NEURAL INFORMATION PROCESSING SYSTEMS

# Improving Out-of-Distribution Generalization by Adversarial Traing with Structured Priors

Qixun Wang*[1], Yifei Wang*[1], Hong Zhu[2], Yisen Wang[1]

[1]Peking University [2]Huawei Noah’s Ark Lab

Neural Information Processing Systems (NeurIPS 2022)

## Out-of-distribution (OOD) Generalization:

NEURAL INFORMATION PROCESSING SYSTEMS

- Train on $m$ training domains $\mathcal{E} = \{E_1, E_2, \ldots, E_m\}$, $E_e \sim \mathcal{P}_e$
- Test domain $E_{m+1}$, $E_{m+1} \sim \mathcal{P}_{te}$
- $\mathcal{P}_{te} \neq \mathcal{P}_i$

Object: $\min_f \mathbb{E}_{(x,y)\sim\mathcal{P}_{te}(x,y)}[\mathcal{L}(f(x), y)]$

Solve

## Adversarial Training (AT):

Optimization problem:

$$\min_f \mathbb{E}_{(x,y)\sim\mathcal{P}(x,y)}\left[\max_{\delta\in\mathcal{S}} \mathcal{L}(f(x+\delta), y)\right] \text{ s.t. } \|\delta\|_p \leq \epsilon,$$

Inner maximization can be solved by:

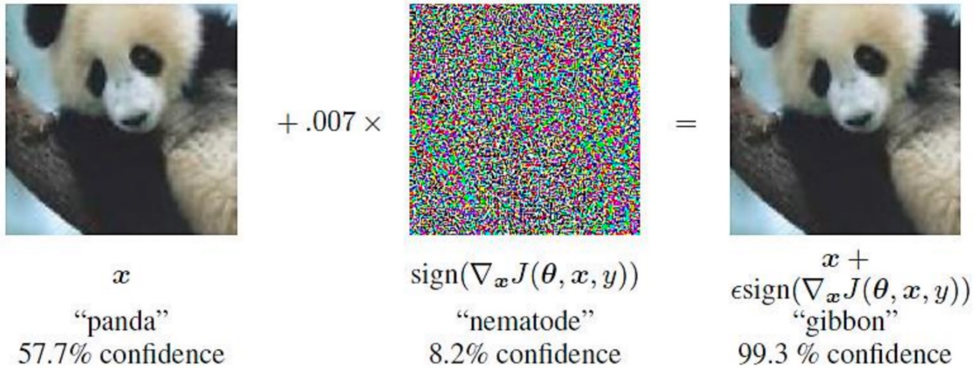
FGSM $x = x + \epsilon \operatorname{sgn}(\nabla_x \mathcal{L}(f(x), y))$

or PGD $x^{t+1} = \prod_{\mathcal{S}}(x^t + \gamma \operatorname{sgn}(\nabla_x \mathcal{L}(f(x), y)))$

originally for defending

**Adversarial Attack**

Ian Goodfellow et al., 2014



# Previous work of using AT to address OOD

**[Yi et al, 2021][Volpi et al, 2018]:**

1. Use Wasserstein distance, less practical

2. No further investigation on the effect of different forms of AT

**Limited, not effective enough**

**[Herrmann et al, 2021]**

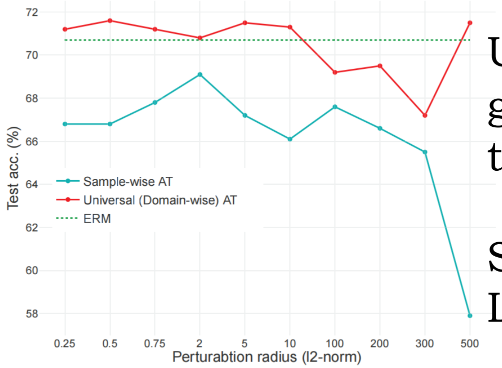
Do not exploit the universal spurious information (background/style)

verify the weaknesses

# Our findings

| Algorithm | Datasets PACS | OfficeHome | VLCS | NICO | avg |
|---|---|---|---|---|---|
| ERM | $79.7 \pm 0.0$ | $59.6 \pm 0.0$ | $74.4 \pm 1.0$ | $\mathbf{70.7 \pm 1.0}$ | 71.1 |
| AT | $\mathbf{81.5 \pm 0.4}$ | $\mathbf{59.9 \pm 0.4}$ | $\mathbf{75.3 \pm 0.7}$ | $68.2 \pm 2.2$ | **71.2** |

The improvement of sample-wise AT is **marginal**.




UAT (Universal AT) remains its generalization performance when the perturbation scale is large.

Samll perturbations ⟺ less like OOD shifts ✗

Large perturbations ⟺ more like OOD shifts ✓

conclude

**Low-rank, domain-wise structures are beneficial for OOD!**

motivate

# Our Methods: MAT & LDAT

# The Proposed Structured AT Method

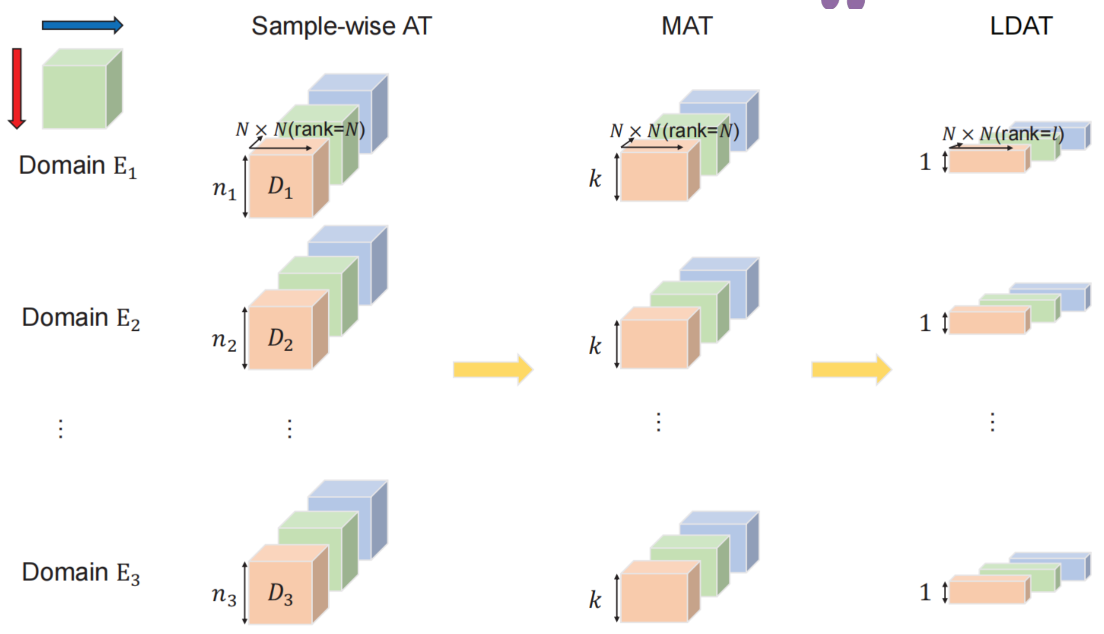
Reduce the rank of the adversarial perturbations along two orientations:

1. Reduce number of the perturbations used in a domain

   → what MAT & LDAT do

2. Reduce the rank of a single perturbation matrix

   → what LDAT does




**Low-rank, domain-wise structures are beneficial for OOD!**

motivate → Our Methods: MAT & LDAT

# The Proposed Structured AT Method

## MAT: AT with Combinations of Multiple Perturbations

NEURAL INFORMATION PROCESSING SYSTEMS

$$\min_f \sum \mathbb{E}_{(x,y)\sim\mathcal{P}_e(x,y)}[\mathcal{L}(f(x+\delta^e),y)]$$

$$\text{s.t. } \delta^e = \sum_{i=1}^{k} \alpha_i^{e*}\delta_i^{e*}, \|\delta_i^{e*}\|_p \le \epsilon, \sum_{i=1}^{k} \alpha_i^{e*} = 1, \alpha_i^{e*} \ge 0 \text{ for } i = 1, 2, \ldots, k$$

$$\alpha_i^{e*}, \delta_i^{e*} = \underset{\alpha_i^e, \delta_i^e}{\operatorname{argmax}} \mathbb{E}_{(x,y)\sim\mathcal{P}_e(x,y)}\left[\mathcal{L}\left(f\left(x + \sum_{i=1}^{k} \alpha_i^e \delta_i^e\right), y\right)\right]$$

- Domain-wise perturbation
- Perturbation is the linear combination of $k$ perturbations with learnable coefficients.
- Reducing the number of perturbations from $n_e$ to $k$

Reduce the number of the perturbations used in a domain ✓

Maintain some diverse structures to model more complex background ✓

# The Proposed Structured AT Method

## LDAT: Adversarial Training with Low-rank Decomposed Perturbations

$$\min_f \sum_e \mathbb{E}_{(x,y)\sim\mathcal{P}_e(x,y)}[\mathcal{L}(f(x+\delta^e),y)], \text{ s.t. } \delta^e = A^{e*}B^{e*}, \|\delta^e\|_p \le \epsilon,$$

$$A^{e*}, B^{e*} = \underset{A^e,B^e}{\text{argmax}}\, \mathbb{E}_{(x,y)\sim\mathcal{P}_e(x,y)}[\mathcal{L}(f(x+A^eB^e),y)], A^e \in \mathcal{R}^{N\times l\times C}, B^e \in \mathcal{R}^{l\times N\times C}$$

- Domain-wise perturbation
- Perturbation is low-rank: $\delta = AB$, $A$ and $B$ are matrices with rank$\le l$.
- Reducing the number of perturbations from $n_e$ to 1.
- Reducing the rank of a single perturbation from $N$ (input hight/width) to $l$.

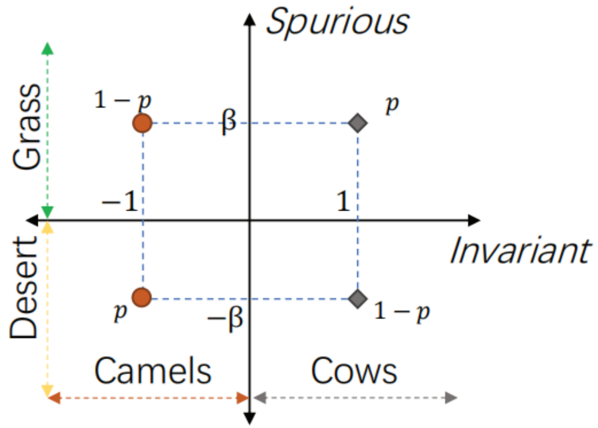
Reduce the number of the perturbations used in a domain ✓

Further reduce the rank of a single perturbation matrix ✓

# Theoretical Analysis

- **Our results:**

$$\Omega\left(\mathbb{E}_{(x_{inv},y)\sim\mathcal{D}_{inv}}\left[\frac{\frac{1}{\beta+\delta y}\ln\left[\frac{c_1+p}{c_2+p^{\frac{1}{2}-\epsilon}(1-p)^{\frac{1}{2}+\epsilon}}\right]}{M\ln(t+1)}\right]\right)\leq\frac{w_{sp}(t)\beta}{|w_{inv}(t)x_{inv}|},$$




**Remark:**

1. Term $\frac{w_{sp}(t)\beta}{|w_{inv}(t)x_{inv}|}$ denotes the **reliance of the model on spurious features.**
2. $p$ measures how strong the spurious correlation is
3. When using domain-wise perturbation adopted by MAT or LDAT, the lower bound of the reliance on spurious features does not increase with $p$ monotonically. However, when conducting ERM, this lower bound grows with $p$ monotonically.

**MAT/LDAT is better than ERM on OOD data!**

# Experiments

NEURAL INFORMATION PROCESSING SYSTEMS

- On Domainbed, an OOD generalization benchmark

| Algorithm | PACS | OfficeHome | VLCS | NICO | CMNIST | avg[1] | avg[2] | avg[3] | avg[4] |
|---|---|---|---|---|---|---|---|---|---|
| | Datasets | | | | | | | | |
| ERM (Our runs) | 81.7 ± 0.3 | 62.1 ± 0.1 | 74.4 ± 1.0 | 73.2 ± 1.9 | 28.1 ± 1.5 | 61.3 | 63.9 | 72.3 | 72.9 |
| AT (Our runs) | 82.6 ± 0.4 | 62.1 ± 0.3 | **76.2** ± **0.3** | 69.7 ± 1.6 | 29.1 ± 1.5 | 60.9 | 64.3 | 71.5 | 72.7 |
| ERM[21] | 81.5 ± 0.0 | 63.3 ± 0.2 | - | 71.4 ± 1.3 | 29.9 ± 0.1 | 61.5 | - | 72.1 | - |
| RSC[24] | **82.8** ± 0.4 | 62.9 ± 0.4 | - | 69.7 ± 0.3 | 28.6 ± 1.5 | 61.0 | - | 71.8 | - |
| MMD[25] | 81.7 ± 0.2 | 63.8 ± 0.1 | - | 68.3 ± 1.8 | 50.7 ± 0.1 | 66.1 | - | 71.3 | - |
| SagNet[26] | 81.6 ± 0.4 | 62.7 ± 0.4 | - | 69.3 ± 1.0 | 30.5 ± 0.7 | 61.0 | - | 71.2 | - |
| CORAL[27] | 81.6 ± 0.6 | 63.8 ± 0.3 | - | 68.3 ± 1.4 | 30.0 ± 0.5 | 61.0 | - | 71.2 | - |
| IRM[1] | 81.1 ± 0.3 | 63.0 ± 0.2 | - | 67.6 ± 1.4 | 60.2 ± 2.4 | 68.0 | - | 70.6 | - |
| VREx[23] | 81.8 ± 0.1 | 63.5 ± 0.1 | - | 71.0 ± 1.3 | 56.3 ± 1.9 | 68.2 | - | 72.1 | - |
| GroupDRO[28] | 80.4 ± 0.3 | 63.2 ± 0.2 | - | 71.8 ± 0.8 | 32.5 ± 0.2 | 62.0 | - | 71.8 | - |
| DANN[29] | 81.1 ± 0.4 | 62.9 ± 0.6 | - | 68.6 ± 1.1 | 24.5 ± 0.8 | 59.3 | - | 70.9 | - |
| MTL[30] | 81.2 ± 0.4 | 62.9 ± 0.2 | - | 70.2 ± 0.6 | 29.3 ± 0.1 | 60.9 | - | 71.4 | - |
| Mixup[31] | 79.8 ± 0.6 | 63.3 ± 0.5 | - | 66.6 ± 0.9 | 27.6 ± 1.8 | 59.3 | - | 69.9 | - |
| ANDMask[32] | 79.5 ± 0.0 | 62.0 ± 0.3 | - | 72.2 ± 1.2 | 27.2 ± 1.4 | 60.2 | - | 71.2 | - |
| MLDG[33] | 73.0 ± 0.4 | 52.4 ± 0.2 | - | 51.6 ± 6.1 | 32.7 ± 1.1 | 52.4 | - | 59.0 | - |
| MAT (Our work) | 82.3 ± 0.5 | **64.5** ± **2.1** | 74.6 ± 0.8 | 74.2 ± 1.5 | **65.4** ± **8.1** | **71.6** | **72.2** | **73.7** | **73.9** |
| LDAT (Our work) | 82.6 ± 0.5 | 61.0 ± 0.9 | 75.3 ± 0.3 | **74.4** ± **1.6** | 52.5 ± 5.4 | 67.6 | 69.1 | 72.7 | 73.3 |

- MAT and LDAT outperform ERM and AT, ranked 1st and 4th among all algorithms.

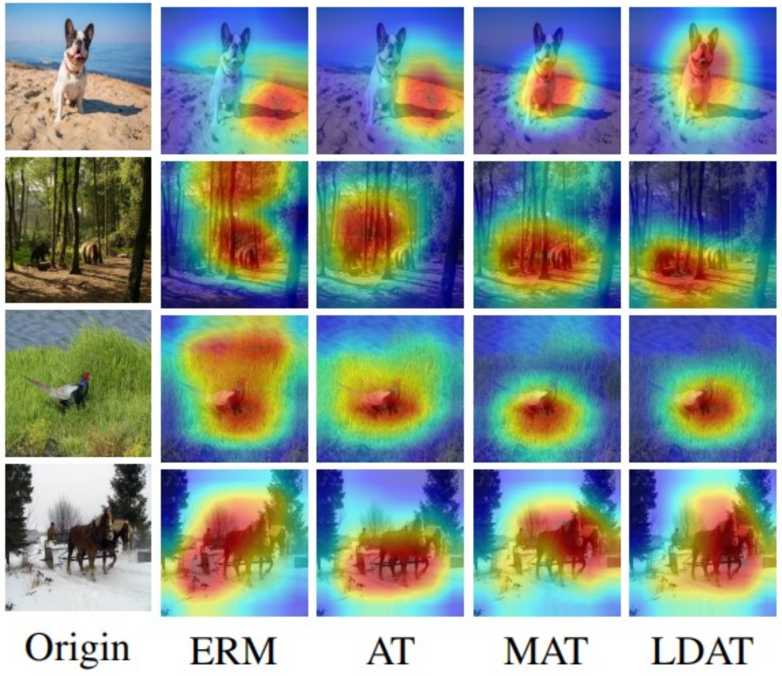
- MAT and LDAT beat GUT (Volpi et al, 2018) and NCDG (Tian et al, 2022), two data augmentation methods for OOD

| ERM | MAT | LDAT | GUT |
|---|---|---|---|
| 73.2 ± 1.9 | **74.2** ± **1.5** | 74.4 ± 1.6 | 66.6 ± 1.7 |

| Algorithm | A | C | P | S | avg |
|---|---|---|---|---|---|
| MAT | - | 73.8 | 94.1 | 74 | 80.6 |
| | 78.5 | - | 94.2 | 75.9 | 82.9 |
| | 80.9 | 73.7 | - | 76.6 | 77.1 |
| | 80.4 | 76.3 | 93.3 | - | 83.3 |
| LDAT | - | 74.8 | 94.2 | 75.8 | 81.6 |
| | 77.2 | - | 93.9 | 75.6 | 82.3 |
| | 78.5 | 77.9 | - | 80.4 | 79 |
| | 74.3 | 76.4 | 94.7 | - | 81.8 |
| NCDG | - | 68.6 | 95.0 | 66.4 | 76.6 |
| | 71.6 | - | 85.8 | 71.9 | 76.4 |
| | 68.8 | 29.8 | - | 48.6 | 49.0 |
| | 45.6 | 65.8 | 47.9 | - | 53.1 |

# Experiments

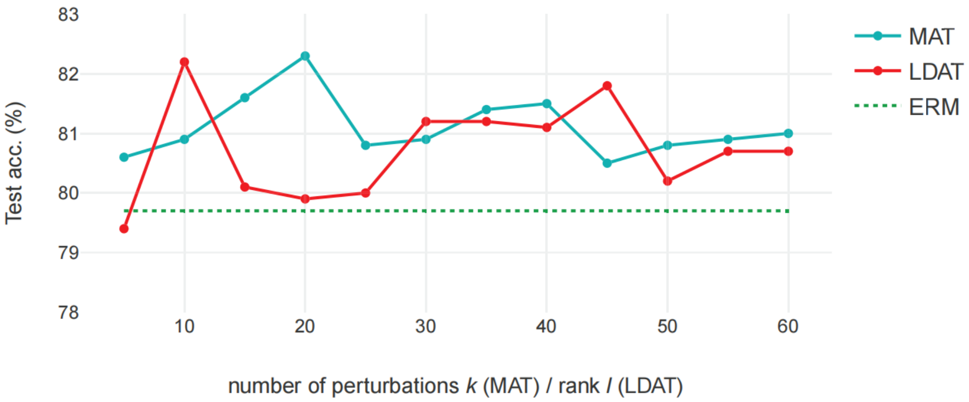
- Visualization (GradCam)




- MAT and LDAT better focus on the object rather than background.

- Impact of the rank hyperparameter $k$ (MAT) and $l$ (LDAT)

on PACS




on CMNIST

| Algorithm | $k \in [5, 20], l \in [10, 20]$ | $k$ or $l = 200$ | $k$ or $l = 500$ | $k$ or $l = 1000$ |
|---|---|---|---|---|
| MAT | $65.4 \pm 8.1$ | $34.9 \pm 20.2$ | $25.6 \pm 8.5$ | $23.4 \pm 10.8$ |
| LDAT | $52.5 \pm 5.4$ | $24.9 \pm 8.9$ | $19.0 \pm 6.6$ | $10.3 \pm 0.1$ |

**Insights:**
- rank is samll enough: good performance
- rank is too samll or too big: bad performance

Thanks!