Gradient Matching for Domain Generalization

ICLR 2022 (Poster)

Motivation

- ML systems typically assume that the distributions of training and test sets match closely.
- A critical requirement of such systems in the real world is their ability to generalize to unseen domains.
    - Examples: Amazon reviews of different reviewers, Speech detection of different people.

- This seemingly difficult task is made possible by the presence of multiple distributions/domains at train time.

Challenge

- If we simply try to minimize the **avg.** loss across different domain, , the classifier is prone to spuriously correlate features with the specific domain.
    - “cow” with grass and “camels” with desert, and predict the species using background.
- We want the model to recognize that while the landscapes change, the biological characteristics of the animals remain **invariant**.
- Using those **invariant** features to determine the species, we have a much better chance at generalizing to unseen domains.

Contributions

- This paper proposes an inter-domain gradient matching (IDGM) objective.
    - interested in learning a model with invariant gradient direction for different domains.
- IDGM objective augments the loss with an auxiliary term that maximizes the gradient inner product between domains, which encourages the alignment between the domain-specific gradients.

Prior Works

ERM: Empirical Risk Minimisation

$$\mathcal{L}(g_i) = \frac{1}{N_{g_i}} \sum_{j=1}^{N_{g_i}} \mathcal{L}(x_j)$$

IRM: Invariant Risk Minimization

$$\mathcal{L}_{IRM} = \frac{1}{G}\left(\sum_{i=1}^{G} \mathcal{L}(g_i) + \lambda * P(g_i)\right) \qquad (10)$$

where $\mathcal{L}_{gi}$ is the loss of the i_{th} instance, which is part of the g_{th} group (label). Refer to Arjovsky et al. (2020) for a more detailed introduction of the group penalty terms (P_g).

IDGM Objective

$\mathcal{D}_{tr} = \{\mathcal{D}_1, \mathcal{D}_2\}$. Given model θ and loss function l, the expected gradients for data in the two domains is expressed as

$$G_1 = \mathbb{E}_{\mathcal{D}_1} \frac{\partial l((x, y); \theta)}{\partial \theta}, \quad G_2 = \mathbb{E}_{\mathcal{D}_2} \frac{\partial l((x, y); \theta)}{\partial \theta}. \tag{3}$$

$$\mathcal{L}_{\text{idgm}} = \mathcal{L}_{\text{erm}}(\mathcal{D}_{tr}; \theta) - \gamma \underbrace{\frac{2}{S(S-1)} \sum_{i,j \in S}^{i \neq j} G_i \cdot G_j}_{\text{GIP, denote as } \widehat{G}},$$

ERM vs IDGM Objective

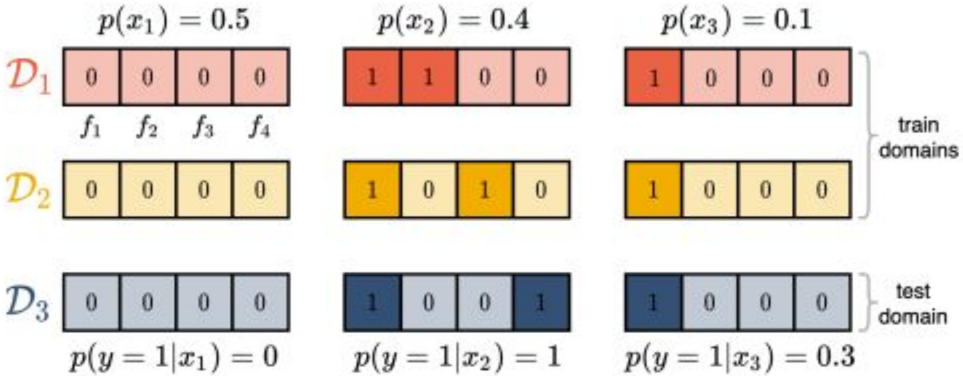

Figure 2: All domains contain 3 types of inputs x_1, x_2 and x_3, each depicted in one column. ***1**st **col.***: $x_1 = [0, 0, 0, 0]$, $y = 0$, makes up for 50% of each dataset; ***2**nd **col.***: x_2 changes for each domain, $y = 1$ always. 40% of each dataset; ***3**rd **col.***: $x_3 = [1, 0, 0, 0]$, 30% of $y = 1$ and 70% of $y = 0$. 10% of each dataset.

ERM vs IDGM Objective

Table 1: Performance comparison on the linear dataset.

Method	train acc.	test acc.	W	b
ERM	97%	57%	$[2.8, 3.3, 3.3, 0.0]$	-2.7
IDGM	93%	93%	$[0.4, 0.2, 0.2, 0.0]$	-0.4
Fish	93%	93%	$[0.4, 0.2, 0.2, 0.0]$	-0.4

ERM vs IDGM Objective

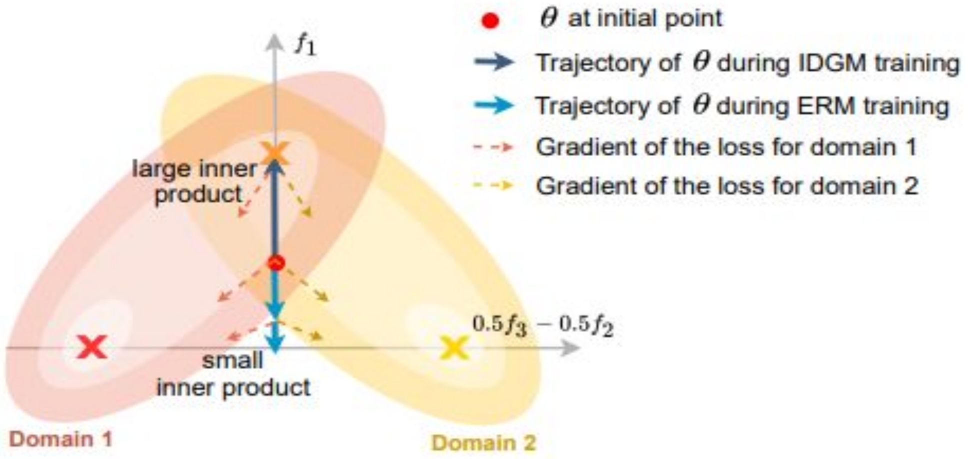

Figure 1: Isometric projection of training with ERM (blue) vs. our IDGM objective (dark blue), using data from Figure 2.

Limitations of IDGM

- 2nd-order derivative.
- Computationally expensive.

$$\mathcal{L}_{\text{idgm}} = \mathcal{L}_{\text{erm}}(\mathcal{D}_{tr};\theta) - \gamma \underbrace{\frac{2}{S(S-1)} \sum_{i,j \in S}^{i \neq j} G_i \cdot G_j}_{\text{GIP, denote as } \widehat{G}},$$

FISH

Algorithm 1 Fish.

for iterations = $1, 2, \cdots$ **do**
  $\widetilde{\theta} \leftarrow \theta$
  for $\mathcal{D}_i \in \texttt{permute}(\{\mathcal{D}_1, \mathcal{D}_2, \cdots, \mathcal{D}_S\})$ **do**
    Sample batch $d_i \sim \mathcal{D}_i$
    $\widetilde{g}_i = \mathbb{E}_{d_i}\left[\frac{\partial l((x, y); \widetilde{\theta})}{\partial \widetilde{\theta}}\right]$ //`Grad wrt` $\widetilde{\theta}$
    Update $\widetilde{\theta} \leftarrow \widetilde{\theta} - \alpha \widetilde{g}_i$
  end for
  Update $\theta \leftarrow \theta + \epsilon(\widetilde{\theta} - \theta)$
end for

Algorithm 2 Direct optimization of IDGM.

for iterations = $1, 2, \cdots$ **do**
  $\widetilde{\theta} \leftarrow \theta$
  for $\mathcal{D}_i \in \texttt{permute}(\{\mathcal{D}_1, \mathcal{D}_2, \cdots, \mathcal{D}_S\})$ **do**
    Sample batch $d_i \sim \mathcal{D}_i$
    $g_i = \mathbb{E}_{d_i}\left[\frac{\partial l((x, y); \theta)}{\partial \theta}\right]$ //`Grad wrt` θ
  end for
  $\bar{g} = \frac{1}{S} \sum_{s=1}^{S} g_s, \quad \widehat{g} = \overbrace{\frac{2}{S(S-1)} \sum_{i,j \in S}^{i \neq j} g_i \cdot g_j}^{\text{GIP (batch)}}$
  Update $\theta \leftarrow \theta - \epsilon\left(\bar{g} - \gamma(\partial \widehat{g} / \partial \theta)\right)$
end for

FISH

- Simplified IDGM

Theorem 3.1 *Given twice-differentiable model with parameters θ and objective l. Let us define the following:*

$$G_f = \mathbb{E}[(\theta - \widetilde{\theta})] - \alpha S \cdot \bar{G}, \qquad \textit{Fish update - } \alpha S \cdot \textit{ERM grad}$$

$$G_g = -\partial \widehat{G} / \partial \theta, \qquad \textit{grad of } \max_{\theta}(\widehat{G})$$

where $\bar{G} = \frac{1}{S} \sum_{s=1}^{S} G_s$ and is the full gradient of ERM. Then we have

$$\lim_{\alpha \to 0} \frac{G_f \cdot G_g}{\|G_f\| \cdot \|G_g\|} = 1.$$

Datasets

- **WILDS:** Multiple real-world distribution shift.

Table 6: Details of the 6 WILDS datasets we experimented on.

Dataset	Domains (# domains)	Data (x)	Target (y)	# Examples			# Domains		
				train	val	test	train	val	test
FMoW	Time (16), Regions (5)	Satellite images	Land use (62 classes)	76,863	19,915	22,108	11, -	3, -	2, -
POVERTY	Countries (23), Urban/rural (2)	Satellite images	Asset (real valued)	10,000	4,000	4,000	13, -	5, -	5, -
CAMELYON17	Hospitals (5)	Tissue slides	Tumor (2 classes)	302,436	34,904	85,054	3	1	1
CIVILCOMMENTS	Demographics (8)	Online comments	Toxicity (2 classes)	269,038	45,180	133,782	-	-	-
IWILDCAM2020	Trap locations (324)	Photos	Animal species (186 classes)	142,202	20,784	38,943	245	32	47
AMAZON	Reviewers (7,676)	Product reviews	Star rating (5 classes)	1,000,124	100,050	100,050	5,008	1,334	1,334

Datasets

- **DomainBed**: 7 domain generalization
  - Colored MNIST
  - Rotated MNIST
  - ...

Results

- **WILDS:**

Table 3: Results on WILDS benchmark.

	PovertyMap	Camelyon17	FMoW	CivilComments	iWildCam	Amazon
	Pearson r	Avg. acc. (%)	Worst acc. (%)	Worst acc. (%)	Macro F1	10-th per. acc. (%)
Fish	**0.80** **(±*1e-2*)**	**74.7** (±*7e-2*)	**34.6** **(±*0.00*)**	**72.8** **(±*0.0*)**	22.0 (±*0.0*)	**53.3** **(±*0.0*)**
IRM	0.78 (±*3e-2*)	64.2 (±*8.1*)	33.5 (±*1.35*)	66.3 (±*2.1*)	15.1 (±*4.9*)	52.4 (±*0.8*)
Coral	0.77 (±*5e-2*)	59.5 (±*7.7*)	31.0 (±*0.35*)	65.6 (±*1.3*)	**32.8**(±*0.1*)	52.9 (±*0.8*)
Reweighted	-	-	-	66.2 (±*1.2*)	-	52.4 (±*0.8*)
GroupDRO	0.78 (±*5e-2*)	68.4 (±*7.3*)	31.4 (±*2.10*)	69.1 (±*1.8*)	23.9 (±*2.1*)	53.5 (±*0.0*)
ERM	0.78 (±*3e-2*)	70.3 (±*6.4*)	32.8 (±*0.45*)	56.0 (±*3.6*)	31.0 (±*1.3*)	**53.8** **(±*0.8*)**
ERM (ours)	0.77 (±*5e-2*)	70.5 (±*12.1*)	30.9 (±*1.53*)	58.1 (±*1.7*)	25.1 (±*0.2*)	**53.3** **(±*0.8*)**

Results

- **DomainBed:**

Table 4: Test accuracy (%) on DOMAINBED benchmark.

	ERM	IRM	GroupDRO	Mixup	MLDG	Coral	MMD	DANN	CDANN	Fish (ours)
CMNIST	52.0 (±0.1)	51.8 (±0.1)	52.0 (±0.1)	51.9 (±0.1)	51.6 (±0.1)	51.7 (±0.1)	51.8 (±0.1)	51.5 (±0.3)	51.9 (±0.1)	51.6 (±0.1)
RMNIST	98.0 (±0.0)	97.9 (±0.0)	98.1 (±0.0)	98.1 (±0.0)	98.0 (±0.0)	98.1 (±0.1)	98.1 (±0.0)	97.9 (±0.1)	98.0 (±0.0)	98.0 (±0.0)
VLCS	77.4 (±0.3)	78.1 (±0.0)	77.2 (±0.6)	77.7 (±0.4)	77.1 (±0.4)	77.7 (±0.5)	76.7 (±0.9)	78.7 (±0.3)	78.2 (±0.4)	77.8 (±0.3)
PACS	85.7 (±0.5)	84.4 (±1.1)	84.1 (±0.4)	84.3 (±0.5)	84.8 (±0.6)	86.0 (±0.2)	85.0 (±0.2)	84.6 (±1.1)	82.8 (±1.5)	85.5 (±0.3)
OfficeHome	67.5 (±0.5)	66.6 (±1.0)	66.9 (±0.3)	69.0 (±0.1)	68.2 (±0.1)	68.6 (±0.4)	67.7 (±0.1)	65.4 (±0.6)	65.6 (±0.5)	68.6 (±0.4)
TerraInc	47.2 (±0.4)	47.9 (±0.7)	47.0 (±0.3)	48.9 (±0.8)	46.1 (±0.8)	46.4 (±0.8)	49.3 (±1.4)	48.7 (±0.5)	47.6 (±0.8)	45.1 (±1.3)
DomainNet	41.2 (±0.2)	35.7 (±1.9)	33.7 (±0.2)	39.6 (±0.1)	41.8 (±0.4)	41.8 (±0.2)	39.4 (±0.8)	38.4 (±0.0)	38.9 (±0.1)	42.7 (±0.2)
Average	67.0	66.0	65.5	67.1	66.8	67.2	66.8	66.4	66.1	67.1

Ablation Study

- **Random Grouping:**

Table 5: Ablation study on random grouping: test accuracy on different datasets.

	CDSPRITES(N=10)	FMoW	VLCS	PACS	OfficeHome
Fish	100.0 (±0.0)	34.3 (±0.6)	77.6 (±0.5)	85.5 (±0.3)	68.6 (±0.9)
Fish, RG	50.0 (±0.0)	33.4 (±1.7)	77.7 (±0.3)	83.9 (±0.7)	66.5 (±1.0)
ERM	50.0 (±0.0)	31.7 (±1.0)	77.5 (±0.4)	85.5 (±0.2)	66.5 (±0.3)

Ablation Study

- **Hyperparameters:**

Dataset	Model	Learning rate	Batch size	Weight decay	Optimizer	Val. metric	Cut-off
CAMELYON17	Densenet-121	1e-3	32	0	SGD	acc. avg.	iter 500
CIVILCOMMENTS	BERT	1e-5	16	0.01	Adam	acc. wg.	Best val. metric
FMoW	Densenet-121	1e-4	64	0	Adam	acc. avg.	Best val. metric
IWILDCAM	Resnet-50	1e-4	16	0	Adam	F1-macro (all)	Best val. metric
POVERTY	Resnet-18	1e-3	64	0	Adam	Pearson (r)	-
AMAZON	BERT	2e-6	8	0.01	Adam	10th percentile acc.	-

In Table 14 we list out the hyperparameters we used to train Fish. Note that we train Fish using the same model, batch size, val metric and optimizer as ERM – these are not listed in Table 14 to avoid repetitions. Weight decay is always set as 0.

Table 14: Hyperparameters for Fish.

Dataset	Group by	α	ϵ	# domains	Meta steps
CAMELYON17	Hospitals	1e-3	0.01	3	3
CIVILCOMMENTS	Demographics $\times$ toxicity	1e-5	0.05	16	5
FMoW	time $\times$ regions	1e-4	0.01	80	5
IWILDCAM	Trap locations	1e-4	0.01	324	10
POVERTY	Countries	1e-3	0.1	23	5
AMAZON	Reviewers	2e-6	0.01	7,676	5

Ablation Study

- **Convergence of Pre-trained ERM:**

Table 15: Ablation study on pretrained ERM models.

Model	FMoW	CAMELYON17	IWILDCAM	CIVILCOMMENTS
	Test Avg Acc	Test Avg Acc	Test Macro F1	Test Worst Acc
10% data	21.7 (±2.5)	79.1 (±12.3)	13.7 (±0.5)	71.8 (±1.3)
50% data	31.0 (±0.8)	64.6 (±12.3)	19.0 (±0.06)	74.2 (±0.5)
Converged	32.7 (±1.2)	63.5 (±8.2)	23.7 (±0.9)	73.8 (±1.8)

Ablation Study

- **Tracking gradient inner product:**

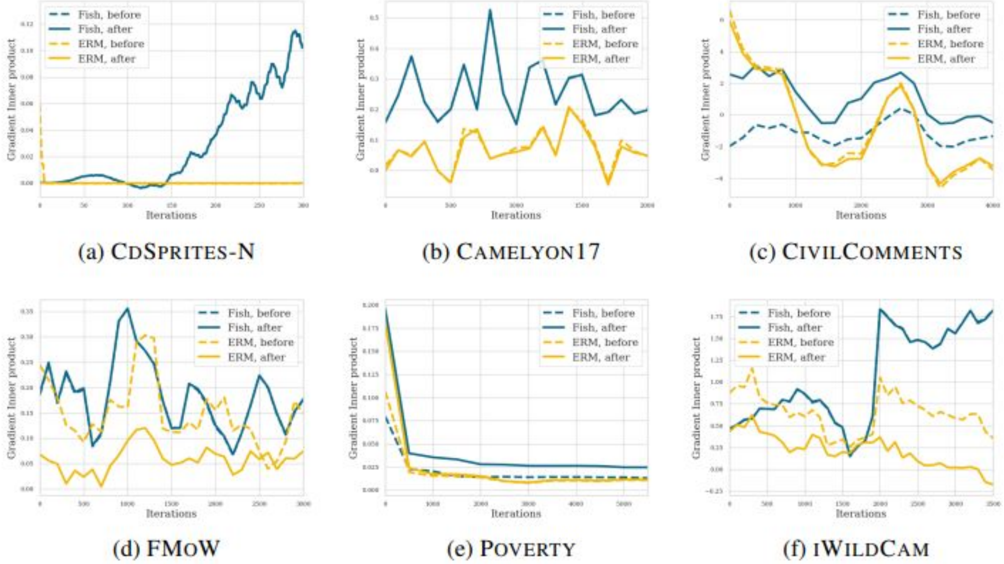

Figure 9: Gradient inner product values during the training for CDSPRITES-N (N=15) and 5 different WILDS datasets.