehensively validate the effectiveness of HGP and Hutchinson, we evaluate the proposed algorithms from multiple views, in terms of transferability (Zhang et al., 2021), OOD generalization on datasets with correlation shift (Colored MNIST), correlation shift with label shift (imbalanced Colored MNIST), and diversity shift (Ye et al., 2021), including PACS (Li et al., 2017) and OfficeHome (Venkateswara et al., 2017) datasets from DomainBed.

5.1. Transferability

To show the role of Hessian discrepancy in improving transferability (i.e, Theorem 3), we use the algorithm presented in Zhang et al. (2021), which computes the worst-case gap $\sup_{\|\boldsymbol{\theta} - \boldsymbol{\theta}^*\| \leq \delta} \mathcal{L}_{\mathcal{S}_i}(\boldsymbol{\theta}) - \mathcal{L}_{\mathcal{S}_j}(\boldsymbol{\theta})$ among all pairs of (i, j), to evaluate transferability among multiple domains. First, we find the domains i and j such that $\mathcal{L}_{\mathcal{S}_i}$ and $\mathcal{L}_{\mathcal{S}_j}$ are maximized and minimized respectively and consider the $\mathcal{L}_{\mathcal{S}_i} - \mathcal{L}_{\mathcal{S}_j}$ as the gap. Then, we run an ascent optimizer on classifier $\boldsymbol{\theta}$ to maximize gap and project $\boldsymbol{\theta}$ on to Euclidean ball $\|\boldsymbol{\theta} - \boldsymbol{\theta}^*\| \leq \delta$. We repeat this procedure for multiple rounds and report the test target accuracy for the worst-case gap. Figure 2 shows that OOD accuracies of both HGP and Hutchinson are robust against the attack of an ascent optimizer, which implies that minimizing $\|\boldsymbol{H}_{\mathcal{S}_1} - \boldsymbol{H}_{\mathcal{S}_2}\|_F$ improves transferability.

5.2. OOD generalization under correlation shift using Colored MNIST

Our Colored MNIST (CMNIST) setup is from Arjovsky et al. (2019), where the dataset contains two training and one test environments and the objective is a binary classification task in which class zero and one contain digits less than 5 and greater and equal 5 respectively. Further, the labels have
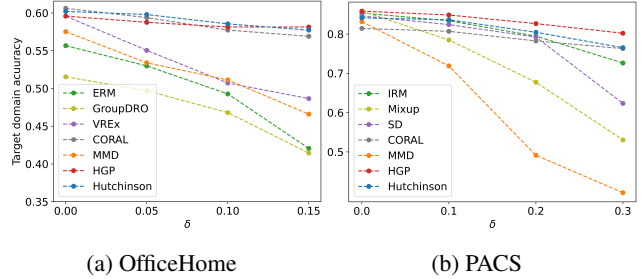

(a) OfficeHome   (b) PACS

Figure 2: OOD accuracy of HGP and Hutchinson compared with multiple baselines on OfficeHome and PACS datasets with different values of δ for the ascent optimizer.

been flipped with a probability 0.25 and each image has been colored either red or yellow such that the colors are heavily correlated with the class label. However, this color correlation is spurious as it has been reversed for test data. The correlations of labels with colors are +90% and +80% for training environments and -90% for the test environment.

Given the structure of data, if our learning algorithm only learns the spurious correlation (in this case color), the test accuracy will be 14%, while if our model learns the invariant features, we can achieve 75% test accuracy. The CMNIST setup (Arjovsky et al., 2019) we use has also been followed by Fishr (Rame et al., 2022) and V-Rex (Krueger et al., 2021). The only difference is that we replace the regularizer in IRM with our proposed regularizers in HGP and Hutchinson. In this setting, a simple fully connected network is used for training where we train the model for 501 steps and at step 190 the regularization parameters α and β increase from 1 to 91257.18 suddenly (found by grid search for IRM). We repeated the experiments 10 times and reported the mean and average over these runs.

Table 2: Comparison of ERM, IRM, V-Rex, Fishr, and our proposed methods HGP and Hutchinson, on Colored MNIST experiment in introduced in IRM (Arjovsky et al., 2019) while the same hyperparameters have been used.

Method	Train acc.	Test acc.
ERM	86.4 ± 0.2	14.0 ± 0.7
IRM	71.0 ± 0.5	65.6 ± 1.8
V-REx	71.7 ± 1.5	67.2 ± 1.5
Fishr	71.0 ± 0.9	69.5 ± 1.0
HGP	71.0 ± 1.5	69.4 ± 1.3
Hutchinson	61.7 ± 1.9	**74.0 ± 1.2**

Table 2 compares the performance of HGP and Hutchinson against common DG baselines. As can be seen, while HGP achieves competitive performance against the state-of-the-art Fishr algorithm, Hutchinson outperforms all methods
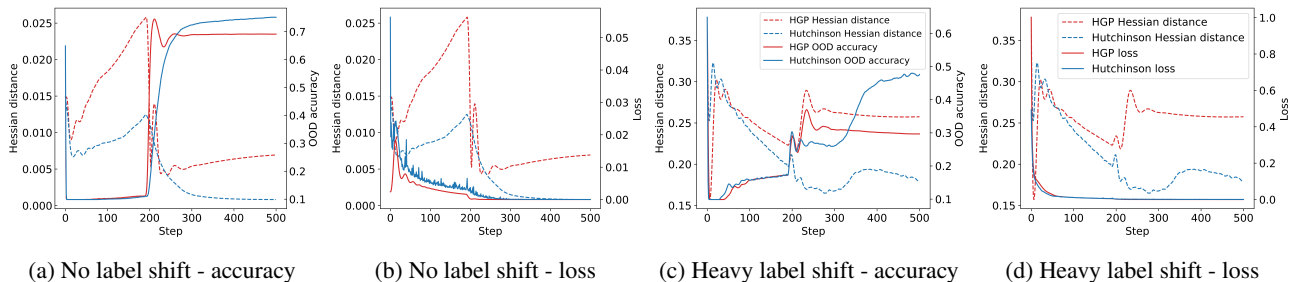
| (a) No label shift - accuracy | (b) No label shift - loss | (c) Heavy label shift - accuracy | (d) Heavy label shift - loss |

Figure 3: Correlation between Hessian distances ($\|\boldsymbol{H}_{\mathcal{S}_1} - \boldsymbol{H}_{\mathcal{S}_2}\|_F$) and OOD accuracies/losses for HGP and Hutchinson regularization during the training for Colored MNIST. (a), (b): no label shift; (c), (d): heavy label shift.

by a large gap and obtains the near-optimal OOD accuracy of 74%, which is close to the maximum achievable accuracy of 75%.

5.3. OOD generalization under label shift

In order to explore the robustness of our proposed algorithm to label shift, we designed an imbalanced Colored MNIST dataset which is the same as the Colored MNIST dataset but we modified the dataset in a way that each of the two domains contain 95% of data points from one class and 5% from the other class. In fact, in the imbalanced Colored MNIST dataset, both correlation shift and label shift exist. We repeated the experiments in Table 2 on imbalanced Colored MNIST and reported the results in Table 3. Looking at Table 3, in the heavy label shift case, the Hutchinson achieves the highest OOD accuracy which outperforms the best-performing algorithm by a margin of 12%.

Table 3: Comparison of ERM, IRM, V-Rex, Fishr, and our proposed methods HGP and Hutchinson on Imbalanced Colored MNIST where each domain has 95% from one class and 5% from other class. Except for the imposed label shift, the setting is same as Colored MNIST experiment introduced in IRM (Arjovsky et al., 2019).

Method	Train acc.	Test acc.
ERM	86.4 ± 0.1	16.7 ± 0.1
IRM	84.9 ± 0.1	14.3 ± 1.4
V-REx	83.3 ± 0.2	35.1 ± 1.2
Fishr	75.7 ± 2.7	35.5 ± 5.3
HGP	83.0 ± 0.2	30.0 ± 0.9
Hutchinson	79.4 ± 0.3	$\mathbf{47.7 \pm 1.4}$

5.4. Training dynamics for Colored MNIST and Imbalanced Colored MNIST

Similar to earlier works (Arjovsky et al., 2019; Krueger et al., 2021; Rame et al., 2022), we study the training and OOD

generalization dynamics of HGP and Hutchinson on CMNIST and imbalanced CMNIST datasets. Figure 3a and Figure 3b show the Hessian distance $\|\boldsymbol{H}_{\mathcal{S}_1} - \boldsymbol{H}_{\mathcal{S}_2}\|_F$ and OOD accuracy/loss of HGP and Hutchinson during the training. From Figure 3a when regularization parameters increase at step 190, the Hessian distance significantly drops and immediately OOD accuracy increases which suggests a strong negative correlation between the Hessian distance and OOD accuracy. This correlation is more evident for Hutchinson compared to HGP since after step 300, the Hessian distance keeps decreasing and OOD accuracy increases correspondingly. In contrast, for HGP, after step 300 when we are close to convergence, OOD accuracy becomes stable and even there is a slight increase in the Hessian distance. This is because close to convergence the gradients become zero and HGP loss cannot align Hessians as well as Hutchinson.
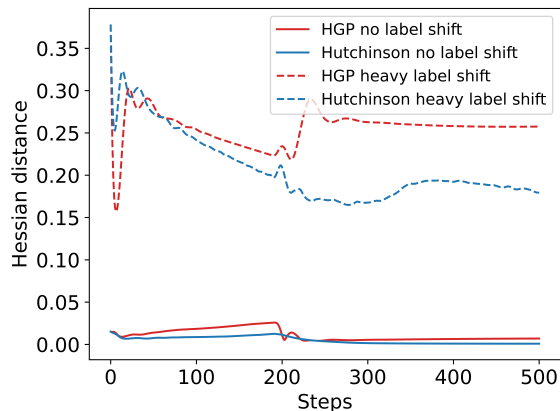

Figure 4: The Hessian distances $\|\boldsymbol{H}_{\mathcal{S}_1} - \boldsymbol{H}_{\mathcal{S}_2}\|_F$ in Hutchinson and HGP regularization losses during the training steps for no label shift and heavy label shift in Colored MNIST experiment. Clearly, for heavy label shift, the Hessian distance is significantly more than no label shift case.

For the heavy label shift case, Figure 3c and 3d show that it is much more difficult to decrease the Hessian distance in

general. Besides, Hutchinson is more capable of reducing the Hessian distance compared to HGP and subsequently, it achieves better OOD accuracy.

Finally, Figure 4 compares Hessian distance while we use HGP and Hutchinson regularizers in no label shift and heavy label shift datasets. Clearly, for heavy label shift, the Hessian distance is significantly larger which shows that transferability between domains becomes more difficult. This experiment validates our argument that Hessian distance captures transferability between domains.

5.5. OOD generalization under diversity shift

In this section we use the Domainbed benchmark (Gulra-jani and Lopez-Paz, 2020) to evaluate OOD accuracy of HGP and Hutchinson. We validate our proposed algorithms on the VLCS (Fang et al., 2013), PACS (Li et al., 2017), OfficeHome (Venkateswara et al., 2017), and DomainNet (Peng et al., 2019) datasets from the DomainBed benchmark. Domainbed provides a standard scheme to validate the performance of DG algorithms in a fair manner with different model selections. Table 4 provides the performance of HGP and Hutchinson methods against multiple benchmarks for leave-one-domain-out model selection reported from Domainbed. It shows that our Hutchinson method gives the state-of-the-art OOD performance in most cases. For HGP, our results are not as good as expected since the gradients diminish near convergence. The most competitive method to Hutchinson is CORAL, which we showed approximately aligns Hessians (by matching covariances) and gradients (by matching features). The results for other model selections are reported in the supplementary material.

5.6. Robustness to adversarial shift

Robustness of deep learning models to adversarial attacks is a notion of their generalization ability. To explore adversarial robustness of models trained using HGP and Hutchinson loss, we evaluate their performance against the Fast Gradient Sign Method (FSGM) attack (Goodfellow et al., 2014) and benchmark their performances against, ERM, IRM, V-Rex, and Fishr on CMNIST in Figure 5. We see that our Hutchinson method is also more transferable to adversarially perturbed domains.

5.7. Computational time comparison

For CMNIST experiments, we recorded the training time of our proposed algorithms (HGP and Hutchinson) and several popular domain generalization algorithms. The results of this experiment are presented in Table 5. We note that our Hessian alignment methods have more computation cost than existing algorithms, due to the Hessian estimation. This is the expense of better OOD generalization. However, the additional cost is controllable.
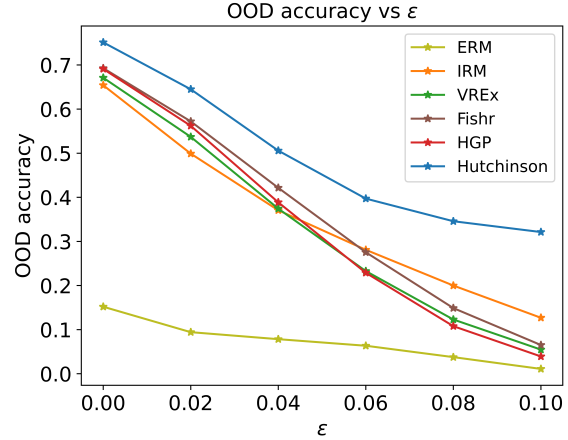

Figure 5: Comparison of OOD accuracy under adversarial attack. ε denotes the perturbation amplitude.

5.8. The roles of Hessian and gradient alignment

Since we have implemented both Hessian and gradient alignment in our algorithms, which part is playing a main role? In this section, we conduct an ablation study to investigate the role of each regularization term in the loss functions of Hessian alignment. Recall that α is the regularization weight of Hessian alignment and β is the weight of the gradient alignment. We set up an experiment with the PACS dataset and set the Sketch domain as test domain. We compare the Hutchinson algorithm in three different scenarios:

- Both: $\alpha = 10^{-5}, \beta = 10^{-5}$;

- Only Hessian alignment: $\alpha = 10^{-5}, \beta = 0$;

- Only gradient alignment: $\alpha = 0, \beta = 10^{-5}$.

The test domain accuracies for models trained with different hyperparameters are presented in Table 6. Clearly, with both Hessian and gradient alignment we achieve the best OOD generalization. Moreover, Hessian alignment plays a more important role than gradient alignment. Additional ablation studies for CMNIST experiment are presented in the appendix.

5.9. Implementation details

We provide the implementation details for all the experiments we conducted in the paper. For the transferability experiments in Figure 2, the experimental setup is exactly the same as the one used in Zhang et al. (2021). For regularization parameters, we used $\alpha = 10^{-5}$ and $\beta = 10^{-3}$ for HGP and Hutchinson methods. For the CMNIST experiment, the experimental setting is exactly the same as the IRM paper (Arjovsky et al., 2019). To set the regularization parameters, we also used the same scheme employed in

Table 4: DomainBed benchmark with leave-one-domain-out cross-validation model selection for CMNIST, VLCS, PACS, and OfficeHome datasets. We show the best and second best number with boldface and underline respectively. Due to time and computation limit, we did not finish the Fishr experiment on DomainNet.

Algorithm	VLCS	PACS	OfficeHome	DomainNet	Avg
ERM (Vapnik, 1999)	77.2	83.0	65.7	40.6	66.6
IRM (Arjovsky et al., 2019)	76.3	81.5	64.3	33.5	63.9
GroupDRO (Sagawa et al., 2020)	77.9	<u>83.5</u>	65.2	33.0	64.9
Mixup (Wang et al., 2020)	77.7	83.2	67.0	38.5	66.6
MLDG (Li et al., 2018a)	77.2	82.9	66.1	41.0	66.8
CORAL (Sun and Saenko, 2016)	<u>78.7</u>	82.6	**68.5**	<u>41.1</u>	<u>67.7</u>
MMD (Li et al., 2018b)	77.3	83.2	60.2	23.4	61.0
DANN (Ganin et al., 2016)	76.9	81.0	64.9	38.2	65.2
CDANN (Zhou et al., 2021)	77.5	78.8	64.3	38.0	64.6
MTL (Blanchard et al., 2021)	76.6	83.7	65.7	40.6	66.7
SagNet (Nam et al., 2020)	77.5	82.3	67.6	40.2	66.9
ARM (Zhang et al., 2020)	76.6	81.7	64.4	35.2	64.5
VREx (Krueger et al., 2021)	76.7	81.3	64.9	33.4	64.1
RSC (Huang et al., 2020)	77.5	82.6	65.8	38.9	66.2
Fishr (Rame et al., 2022)	78.2	**85.4**	<u>67.8</u>	-	-
HGP	76.7	82.2	67.5	<u>41.1</u>	66.9
Hutchinson	**79.3**	<u>84.8</u>	**68.5**	**41.4**	**68.5**

Table 5: Training time comparison on CMNIST between ERM, IRM, V-Rex, Fishr, and our proposed methods HGP and Hutchinson for 100 steps averaged over 10 runs.

Method	Train acc.
ERM	15.3 ± 0.1
CORAL	32.0 ± 3.2
V-REx	34.1 ± 2.3
IRM	27.0 ± 4.6
GroupDRO	32.8 ± 2.9
Fishr	25.3 ± 2.1
HGP	39.5 ± 4.4
Hutchinson	62.4 ± 1.5

Table 6: Ablation study of the Hutchinson method on different alignment regularization on the PACS dataset when the test domain is sketch.

Method	Test acc.
Hessian & gradient	81.4
Gradient only	77.0
Hessian only	79.4

the IRM paper (Arjovsky et al., 2019). The only difference here is that we have two regularization parameters so we set $\alpha = \beta$. For DomainBed experiments, the setup is the same as in Gulrajani and Lopez-Paz (2020). For α and β, we used ranges $(-3, -5)$, $(-1, -3)$ for HGP and $(-1, -3)$, $(-2, -4)$ for Hutchinson, respectively.

6. Conclusions

In this paper, we study the role Hessian alignment in domain generalization. Hessian alignment of the classifier head does not only improve the transferability to new domains, but also serves as an effective feature matching mechanism. In order to overcome the difficulty of heavy computation, we propose two estimation methods for Hessians, using Hessian-gradient product and Hutchinson's diagonal estimation. These methods are tested to be effective in various scenarios, including spurious correlation (CMNIST), standard diversity shift benchmark (Domainbed), and adversarial shift. At the expense of slightly heavier computation, Hessian alignment performs competitively, often achieving the state-of-the-art. To our knowledge, our method provides the first Hessian alignment in domain generalization. In the future, it would be interesting to study and compare more efficient ways to align Hessians and gradients across domains.

Acknowledgements

References

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

Bekas, C., Kokiopoulou, E., and Saad, Y. (2007). An estimator for the diagonal of a matrix. *Applied numerical mathematics*, 57(11-12):1214–1229.

Ben-David, S., Blitzer, J., Crammer, K., Kulesza, A., Pereira, F., and Vaughan, J. W. (2010). A theory of learning from different domains. *Machine learning*, 79(1):151–175.

Blanchard, G., Deshmukh, A. A., Dogan, Ü., Lee, G., and Scott, C. (2021). Domain generalization by marginal transfer learning. *The Journal of Machine Learning Research*, 22(1):46–100.

Chang, J.-R., Wu, M.-S., Yu, W.-H., Chen, C.-C., Yang, C.-K., Lin, Y.-Y., and Yeh, C.-Y. (2021). Stain Mix-Up: Unsupervised domain generalization for histopathology images. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 117–126. Springer.

Fang, C., Xu, Y., and Rockmore, D. N. (2013). Unbiased metric learning: On the utilization of multiple datasets and web images for softening bias. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1657–1664.

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

Goodfellow, I. J., Shlens, J., and Szegedy, C. (2014). Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*.

Gulrajani, I. and Lopez-Paz, D. (2020). In search of lost domain generalization. *arXiv preprint arXiv:2007.01434*.

Hu, W., Niu, G., Sato, I., and Sugiyama, M. (2018). Does distributionally robust supervised learning give robust classifiers? In *International Conference on Machine Learning*, pages 2029–2037. PMLR.

Huang, Z., Wang, H., Xing, E. P., and Huang, D. (2020). Self-challenging improves cross-domain generalization. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*, pages 124–140. Springer.

Ilse, M., Tomczak, J. M., and Forré, P. (2021). Selecting data augmentation for simulating interventions. In *International Conference on Machine Learning*, pages 4555–4562. PMLR.

Jaakkola, T. and Haussler, D. (1998). Exploiting generative models in discriminative classifiers. *Advances in Neural Information Processing Systems*, 11.

Koltchinskii, V. (2010). Rademacher complexities and bounding the excess risk in active learning. *The Journal of Machine Learning Research*, 11:2457–2485.

Koyama, M. and Yamaguchi, S. (2021). When is invariance useful in an out-of-distribution generalization problem?

Krueger, D., Caballero, E., Jacobsen, J.-H., Zhang, A., Binas, J., Zhang, D., Le Priol, R., and Courville, A. (2021). Out-of-distribution generalization via risk extrapolation (REx). In *International Conference on Machine Learning*, pages 5815–5826. PMLR.

Li, D., Yang, Y., Song, Y.-Z., and Hospedales, T. (2018a). Learning to generalize: Meta-learning for domain generalization. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32.

Li, D., Yang, Y., Song, Y.-Z., and Hospedales, T. M. (2017). Deeper, broader and artier domain generalization. In *Proceedings of the IEEE international conference on computer vision*, pages 5542–5550.

Li, H., Pan, S. J., Wang, S., and Kot, A. C. (2018b). Domain generalization with adversarial feature learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5400–5409.

Mansilla, L., Echeveste, R., Milone, D. H., and Ferrante, E. (2021). Domain generalization via gradient surgery. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 6630–6638.

Muandet, K., Balduzzi, D., and Schölkopf, B. (2013). Domain generalization via invariant feature representation. In *International Conference on Machine Learning*, pages 10–18. PMLR.

Nam, J., Cha, H., Ahn, S., Lee, J., and Shin, J. (2020). Learning from failure: De-biasing classifier from biased classifier. *Advances in Neural Information Processing Systems*, 33:20673–20684.

Parascandolo, G., Neitz, A., Orvieto, A., Gresele, L., and Schölkopf, B. (2020). Learning explanations that are hard to vary. *arXiv preprint arXiv:2009.00329*.

Peng, X., Bai, Q., Xia, X., Huang, Z., Saenko, K., and Wang, B. (2019). Moment matching for multi-source domain adaptation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1406–1415.

Rame, A., Dancette, C., and Cord, M. (2022). Fishr: Invariant gradient variances for out-of-distribution generalization. In *International Conference on Machine Learning*, pages 18347–18377. PMLR.

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

Shahtalebi, S., Gagnon-Audet, J.-C., Laleh, T., Faramarzi, M., Ahuja, K., and Rish, I. (2021). SAND-mask: An enhanced gradient masking strategy for the discovery of invariances in domain generalization. *arXiv preprint arXiv:2106.02266*.

Shi, Y., Seely, J., Torr, P. H., Siddharth, N., Hannun, A., Usunier, N., and Synnaeve, G. (2021). Gradient matching for domain generalization. *arXiv preprint arXiv:2104.09937*.

Skorski, M. (2021). Modern analysis of Hutchinson's trace estimator. In *2021 55th Annual Conference on Information Sciences and Systems (CISS)*, pages 1–5. IEEE.

Sun, B. and Saenko, K. (2016). Deep CORAL: Correlation alignment for deep domain adaptation. In *European conference on computer vision*, pages 443–450. Springer.

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

Vapnik, V. N. (1999). An overview of statistical learning theory. *IEEE Transactions on Neural Networks*, 10(5):988–999.

Venkateswara, H., Eusebio, J., Chakraborty, S., and Panchanathan, S. (2017). Deep hashing network for unsupervised domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5018–5027.

Wang, J., Lan, C., Liu, C., Ouyang, Y., Qin, T., Lu, W., Chen, Y., Zeng, W., and Yu, P. (2022). Generalizing to unseen domains: A survey on domain generalization. *IEEE Transactions on Knowledge and Data Engineering*.

Wang, Y., Li, H., and Kot, A. C. (2020). Heterogeneous domain generalization via domain mixup. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3622–3626. IEEE.

Ye, N., Li, K., Hong, L., Bai, H., Chen, Y., Zhou, F., and Li, Z. (2021). OoD-Bench: Benchmarking and understanding out-of-distribution generalization datasets and algorithms. *arXiv preprint arXiv:2106.03721*, 1(3):5.

Zhang, G., Zhao, H., Yu, Y., and Poupart, P. (2021). Quantifying and improving transferability in domain generalization. *Advances in Neural Information Processing Systems*, 34:10957–10970.

Zhang, H., Cisse, M., Dauphin, Y. N., and Lopez-Paz, D. (2018). mixup: Beyond empirical risk minimization. In *International Conference on Learning Representations*.

Zhang, M., Marklund, H., Gupta, A., Levine, S., and Finn, C. (2020). Adaptive risk minimization: A meta-learning approach for tackling group shift. *arXiv preprint arXiv:2007.02931*, 8:9.

Zhao, H., Des Combes, R. T., Zhang, K., and Gordon, G. (2019). On learning invariant representations for domain adaptation. In *International conference on machine learning*, pages 7523–7532. PMLR.

Zhao, H., Zhang, S., Wu, G., Moura, J. M., Costeira, J. P., and Gordon, G. J. (2018). Adversarial multiple source domain adaptation. *Advances in neural information processing systems*, 31.

Zhou, K., Liu, Z., Qiao, Y., Xiang, T., and Loy, C. C. (2021). Domain generalization in vision: A survey. *arXiv preprint arXiv:2103.02503*.