# Empirical Study on Optimizer Selection for Out-of-Distribution Generalization

Hiroki Naganuma*[1,2], Kartik Ahuja[1,2], Shiro Takagi†, Tetsuya Motokawa[4], Rio Yokota[5], Kohta Ishikawa[6], Ikuro Sato[5,6], Ioannis Mitliagkas[1,2,3]

*: naganuma.hiroki@mila.quebec    †: Independent Researcher

[1] Mila [2] Université de Montréal [3] CIFAR [4] 筑波大学 University of Tsukuba [5] 東京工業大学 Tokyo Institute of Technology [6] DENSO IT LAB

TRANSACTIONS on ML RESEARCH

## Introduction

【Motivation】

- **Optimizer selection**
  - Crucial for the successful training of DNNs.
  - Influences training speed, stability, and generalization performance.
  - Previous studies of are based on a IID assumption
- **Out-of-distribution (OOD) generalization**
  - In real-world applications, it is often the case that the test data obey a distribution different from the training data
  - Distributional shift violates the typical IID assumption for training
  - Comparing the OOD generalization performance among different optimizers is of great interest in theory and in practice

【Contribution】

- **Design and perform a comparison of the effect of optimizers on OOD generalization on OOD benchmarks**
  - Evaluate 10 out-of-distribution generalization datasets (including image classification and NLP)
  - Wide range of hyperparameter configurations (examining over 20,000 models)
- **Demonstrate optimizer characteristic under distributional shift**
  - The adaptive optimizers provide more in-distribution (ID) overfitting and degrade OOD performance more than the non-adaptive optimizers
  - Non-adaptive optimizer outperformed adaptive optimizer in terms of best OOD accuracy (8 out of 10 datasets)
- **Observed correlation behaviors: ID vs OOD performance**
  - It can be categorized into typical patterns:linear return, diminishing return, and increasing return

## Limitation of IID Assumption

Empirical risk minimization (ERM) as known for standard training method could achieve high ID performance by learning spurious correlations.

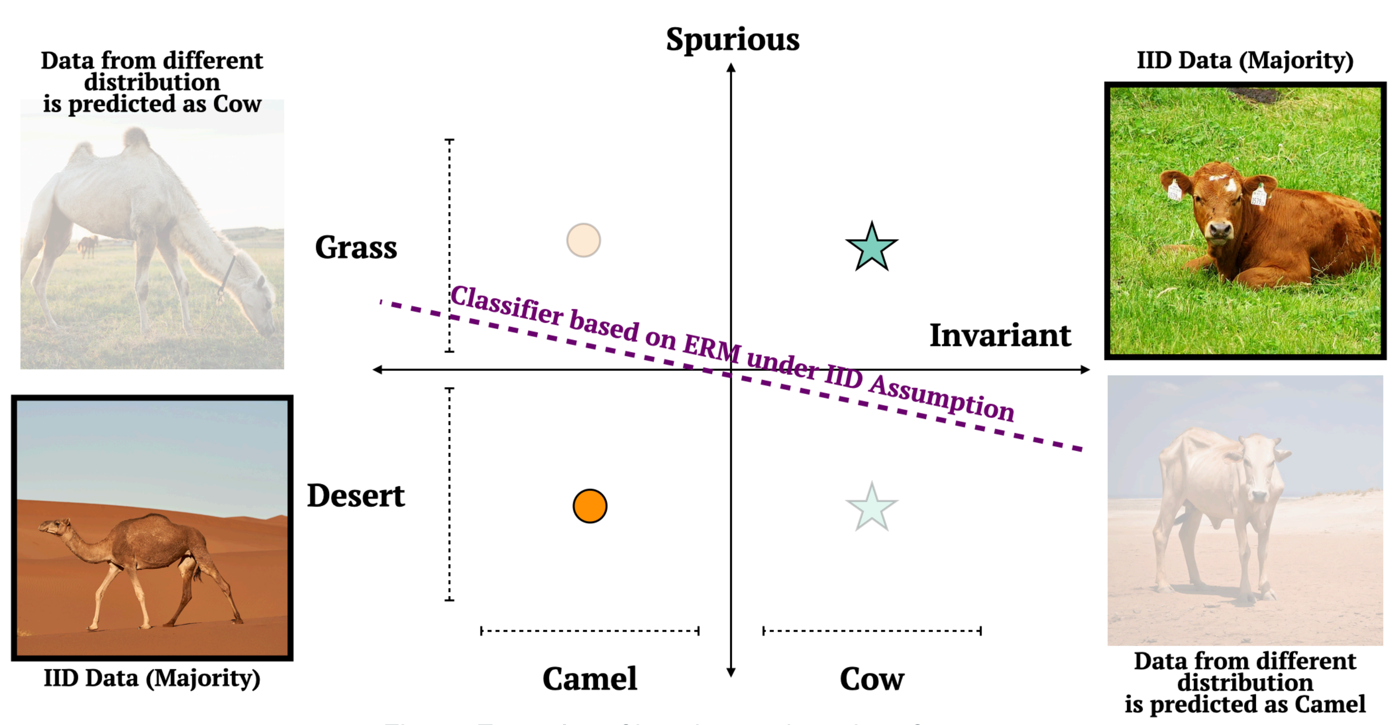


Figure: Examples of invariant and spurious features.

## Why Optimizer Selection?

- Learning method to mitigate the mentioned above is also studied
  - Invariant risk minimization (IRM) [Arjovsky19] is also conducted in our study
- However, these methods have not provided sufficient OOD performance, and the influence of the optimizer has not been taken into account so far
- Adam, due to its update formula, is likely to capture noise that is not an invariant feature, although it converges quickly

## Optimizers Subjected in Our Analysis

We target five of the most popular and standard optimizers that have been used and studied in recent years

【Non-Adaptive Optimizers】

In addtition to SGD, optimizers with momentum terms such as Momentum SGD, and Nesterov momentum are also classified as non-adaptive optimizers

$$\boldsymbol{v}_t \leftarrow \gamma \boldsymbol{v}_{t-1} + \eta_t \tilde{\nabla}_{\boldsymbol{\theta}_{t-1}} \ell(\boldsymbol{\theta}_{t-1}), \ \ \boldsymbol{\theta}_t \leftarrow \boldsymbol{\theta}_{t-1} - \boldsymbol{v}_t$$

*where $\theta_t$ is model parameter, $\eta_t$ is learning rate, $\ell(\theta)$ is loss $\tilde{\nabla}_{\theta_{t-1}}$ is stochastic gradient and $\gamma$ is momentum.*

【Adaptive Optimizers】

Adam and RMSprop are adaptive optimizers and they can be written in the form of the generic adaptive optimization method

**Algorithm 1** Generic adaptive optimization method setup.

**Require:** $\{\eta_t\}_{t=1}^{T}$: step size, $\{\phi_t, \psi_t\}_{t=1}^{T}$ function to calculate momentum and adaptive rate, $\boldsymbol{\theta}_0$: initial parameter, $\ell(\boldsymbol{\theta})$: objective function

```
1: for t = 1 to T do
2:     g_t ← ∇̃_θ f_t(θ_{t−1}) (Calculate stochastic gradients w.r.t. objective at timestep t)
3:     w_t ← φ_t(g_1, ..., g_t) (Calculate momentum)
4:     l_t ← ψ_t(g_1, ..., g_t) (Calculate adaptive learning rate)
5:     θ_t ← θ_{t−1} − η_t w_t l_t (Update parameters)
6: end for
```

## Experimental Protocol

【Datasets】

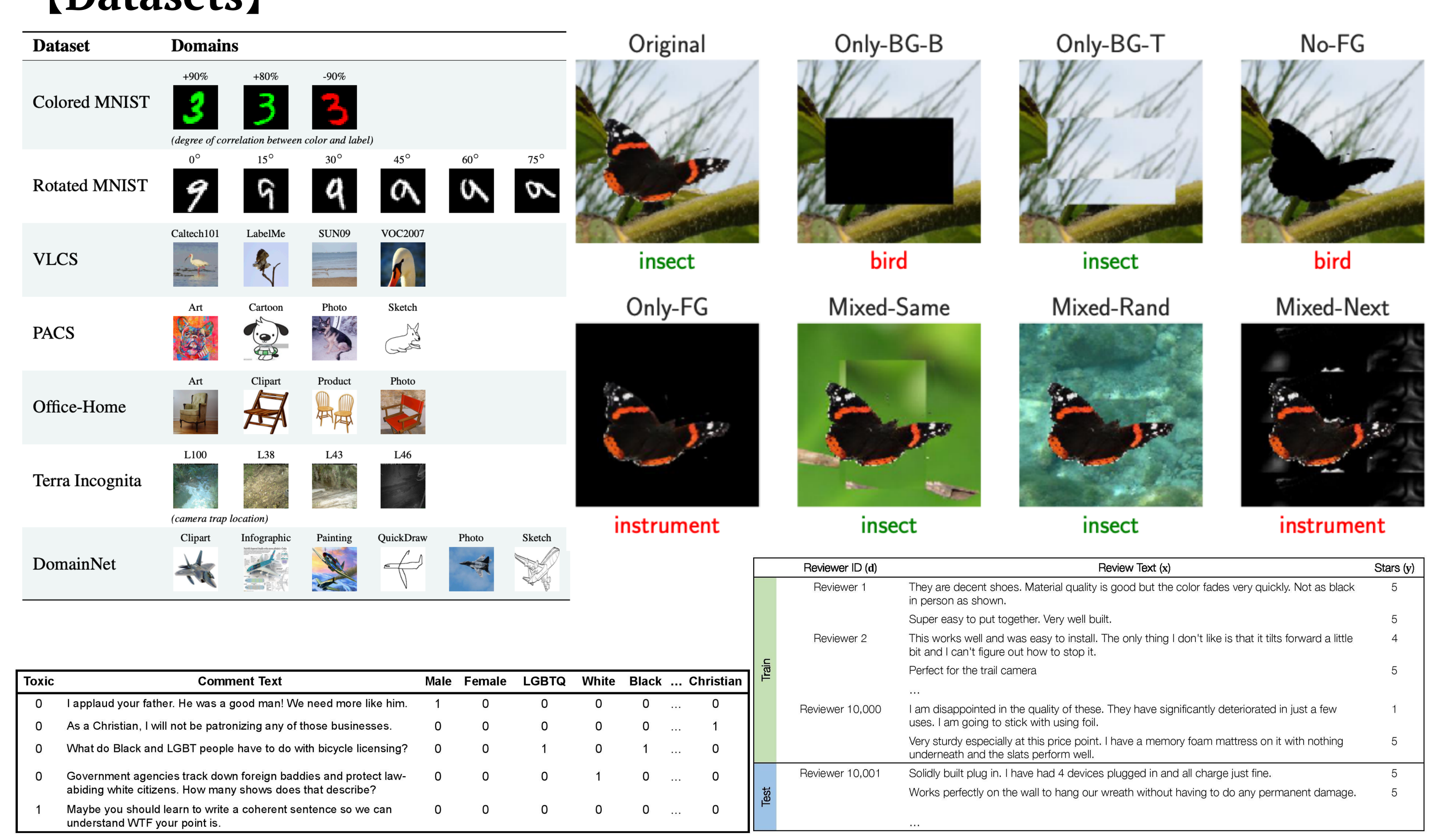


Figure: OOD Datasets we evaluate in our study (Image taken from [Gulrajani21](**Domainbed** / left), [Xiao21](**Background Challenge** / right), and [Koh2021] (**WILDS** / bottom))

【Model Selection Method and Evaluation Metrics】

- We follow the benchmark respectively [Gulrajani21],[Xiao21] and [Koh2021]
- For the image classification tasks
  - The training domain is split into training and validation data
  - OOD performance is evaluated in the test domain
- For the NLP tasks, the worst group is evaluated as the OOD performance

【Hyperparameter Tuning】

- The exhaustiveness of the hyperparameter search is crucial for empirical investigation of an optimizer's effect
- We basically follow [Choi19], which most exhaustively searched hyperparameters for optimizer comparison and explored more hyperparameters than did previous studies

## Optimizer Comparison in OOD Accuracy

We compared Momentum SGD as the best non-adaptive optimizer, with Adam as the best adaptive optimizer

【Experimental results and implication】

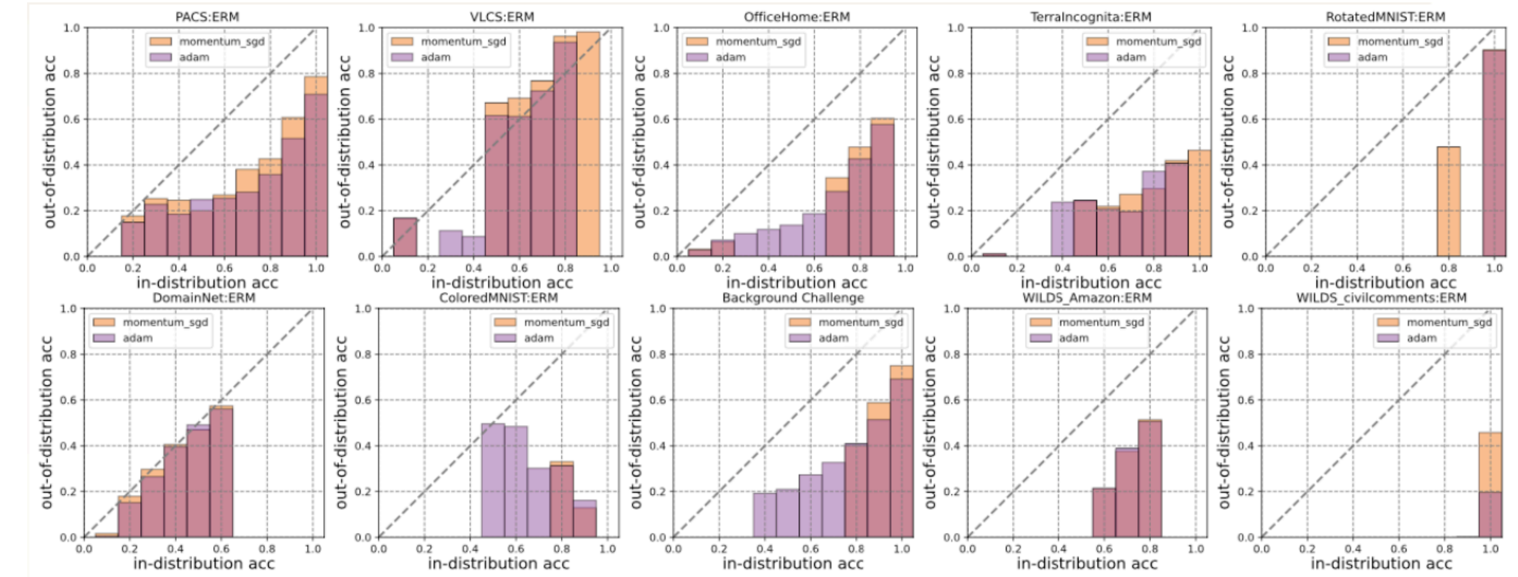


Figure: Relationship between the ID accuracy and the OOD accuracy in the ERM setting.
The x-axis of the plot is the in-distribution accuracy and the y-axis is the OOD accuracy. To make the trend more clear, the in-distribution accuracy corresponding to the x-axis is divided into 10 bins, and the **average performance of the OOD accuracy** in each bin is shown on the y-axis.accuracy in each bin is shown on the y-axis.

- In our area of interest, where a high in-distribution performance is achieved, **Momentum SGD outperforms Adam on 9 of the 10 datasets in the sense of average OOD accuracy (Figure)**
- This indicates that non adaptive optimizer is more advantageous than adaptive optimizer in OOD, even though the performance is similar in the IID environment

| Model | OOD Dataset | Non-Adaptive Optimizer | | | Adaptive Optimizer | |
|---|---|---|---|---|---|---|
| | | SGD | Momentum | Netsterov | RMSProp | Adam |
| 4-Layer CNN | ColoredMNIST | 34.01% | 34.23% | 40.56% | **89.30%** | 73.92% |
| | RotatedMNIST | 90.00% | 95.41% | 94.06% | 96.27% | **96.40%** |
| ResNet50 | VLCS | 99.43% | **99.43%** | 99.29% | 99.36% | 99.36% |
| | PACS | 88.67% | **89.55%** | 89.25% | 88.81% | 89.30% |
| | OfficeHome | 64.64% | **65.01%** | 63.82% | 62.91% | 63.12% |
| | TerraIncognita | **63.21%** | 62.41% | 62.85% | 62.31% | 61.35% |
| | DomainNet | 58.38% | 61.91% | **62.24%** | 55.74% | 58.48% |
| | BackgroundChallenge | - | **80.09%** | - | - | 77.90% |
| DistilBERT | WILDSAmazon | 52.00% | **54.66%** | 54.66% | 53.33% | 51.99% |
| | WILDSCivilComment | 51.66% | 57.69% | **60.07%** | 45.39% | 46.82% |

Table: Comparison of the best OOD accuracy of ERM between five optimizers.
Except for a small set of problems, momentum SGD outperforms Adam. As a soundness check, we confirm that our Adam results outperform all existing benchmark results using Adam.

- When comparing the performance of the best OOD accuracy, the non-adaptive optimisers outperformed the adaptive optimizers in 8 out of 10 data sets (Table)

## Correlation Behaviour (IID vs OOD)

Our results show that three typical types of behavior are observed in terms of the correlation between in-distribution performance and OOD performance for different datasets. These show how much performance in OOD can be expected if we increase the in-distribution performance.

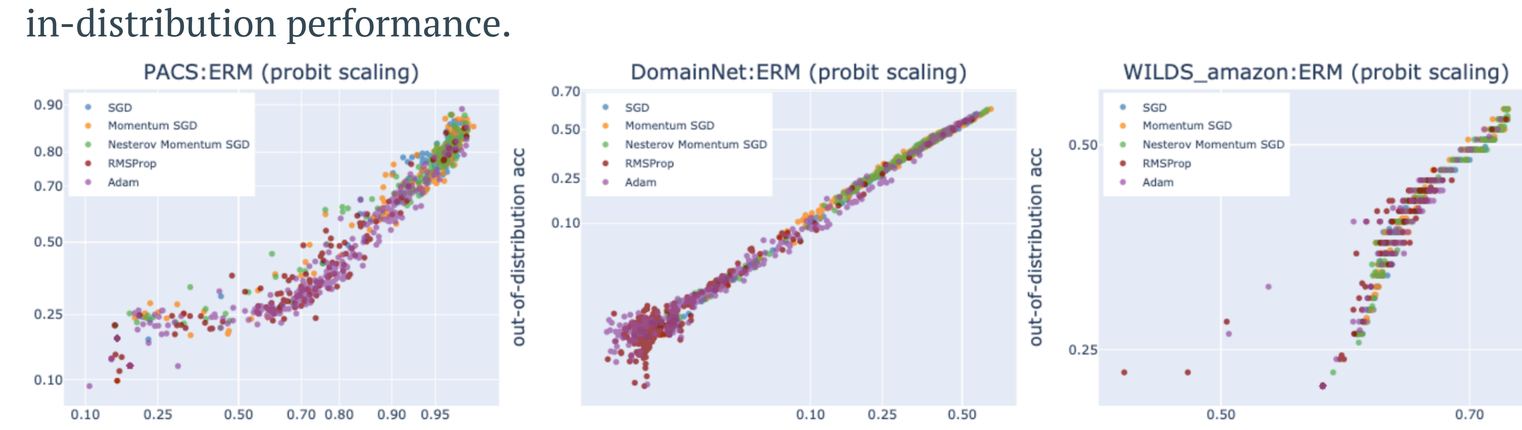


Figure: Three-types of correlation Behaviour:
increasing return (PACS), linear return (DomainNet), and diminishing return (Amazon-WILDS).

## References

[Arjovsky19] Martin Arjovsky et al. "Invariant risk minimization". In: arXiv preprint arXiv:1907.02893 (2019)

[Gulrajani21] Ishaan Gulrajani and David Lopez-Paz. "In Search of Lost Domain Generalization". In: International Conference on Learning Representations. 2021

[Xiao21] Kai Yuanqing Xiao et al. "Noise or Signal: The Role of Image Backgrounds in Object Recognition". In: *International Conference on Learning Representations*. 2021.

[Koh2021] Pang Wei Koh et al. "Wilds: A benchmark of in-the-wild distribution shifts". In: *Interna- tional Conference on Machine Learning*. PMLR. 2021, pp. 5637–5664.