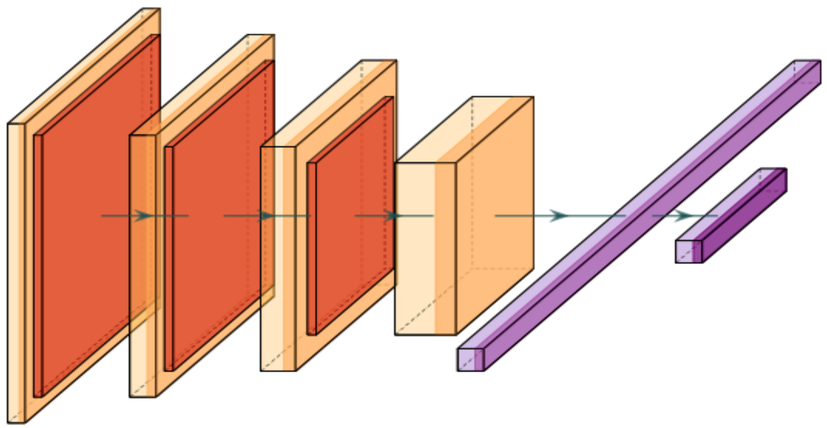
PGrad: Learning Principal Gradients for Domain Generalization

Zhe Wang, Jake Grigsby, Yanjun Qi

University of Virginia

UNIVERSITY of VIRGINIA

Domain Generalization Challenge



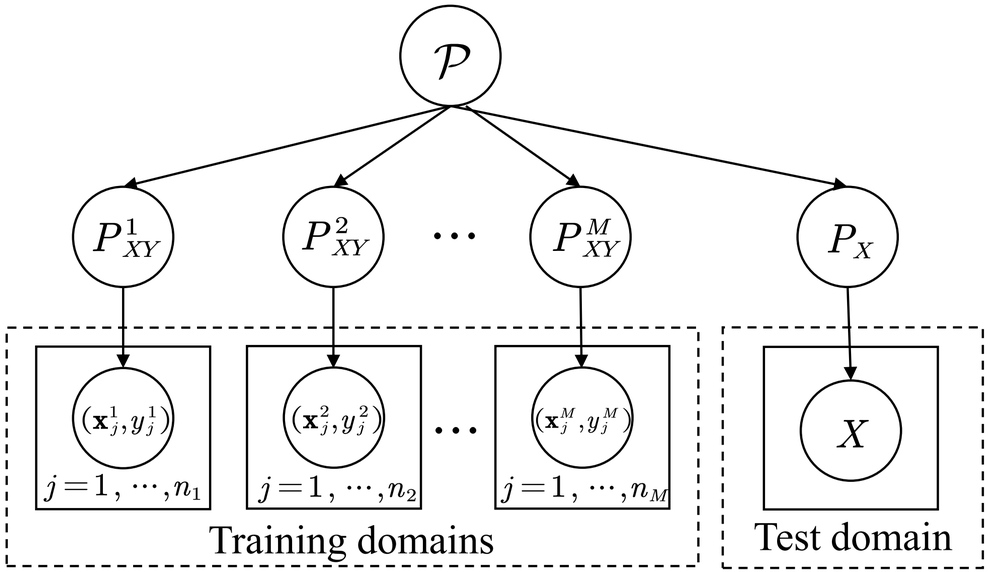
$f_{\theta*}$

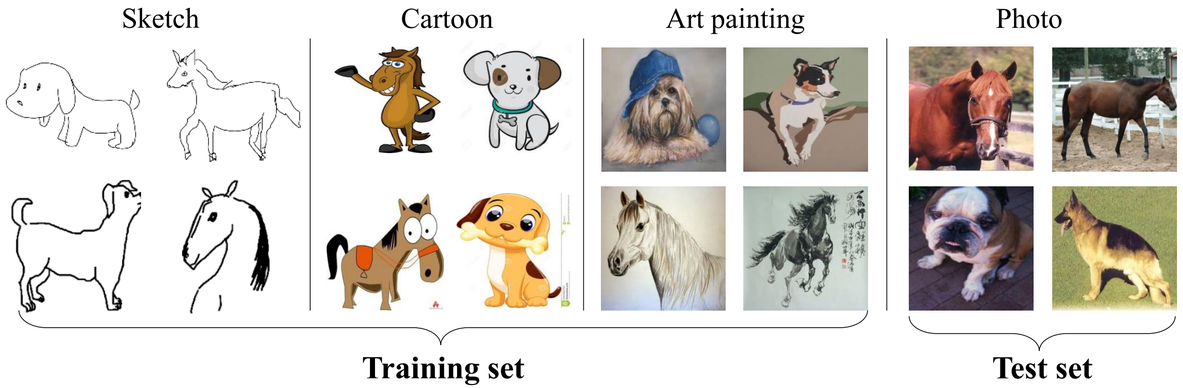
T_1

T_2

T_3

Domain Generalization Challenge





Zhou, et.al, "Domain generalization: A survey," in IEEE TPAMI, 2022.

Math Formulation

Domain generalization assumes no access to instances from future test domains.

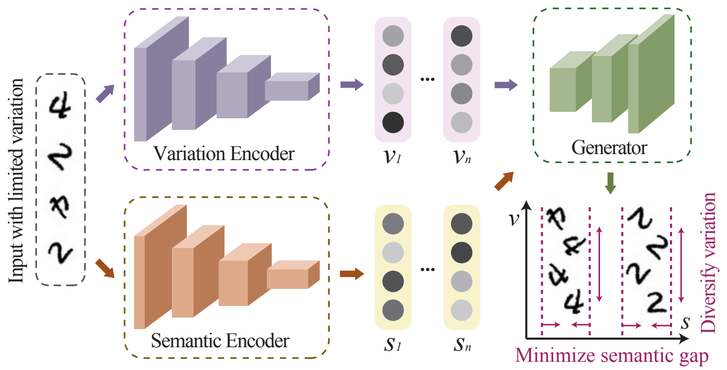
Training:

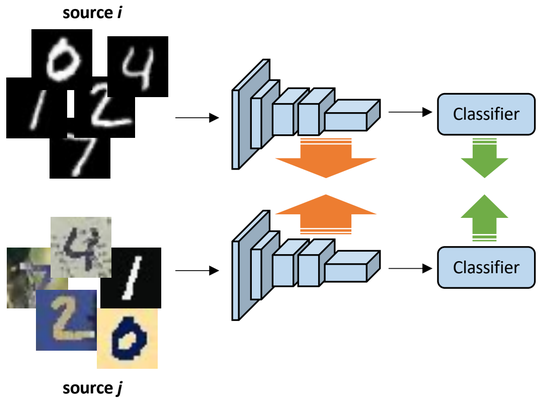
- We have a set of training domains $\mathscr{D}_{tr} = \{D_i\}_{i=1}^{n}$;
- D_i is characterized by i.i.d samples $\{\mathbf{x}_k^i, \mathbf{y}_k^i\}$
- D_i is associated with a joint distribution $P_{X\times Y}^{D_i}$, and $P_{X\times Y}^{D_i} \neq P_{X\times Y}^{D_j}$

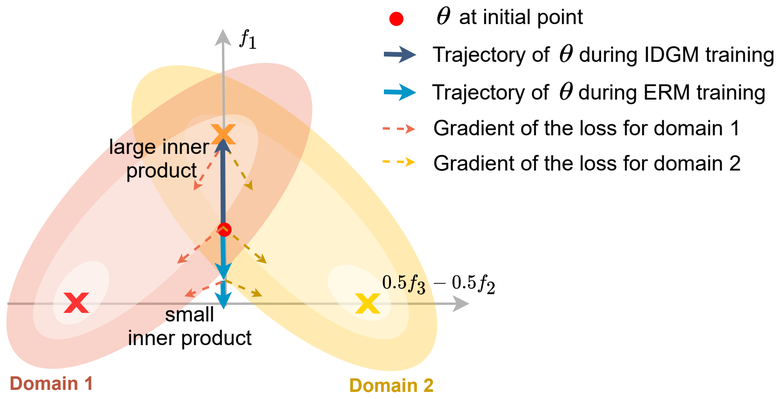
Testing:

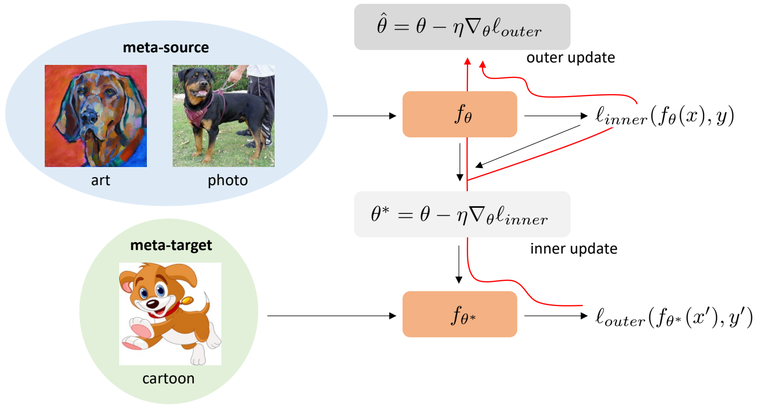
- we evaluate the model on test domains $\mathscr{T}_{te} = \{T_j\}_{j=1}^{m}$, where $P_{X\times Y}^{D_i} \neq P_{X\times Y}^{T_j}$.

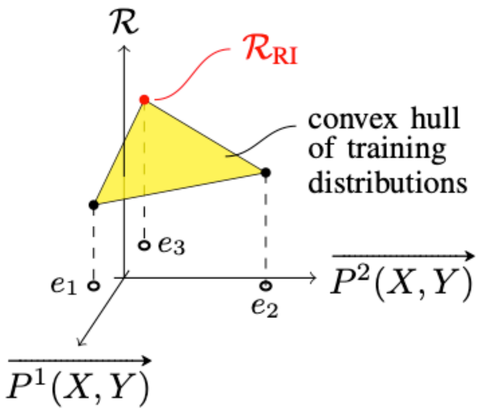
Previous Work: Feature, Optimization, and Beyond

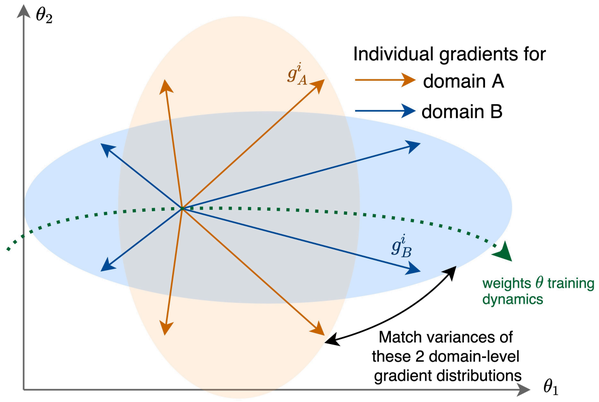












Piratla, et.al, "Efficient domain generalization via common-specific low-rank decomposition," in ICML, 2020.

Li et.al, "Learning to generalize: Meta-learning for domain generalization," in AAAI, 2018.

Shi et.al, "Gradient Matching for Domain Generalization," in ICLR, 2022.

Chattopadhyay et.al "Learning to balance specificity and invariance for in and out of domain generalization," in ECCV, 2020.

Krueger et.al "Out-of-Distribution Generalization via Risk Extrapolation (REx)" in Arxiv, 2021.

Rame et.al, "Fishr: Invariant Gradient Variances for Out-of-Distribution Generalization," in ICML 2022.

Limitations

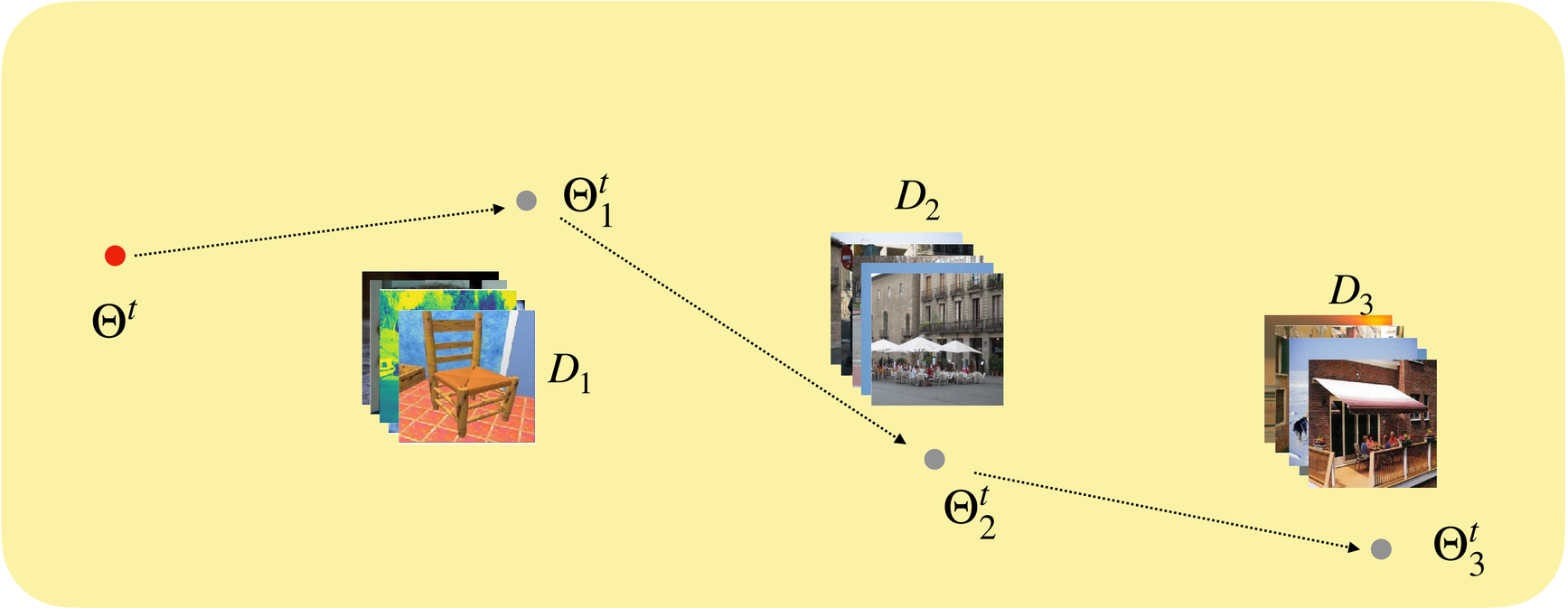
- Unsatisfied Empirical Performance;
  - *Empirical Risk Minimization* remains the strong baseline.
- Strong Distributional Assumption;
  - Invariant marginal or conditional distributions.
- Dataset or Task Dependent.

Gulrajani et.al, "In Search Of Lost Domain Generalization," in ICLR, 2021.

Galstyan et.al "Failure Mode of Domain Generalization Algorithms" in CVPR, 2022.

Blanchard et.al "Generalizing from several related classification tasks to a new unlabeled sample", in NeurIPS, 2011

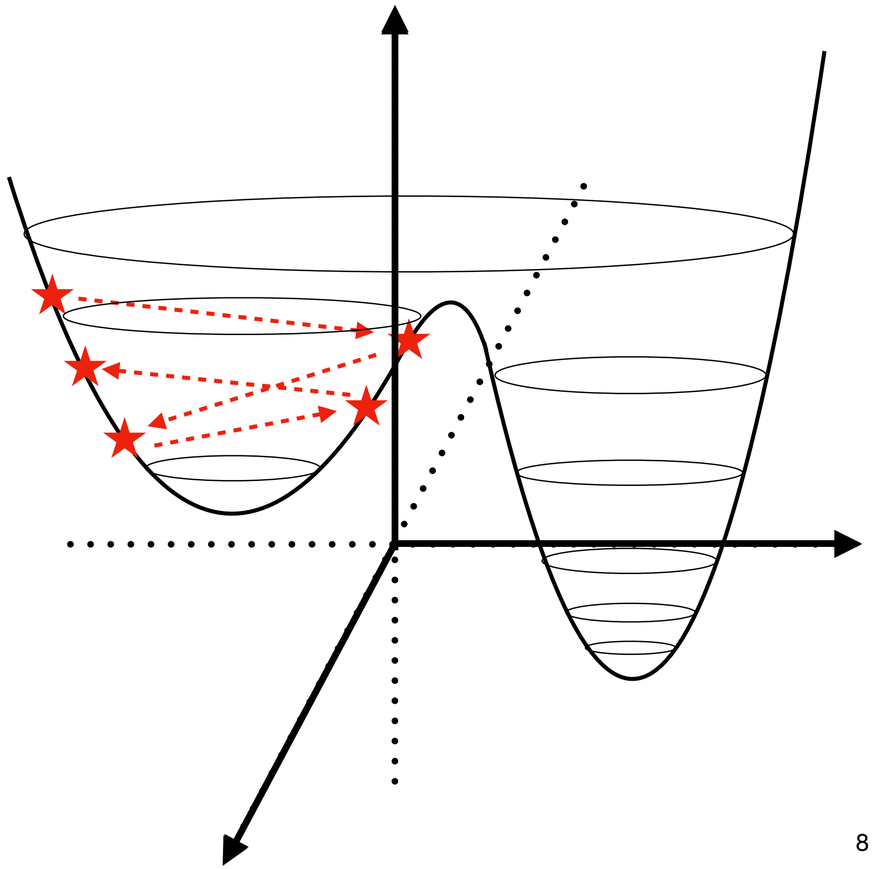
Proposed: Trajectory Based Direction Learning

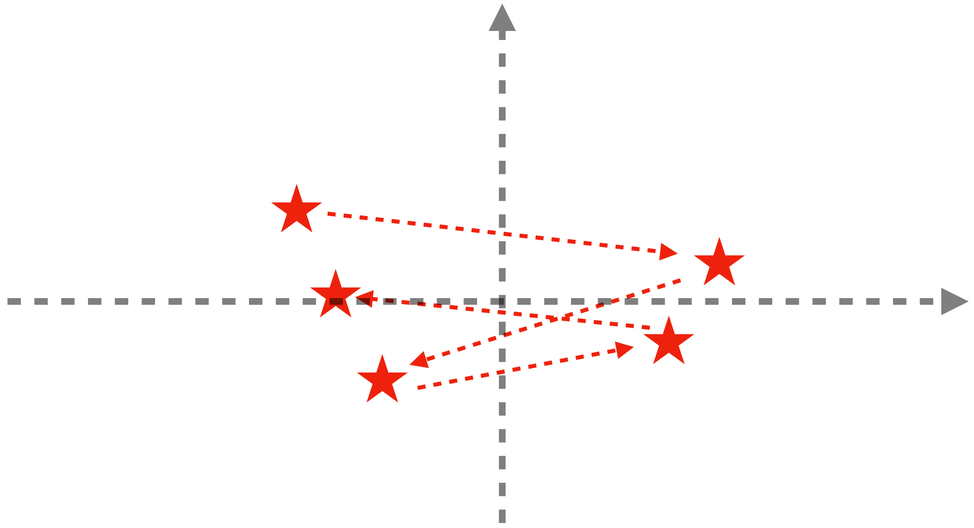


Learning From Optimization Trajectories

Why Trajectory Based?

- Disentangle dominant directions for analysis. For example:
  - Zeroing out the insignificant directions.
  - Adding weights for significant directions.





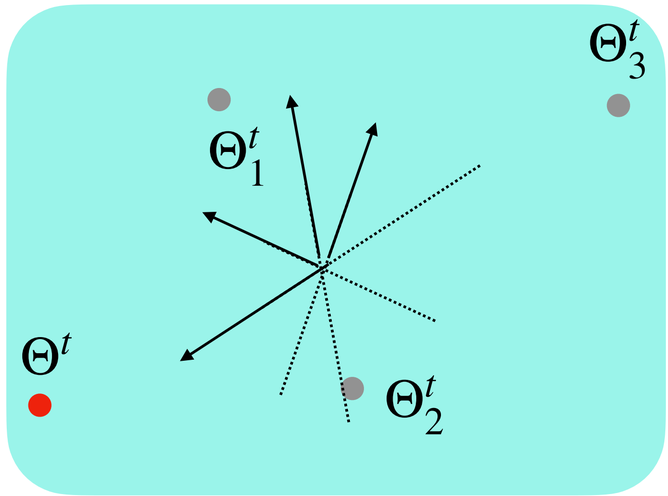
Principal Direction Learning (Our Main Idea)

1. Trajectory Sampling;

$$S = \Theta^t \to \Theta_1^t \to \Theta_2^t \to \cdots \to \Theta_n^t .$$

2. Local Principal Coordinate Construction

$$\max_{V_z} \mathrm{Var}([Sv_z]), \quad \text{s.t. } V^T V = \mathbf{I}_d .$$



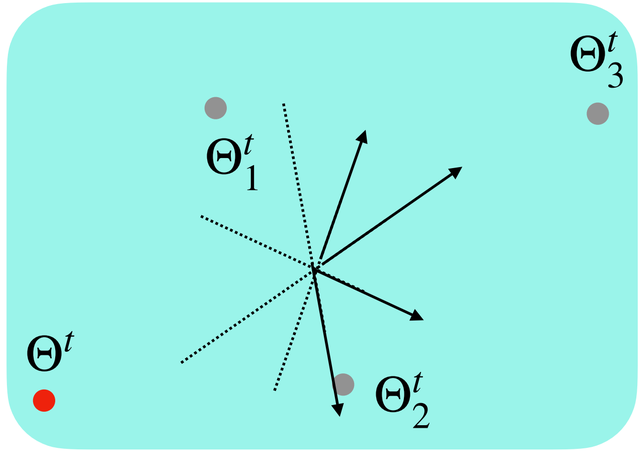
Principal Direction Learning (Our Main Idea)

3. Direction Calibration

Teach the model to climb down the loss landscape with:

$$\nabla_r = \Theta^t - \Theta^t_n .$$

$$\mathbf{w}_z = r_z \mathbf{v}_z, \quad r_z = \begin{cases} 1, & \text{if} < \mathbf{v}_z, \nabla_r > \geq 0 \\ -1, & \text{otherwise} \end{cases}$$



4. Principal Direction Learning

$$\nabla_p = \sum_{z=0}^{n} \frac{\lambda_z}{||\lambda||_2} w_z . \quad \Longrightarrow \quad ||\nabla_p|| = 1.$$

Principal Direction Learning (Our Main Idea)

3. Direction Calibration

$$\nabla_r = \Theta^t - \Theta_n^t .$$

4. Principal Direction Learning

$$\nabla_p = \sum_{z=0}^{n} \frac{\lambda_z}{||\lambda||_2} w_z . \quad \Longrightarrow \quad ||\nabla_p|| = 1.$$

5. Optimization Process Aware LR Tuning

$$\nabla_p = \sum_{z=0}^{k} \frac{\lambda_z ||\nabla_r||}{||\lambda||_2} w_z . \quad \Longrightarrow \quad \text{As Training Continues, } ||\nabla_r|| \to 0$$

Principal Direction Learning (Computational Challenge in Step 2)

$$\max_{V_z} \text{Var}([Sv_z]), \quad \text{s.t. } V^T V = \mathbf{I}_d .$$

- Closed form solution:

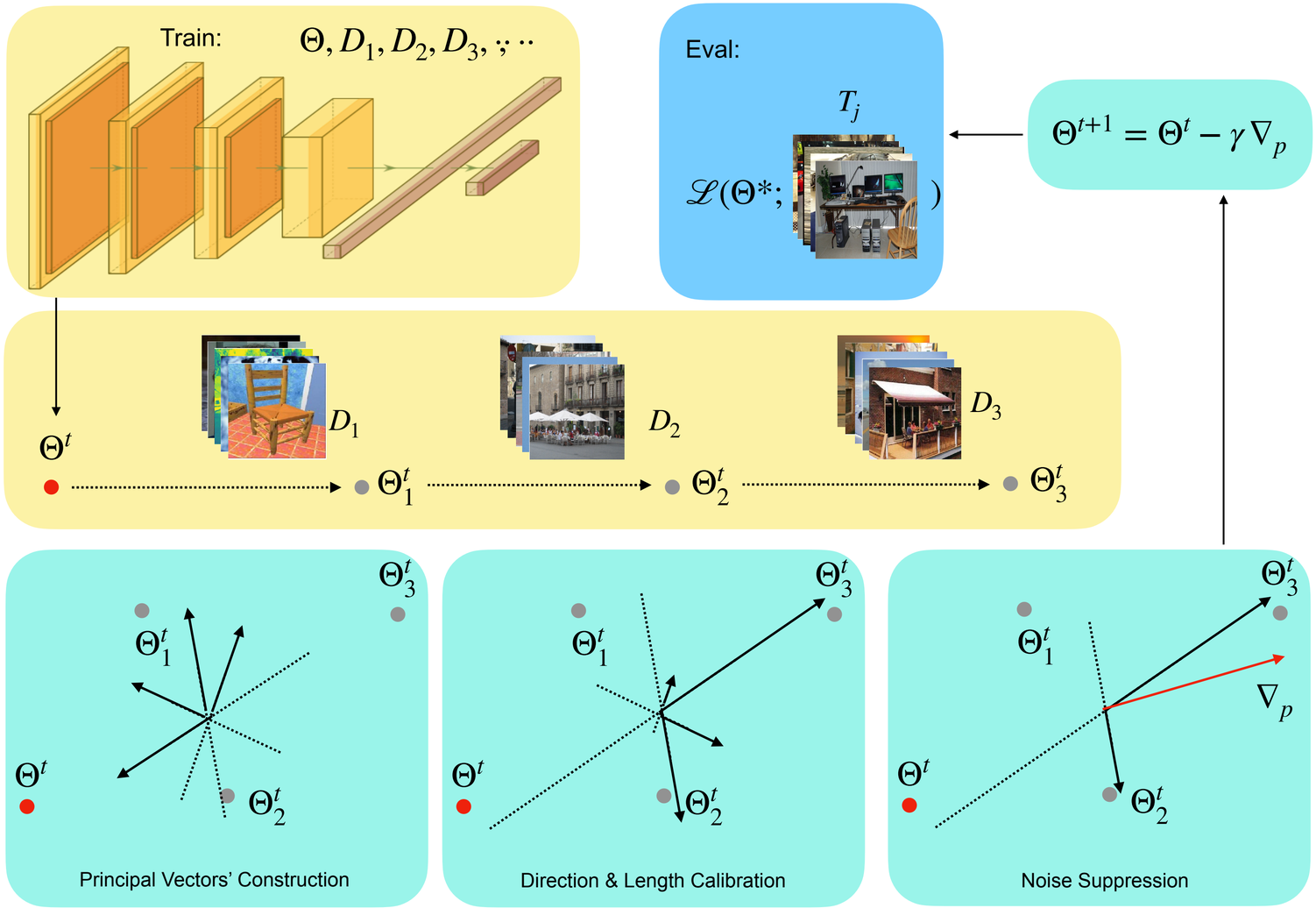
$$\lambda_z, \mathbf{v}_z = \text{SVD}_z(\frac{1}{n}\hat{S}^T\hat{S}), \quad \text{where} \quad \hat{S}^T\hat{S} \in \mathbb{R}^{d\times d}$$

- Challenge: The computational complexity of the SVD $\mathcal{O}(d^3)$, where $d \sim \mathcal{O}(10^{10})$
- Solution: Transpose trick.

$$\hat{S}\hat{S}^T\mathbf{e}_z = \lambda_z\mathbf{e}_z \implies \hat{S}^T\hat{S}\hat{S}^T\mathbf{e}_z = \lambda_z\hat{S}^T\mathbf{e}_z \implies \mathbf{v}_z = \hat{S}^T\mathbf{e}_z$$

- Analysis: The computational complexity is $\mathcal{O}(n^3)$, where $n = 3, 4, \cdots$

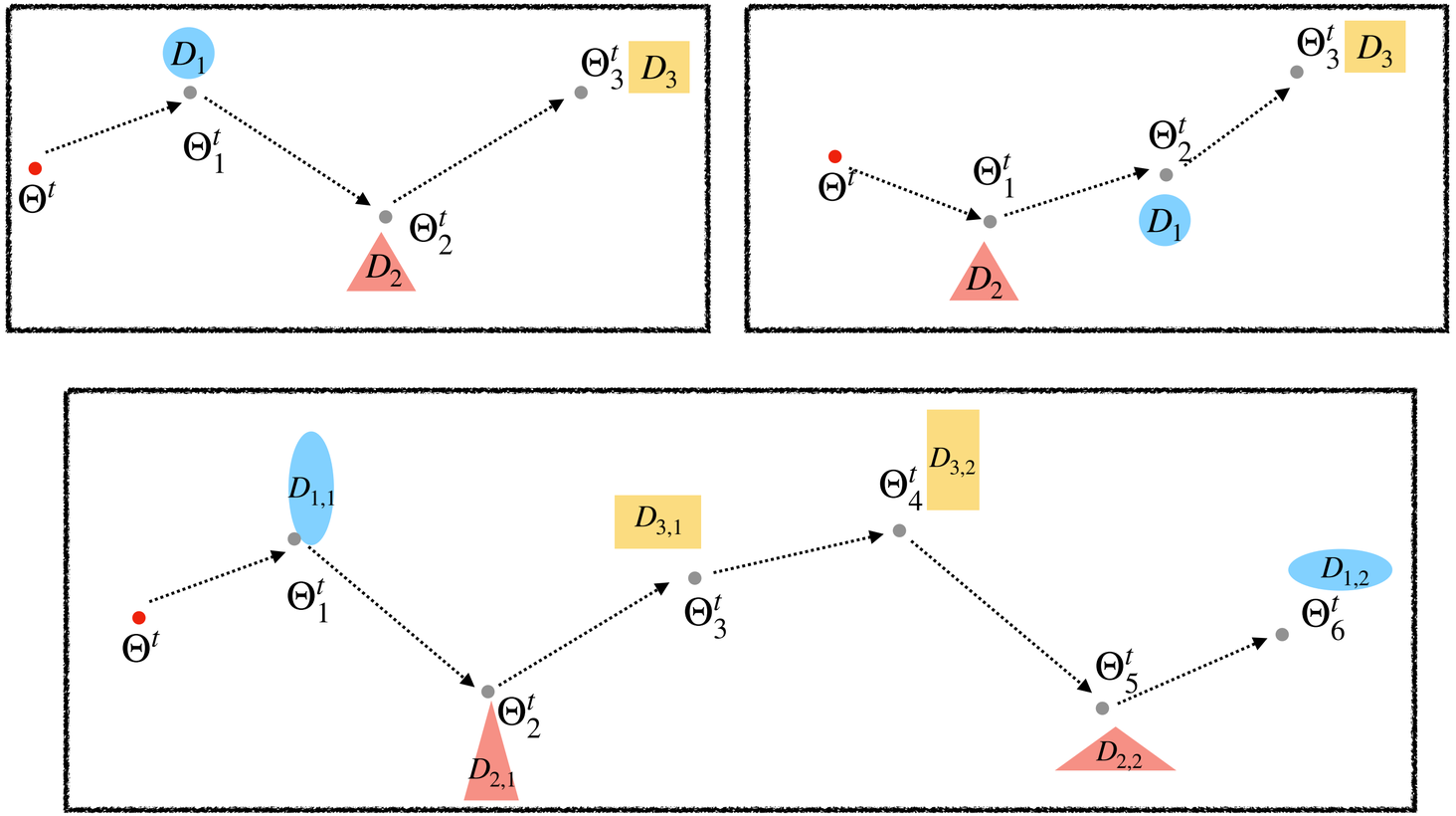
PGrad: Learning Principal Gradients for Domain Generalization



Wang et.al, "PGrad: Learning Principal Gradients for Domain Generalization," in ICLR, 2023.

Principal Direction Learning (Optional)

- Can we sample more flexible and longer trajectories?



PGrad & Variants

- ***PGrad***;
  - Random shuffling the domain order during training.
- ***PGrad-B***(default);
  - Split each training batch into B smaller batches.
- ***PGrad-BMix***
  - Combine **PGrad-B** and **MixUp**

Zhang et.al, "Mixup: Beyond Empirical Risk Minimization," in ICLR, 2018.

One of the Benchmark Suits – **DomainBed**

- Image Dataset - Classification - ResNet-50

Table 1: A summary on DomainBed dataset, metrics, and architectures we used.

Dataset	# of Images	Domains	# of Classes
PACS (Li et al., 2017)	9,991	Artpaint, Cartoon, Sketches, Photo	7
VLCS (Fang et al., 2013)	10,729	PASCAL VOC 2007, LabelMe, Caltech, Sun	5
OfficeHome (Venkateswara et al., 2017)	15,588	Art, Clipart, Product, Real-World	65
TerraIncognita (Beery et al., 2018)	24,788	Location #100, #38, #43, #46	10
DomainNet (Peng et al., 2019)	586,575	Clipart, Infograph, Painting, Quickdraw, Real, Sketch	345

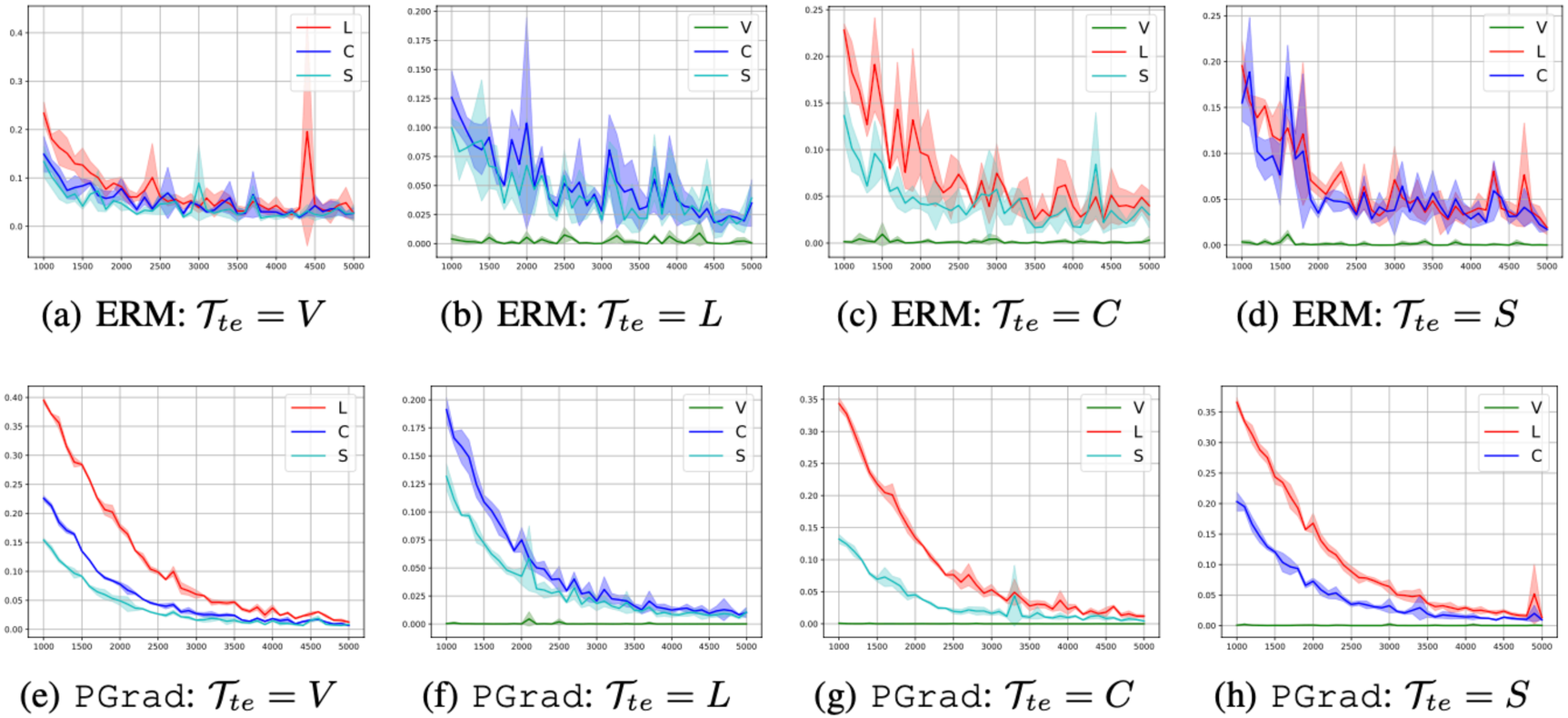
- Evaluation protocol
  - We select $k-1$ as training domains and the remaining for testing.
  - For each combination, we repeat 3(random seeds) × 2(hyperparameters) running.

DomainBed Results - Prediction Acc Improvement

Table 2: Test accuracy (%) on five datasets from the DomainBed benchmark. We group 20% data from each training domain to construct validation set for model selection.

Categories	Algorithms	VLCS	PACS	OfficeHome	TerraInc	DomainNet	**Avg**
Baseline	ERM	77.5 ± 0.4	85.5 ± 0.2	66.5 ± 0.3	46.1 ± 1.8	40.9 ± 0.1	63.3
Invariant	IRM	78.5 ± 0.5	83.5 ± 0.8	64.3 ± 2.2	47.6 ± 0.8	33.9 ± 2.8	61.6 **–1.7**
	MMD	77.5 ± 0.9	84.6 ± 0.5	66.3 ± 0.1	42.2 ± 1.6	23.4 ± 9.5	58.8 **–4.5**
	DANN	78.6 ± 0.4	83.6 ± 0.4	65.9 ± 0.6	46.7 ± 0.5	38.3 ± 0.1	62.6 –0.7
	CORAL	78.8 ± 0.6	<u>86.2</u> ± 0.3	68.7 ± 0.3	47.6 ± 1.0	41.5 ± 0.1	64.5 **+1.2**
Optimization	GroupDRO	76.7 ± 0.6	84.4 ± 0.8	66.0 ± 0.7	43.2 ± 1.1	33.3 ± 0.2	60.7 **–2.6**
	MLDG	77.2 ± 0.4	84.9 ± 1.0	66.8 ± 0.6	47.7 ± 0.9	41.2 ± 0.1	63.6 +0.3
Augmentation	MixUp	77.4 ± 0.6	84.6 ± 0.6	68.1 ± 0.3	47.9 ± 0.8	39.2 ± 0.1	63.4 +0.1
	ARM	77.6 ± 0.3	85.1 ± 0.4	64.8 ± 0.3	45.5 ± 0.3	35.5 ± 0.2	61.7 **–1.6**
Gradient Manipulation	Fish	77.8 ± 0.3	85.5 ± 0.3	68.6 ± 0.4	45.1 ± 1.3	**42.7** ± 0.2	63.9 +0.6
	Fishr	77.8 ± 0.1	85.5 ± 0.4	67.8 ± 0.1	47.4 ± 1.6	41.7 ± 0.0	64.0 +0.7
	PGrad	**79.3** ± 0.3 **+1.8**	85.1 ± 0.3 –0.4	69.3 ± 0.1 **+2.8**	49.0 ± 0.3 **+2.9**	41.0 ± 0.1 +0.1	64.7 **+1.4**
	PGrad-B	<u>78.9</u> ± 0.3 **+1.4**	**87.0** ± 0.1 **+1.5**	<u>69.6</u> ± 0.1 **+3.1**	<u>49.4</u> ± 0.8 **+3.3**	41.4 ± 0.1 +0.5	<u>65.3</u> **+2.0**
	PGrad-BMix	<u>78.9</u> ± 0.2 **+1.4**	<u>86.2</u> ± 0.4 **+0.7**	**69.8** ± 0.1 **+3.3**	**50.7** ± 0.6 **+4.6**	<u>42.6</u> ± 0.2 **+1.7**	**65.7** **+2.4**

Analyzing Training Process – Loss Smoothing

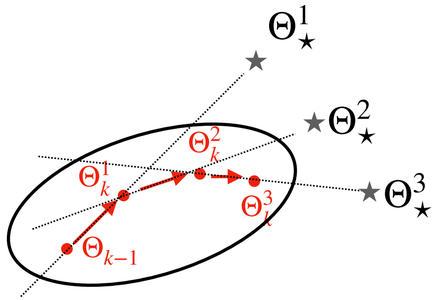


(a) ERM: $\mathcal{T}_{te} = V$ (b) ERM: $\mathcal{T}_{te} = L$ (c) ERM: $\mathcal{T}_{te} = C$ (d) ERM: $\mathcal{T}_{te} = S$

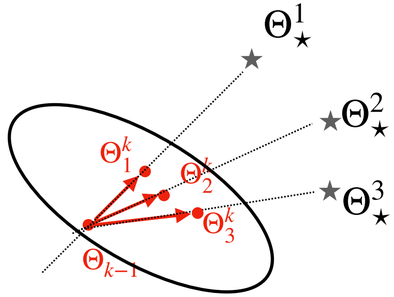
(e) PGrad: $\mathcal{T}_{te} = V$ (f) PGrad: $\mathcal{T}_{te} = L$ (g) PGrad: $\mathcal{T}_{te} = C$ (h) PGrad: $\mathcal{T}_{te} = S$

Figure 2: Visualzing domain-wise training losses on VLCS. Curves are the average over 9 runs, surrounded by $\pm\sigma$ the standard deviation. For comparison, the loss curves start from 1,000 epochs.

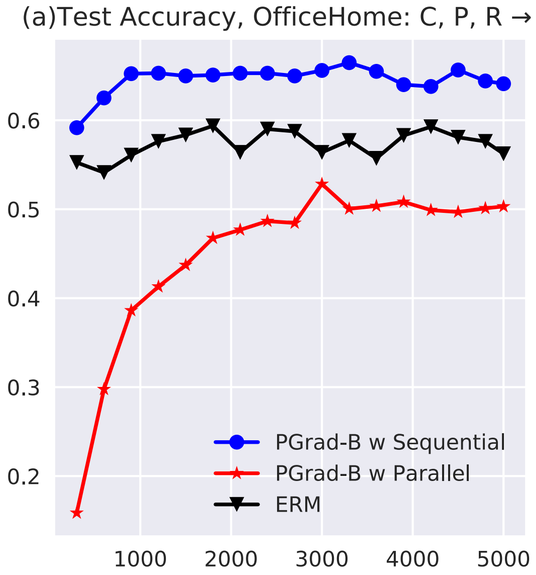
Comparing Sequential VS. Parallel Training

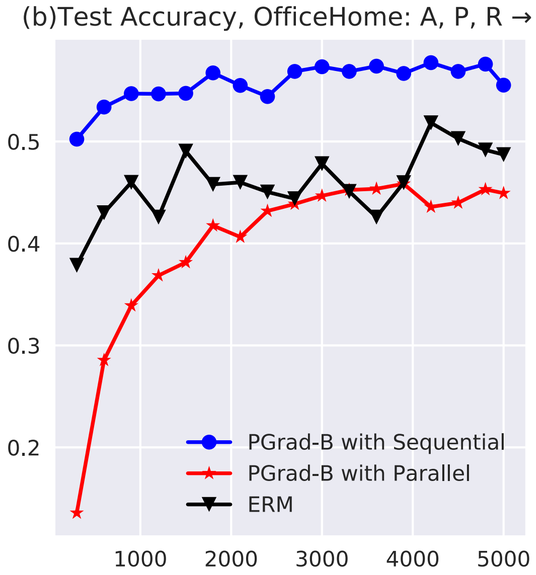


(a): Sequential training will reinforce robust direction



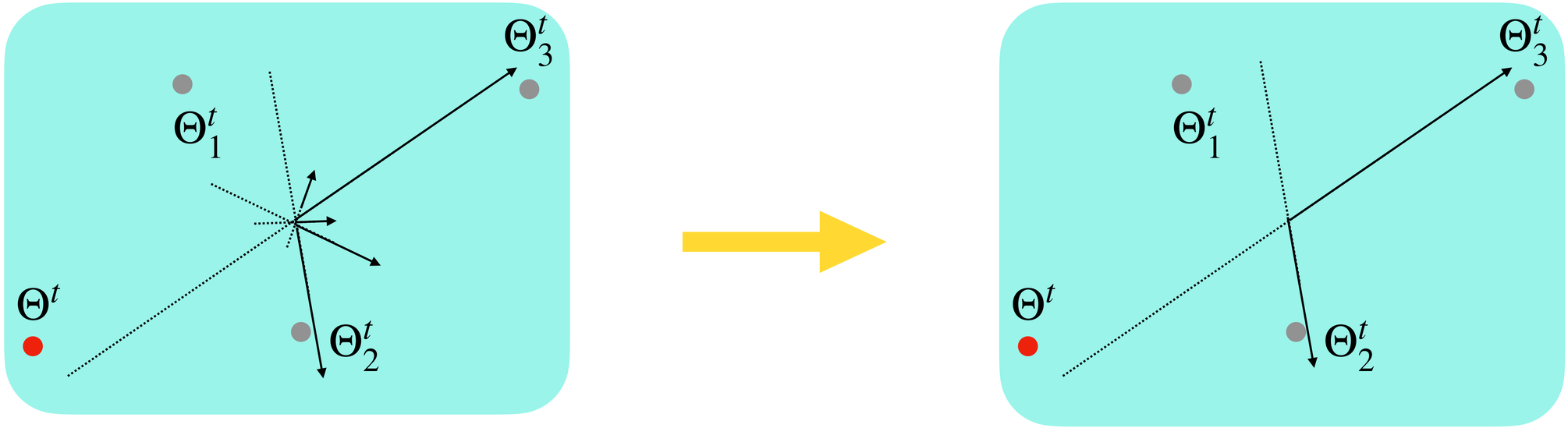
(b): Parallel training will suppress robust direction





Analyzing PGrad Hyperparameter-k

- Suppressing insignificant principal directions



How Many Bottom Directions to Suppress?

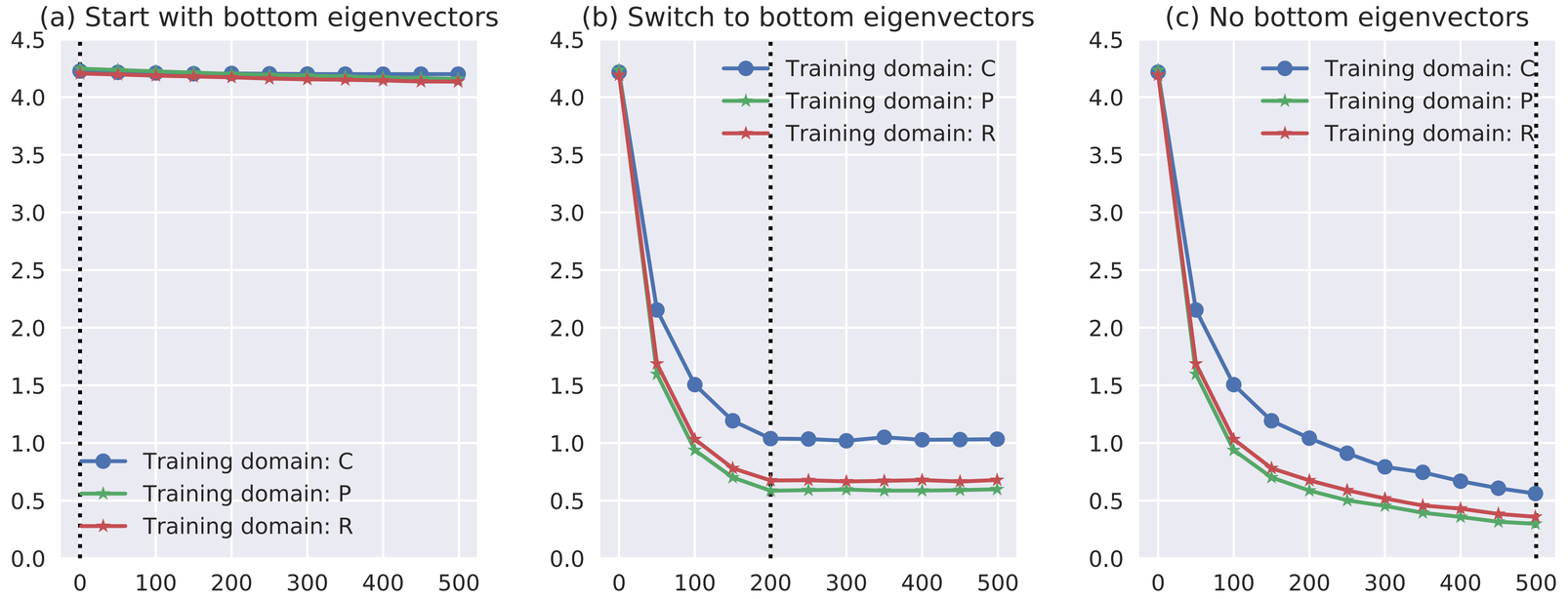


Table 3: Analysis the effect of varying k. The experiments are performed on PACS dataset. We highlight **first** and second best results.

Method	Algorithms	P	A	C	S	Avg
PGrad	$k=0$	98.0±0.2	87.3±0.2	76.8±0.4	73.4±1.3	83.9
	$k=2$	97.8±0.0	87.5±0.3	78.2±0.8	74.0±1.5	84.4
	$k=3$	97.8±0.0	87.8±0.4	78.4±0.6	77.2±1.1	85.3
	$k=4$	97.4±0.1	87.6±0.3	79.1±1.0	76.3±1.2	85.1
PGrad-B	$k=0$	97.5±0.1	89.1±0.8	80.3±0.6	77.5±0.4	86.1
	$k=2$	97.7±0.2	88.5±1.0	79.9±1.1	79.2±0.7	86.4
	$k=4$	98.0±0.2	**89.9**±0.2	80.0±0.6	**80.1**±0.9	**87.0**
	$k=7$	97.6±0.3	88.2±0.8	**81.1**±1.3	79.0±1.5	86.5

Analyzing Training Time Cost

Model's time efficiency is defined as the required time (seconds) for each update

Table 12: Training time of the PGrad-B evaluated on both PACS and OfficeHome dataset.

Method	PACS				OFFICEHOME			
	P	A	C	S	A	C	P	R
DANN	1.18	1.23	1.10	1.17	1.39	1.47	1.25	1.26
MLDG	1.03	1.02	1.00	1.01	0.49	0.46	0.43	0.41
Fish	1.63	1.66	1.61	1.58	1.47	1.57	1.45	1.58
PGrad-B	1.62	1.67	1.63	1.70	1.70	1.73	1.75	1.71

Thanks for attending our presentation!