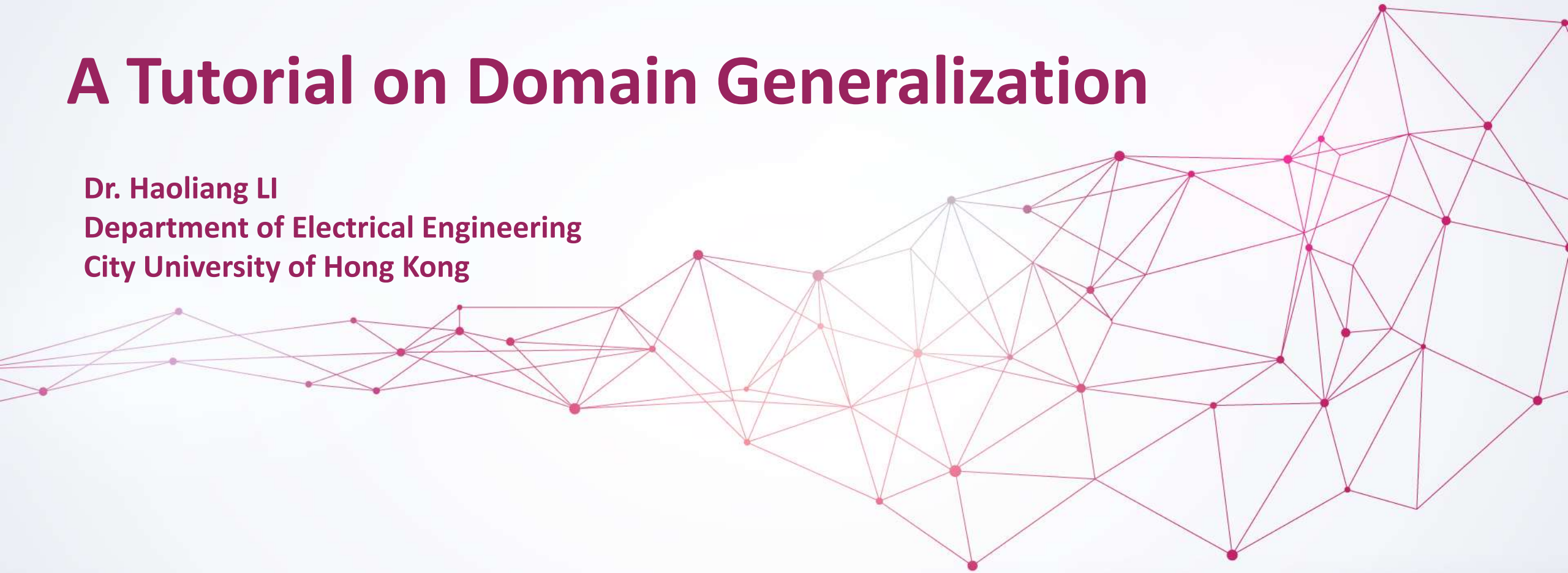
香港城市大學
City University of Hong Kong

A Tutorial on Domain Generalization

Dr. Haoliang LI
Department of Electrical Engineering
City University of Hong Kong



專業 創新 胸懷全球
Professional · Creative
For The World

Organizers

- Jindong Wang
  - Microsoft Research Asia, jindong.wang (at) microsoft.com
- Haoliang Li
  - City University of Hong Kong, haoliang.li (at) cityu.edu.hk
- Sinno Jialin Pan
  - The Chinese University of Hong Kong, sinnopan (at) cuhk.edu.hk
- Xing Xie
  - Microsoft Research Asia, xingx (at) microsoft.com

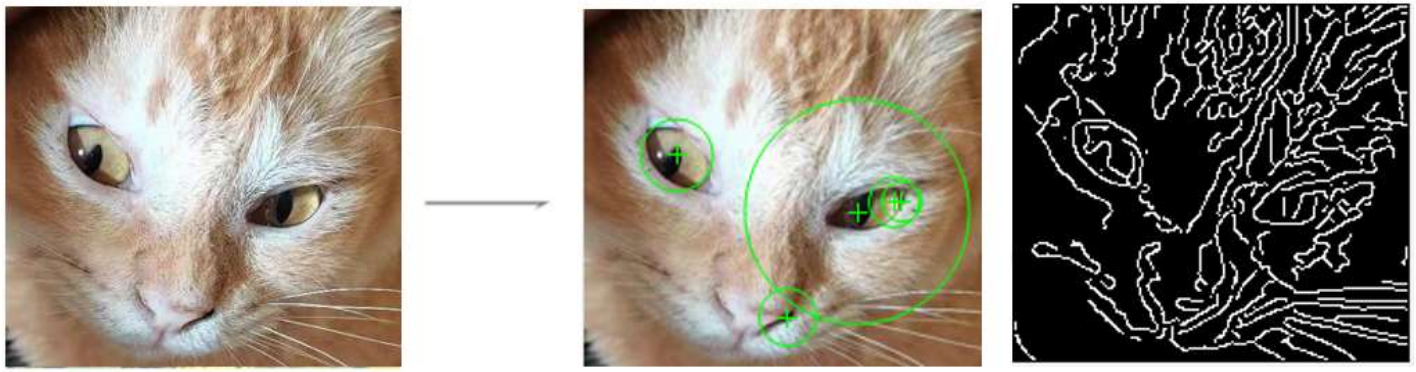
Outlines

- Fundamental of Domain Generalization (Dr. Haoliang Li)
  - Problem, Algorithms, Benchmark, and Beyond
- When DG meets ChatGPT (Dr. Jindong Wang)

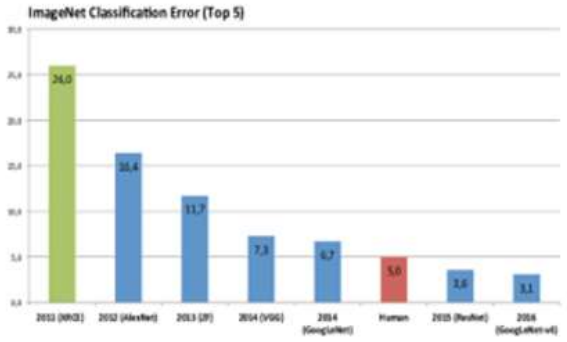
Background

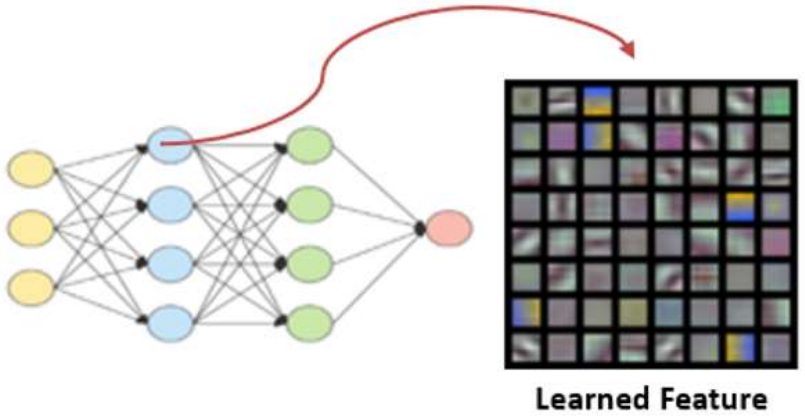
- Computer vision: How do we represent an image?

Past:



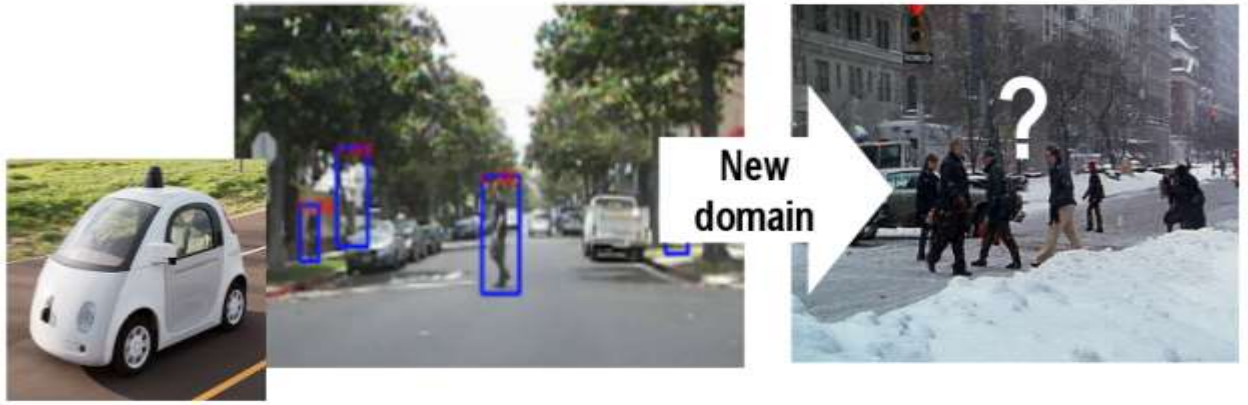
Now:





Background

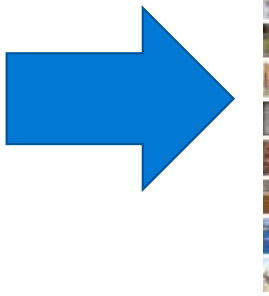
- Models do not generalize well to new domains; not like humans!
- Are big data always available?
  - It is impossible to consider data in all scenarios.
  - Data can be protected under privacy regulation.



Sinno Jialin Pan, *et.al.*, "A Survey on Transfer Learning." IEEE TKDE 2010

Domain Adaptation: Train on Source and Adapt to Target

ImageNet



CiFAR100

Source Domain $\sim P_S(X, Y)$

lots of **labeled** data

$\neq$

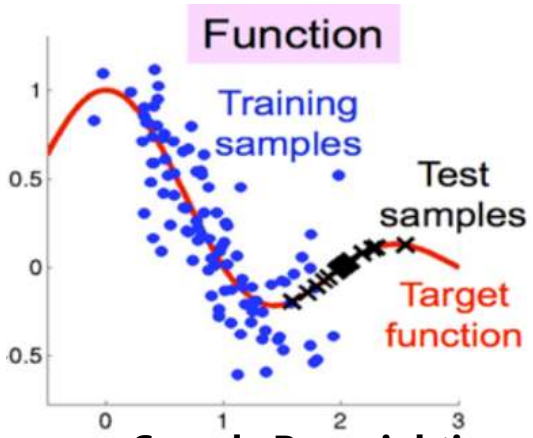
Target Domain $\sim P_T(Z, H)$

unlabeled or limited labels

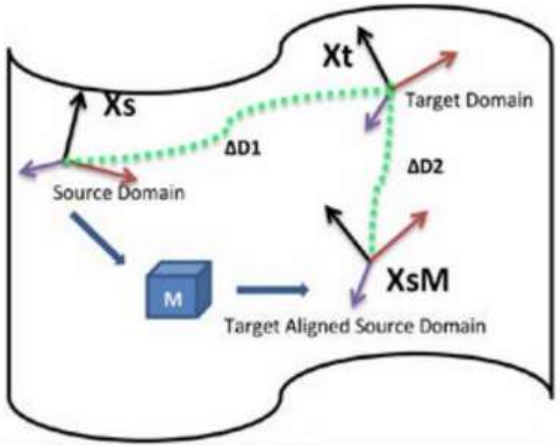
$$D_S = \{(\mathbf{x}_i, y_i), \forall i \in \{1, \ldots, N\}\}$$

$$D_T = \{(\mathbf{z}_j, ?), \forall j \in \{1, \ldots, M\}\}$$

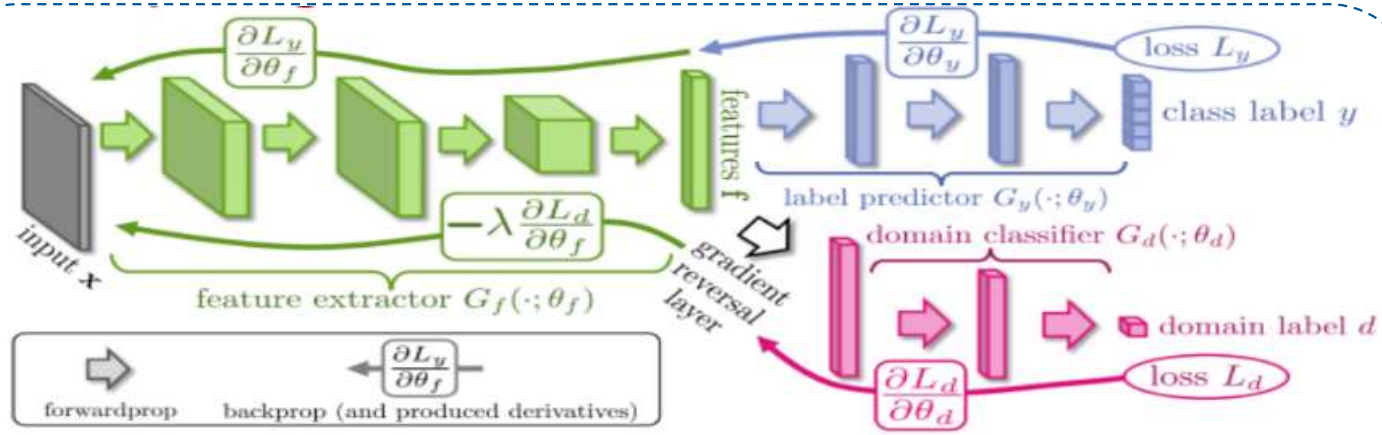
Domain Adaptation: Train on Source and Adapt to Target



Sample Re-weighting [Huang,NIPS'06]

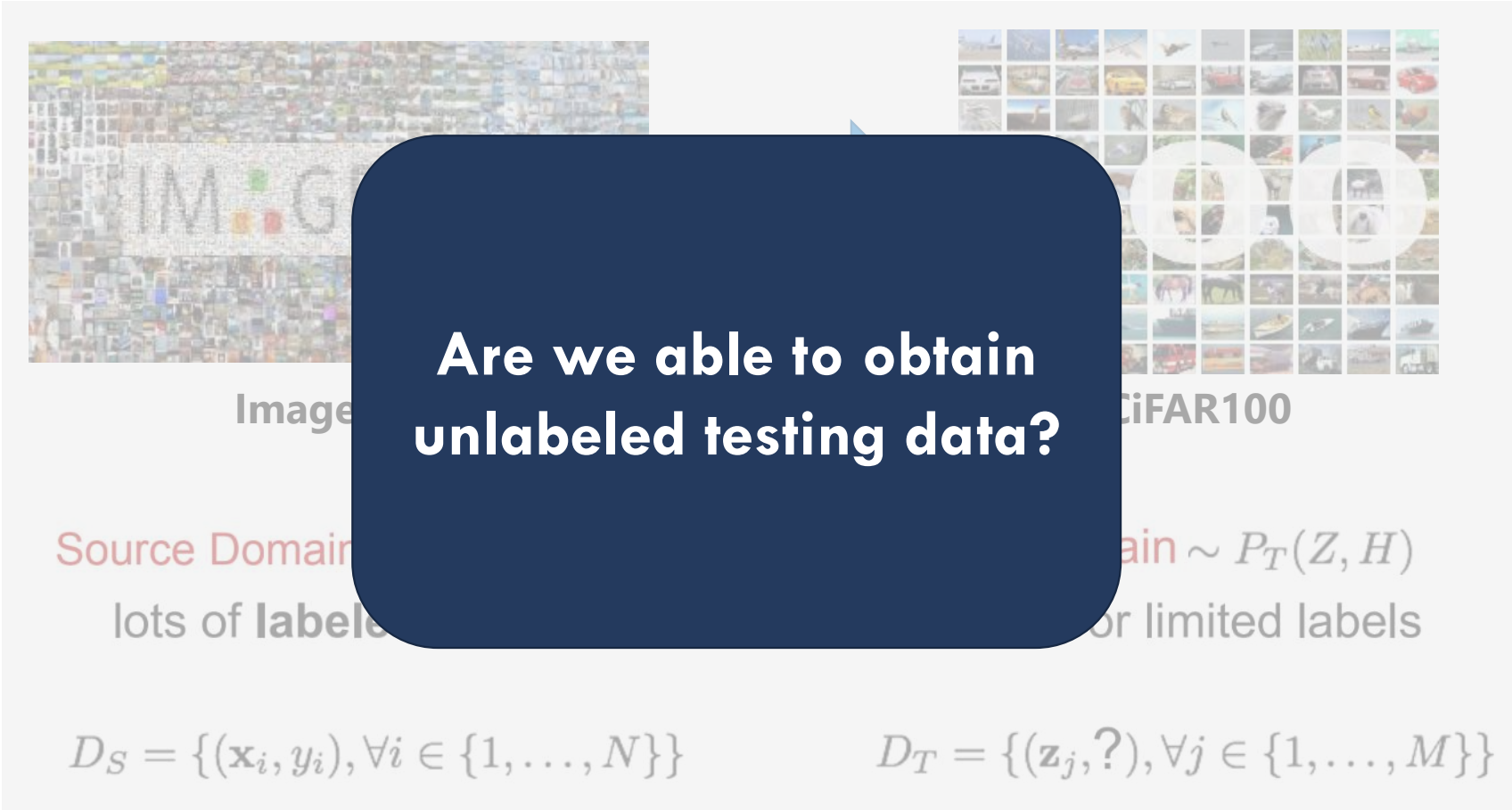


Subspace Learning [Fernando, ICCV'13]



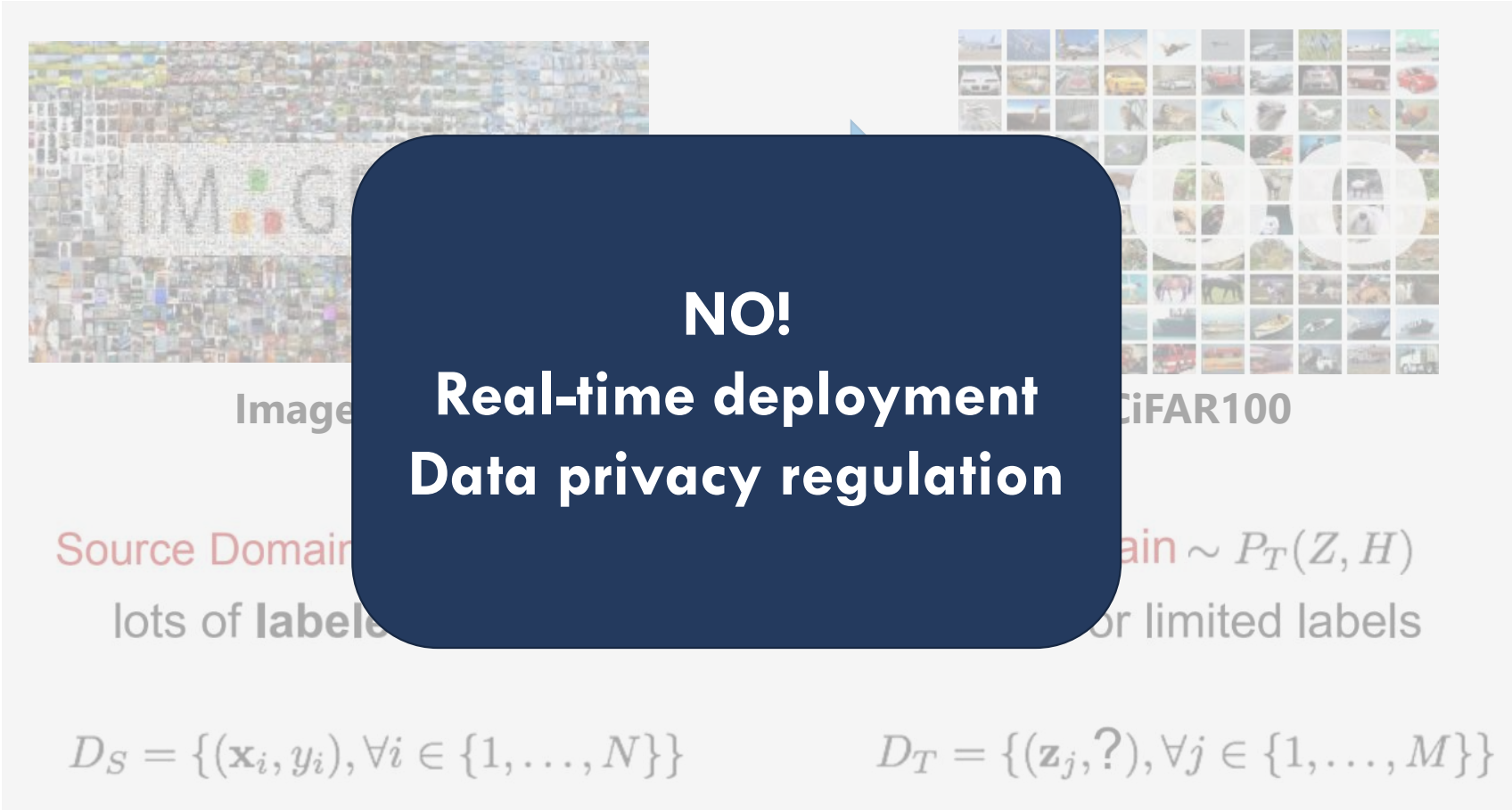
Adversarial Learning [Ganin, ICML'15, Tzeng, CVPR'17]

Domain Adaptation: Train on Source and Adapt to Target



Are we able to obtain unlabeled testing data?

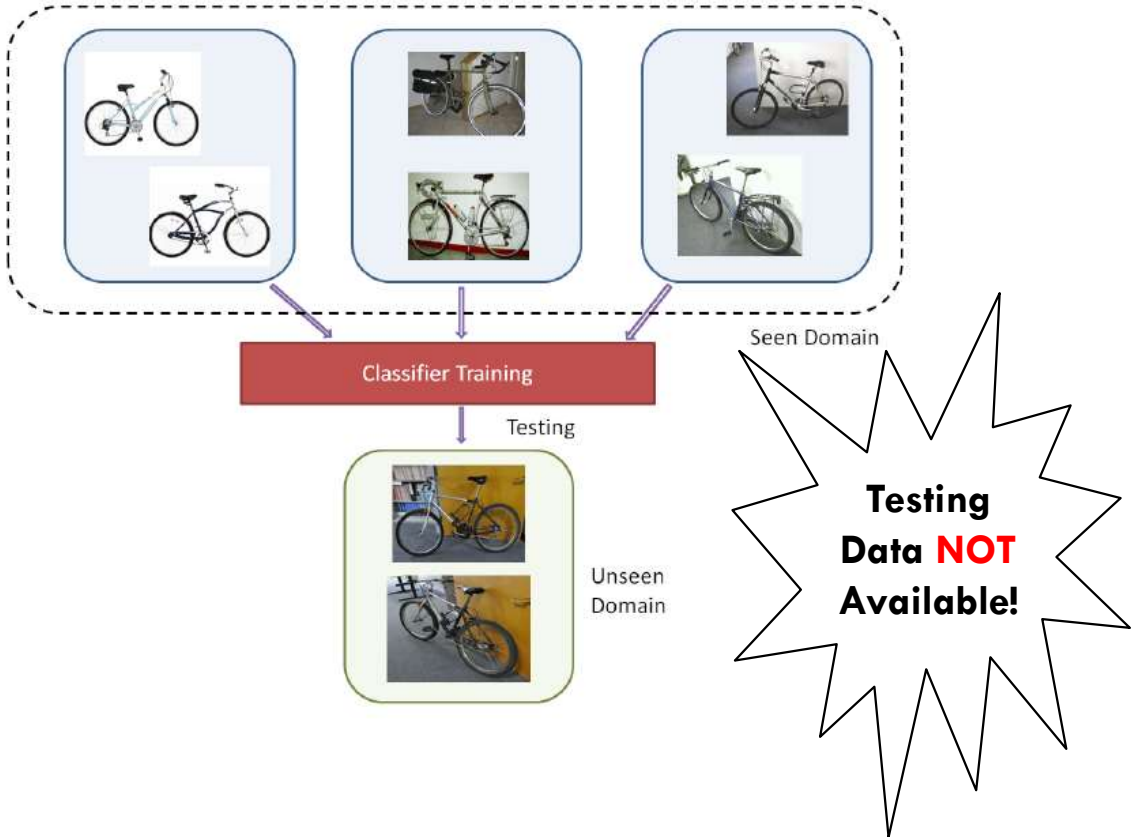
Domain Adaptation: Train on Source and Adapt to Target



NO!
Real-time deployment
Data privacy regulation

Domain Generalization

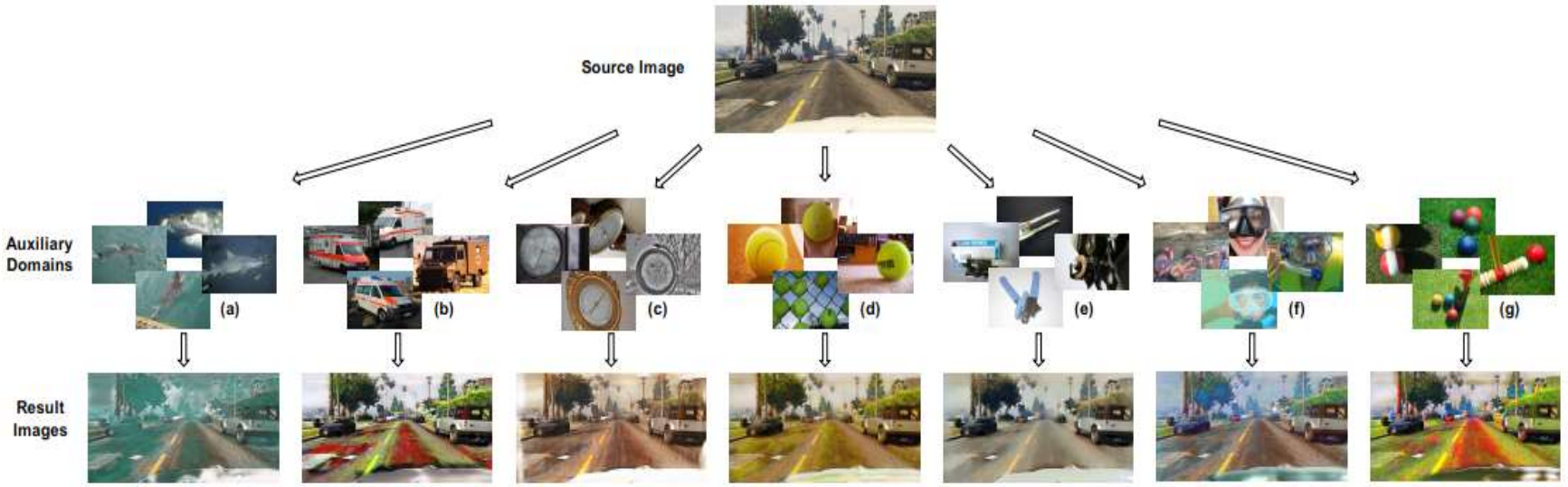
Domain Generalization (DG): Build a system for previously unseen datasets, given one or multiple training datasets.



- Data augmentation and generation
- Distribution alignment
- Meta-learning
- Contrastive Learning
- Adversarial Training
- …

Data augmentation

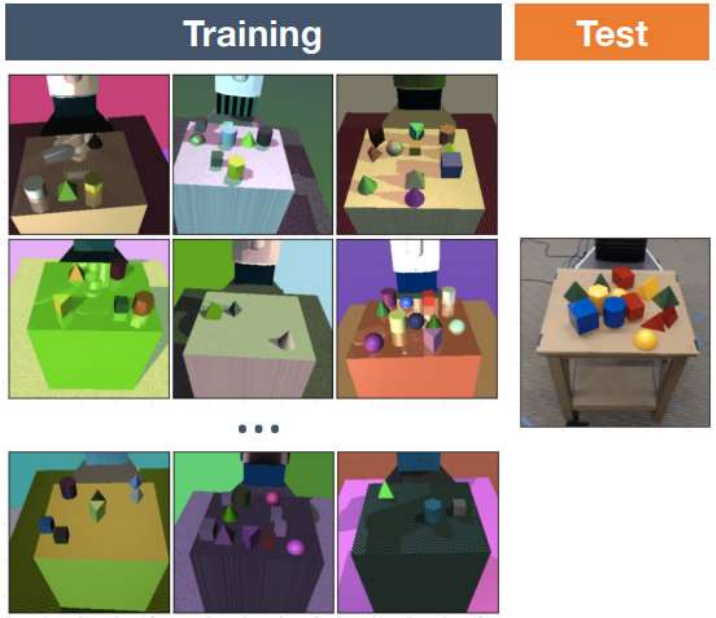
- Typical augmentation
  - Rotation, noise, color…
- Domain randomization (DR)
  - Randomly draw K real-life categories from ImageNet for stylizing the synthetic images.



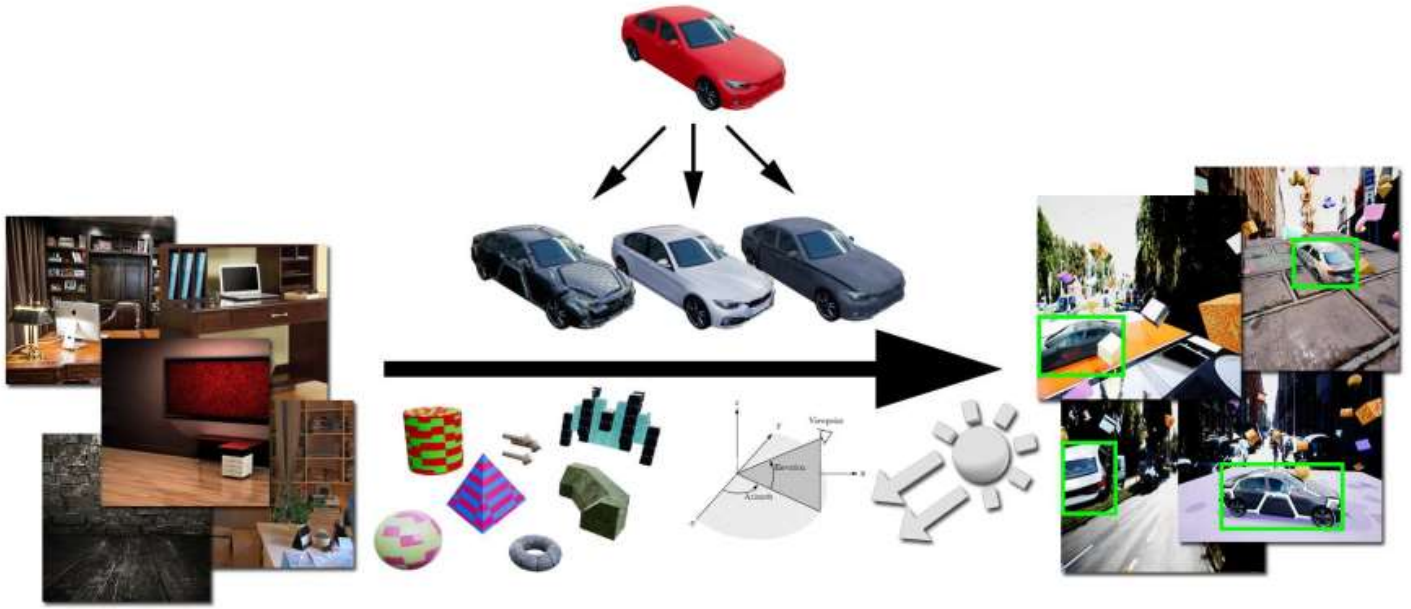
Yue et al. Domain Randomization and Pyramid Consistency: Simulation-to-Real Generalization without Accessing Target Domain Data. ICCV,2019.

Domain randomization

Domain randomization through graphics software.



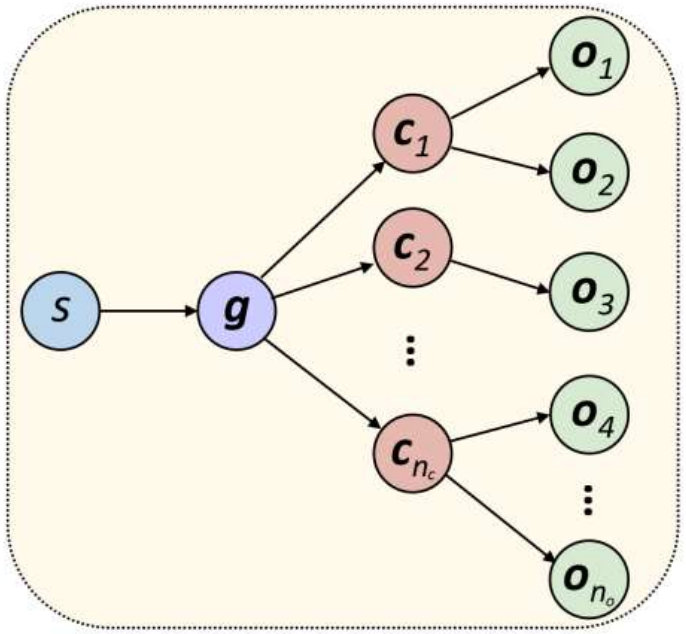
Sim->Real robot control



Synthetic images -> Real images

- Tobin, et al. Domain Randomization for Transferring Deep Neural Networks from Simulation to the Real World. IROS 2017.
- Tremblay et al. Training Deep Networks with Synthetic Data: Bridging the Reality Gap by Domain Randomization. CVPR workshop 2018.

Context-aware randomization



$$p(I, s, \mathbf{g}, \mathbf{o}_{1..n_o}, \mathbf{c}_{1..n_c}) = p(I|s, \mathbf{g}, \mathbf{o}_{1..n_o}, \mathbf{c}_{1..n_c}) \cdot \prod_{j=1}^{n_o} p(\mathbf{o}_j|\mathbf{c}_i) \prod_{i=1}^{n_c} p(\mathbf{c}_i|\mathbf{g}) p(\mathbf{g}|s) p(s)$$

Urban Suburban Rural

Prakash et al. Structured Domain Randomization: Bridging the Reality Gap by Context-Aware Synthetic Data. 2018.

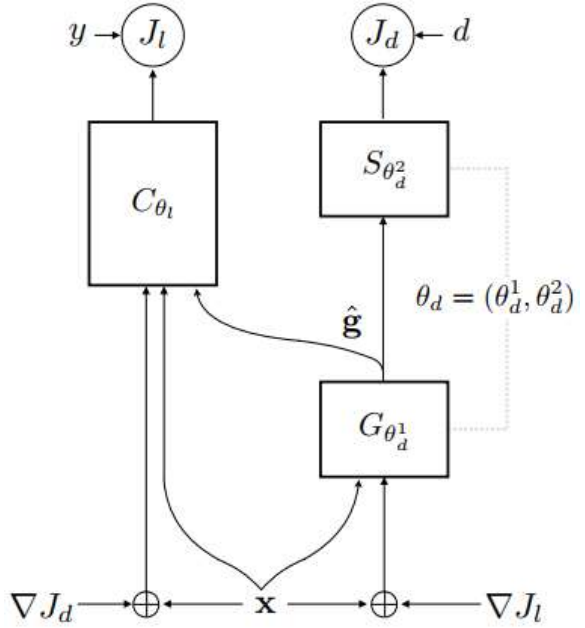
Adversarial data augmentation

- CrossGrad: Adversarially augment data via gradient training
  - Generate data that are with *same* label y, but *different* domain label d

$$\mathbf{x}_i' = \mathbf{x}_i + \epsilon \nabla_{\mathbf{x}_i} J_d(\mathbf{x}_i, d_i)$$

- ADV augmentation
  - Learning the *worse-case* distribution to enable generalization

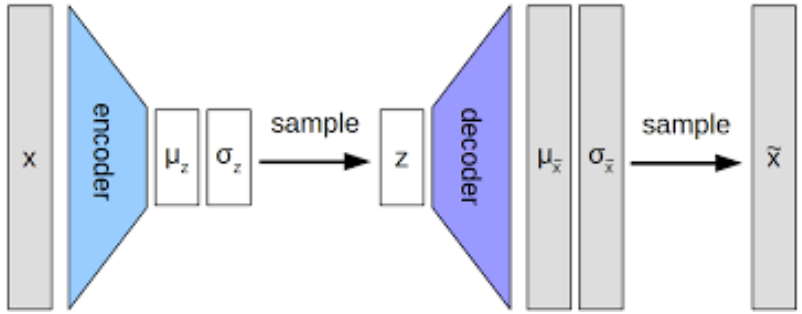
$$\underset{\theta \in \Theta}{\text{minimize}} \ \sup_{P} \{\mathbb{E}_P[\ell(\theta; (X, Y))] : D_\theta(P, P_0) \leq \rho\}$$



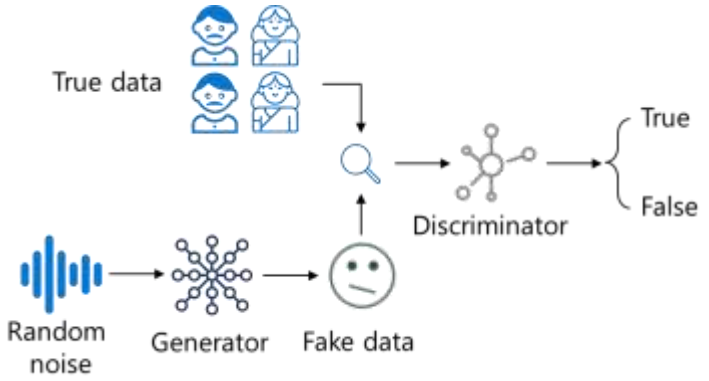
- Shankar et al. Generalizing across Domains via Cross-Gradient Training. ICLR 2018.
- Volpi, et al. Generalizing to Unseen Domains via Adversarial Data Augmentation. NeurIPS 2018.

Data generation

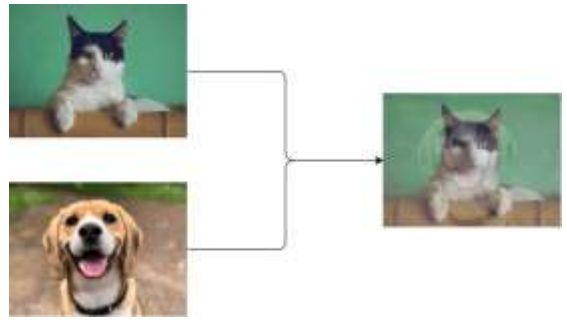
- Directly generate data
  - *Learning* to generate, instead of randomization / adversarial augmentation (Fixed scheme)



Variational auto-encoder (VAE)



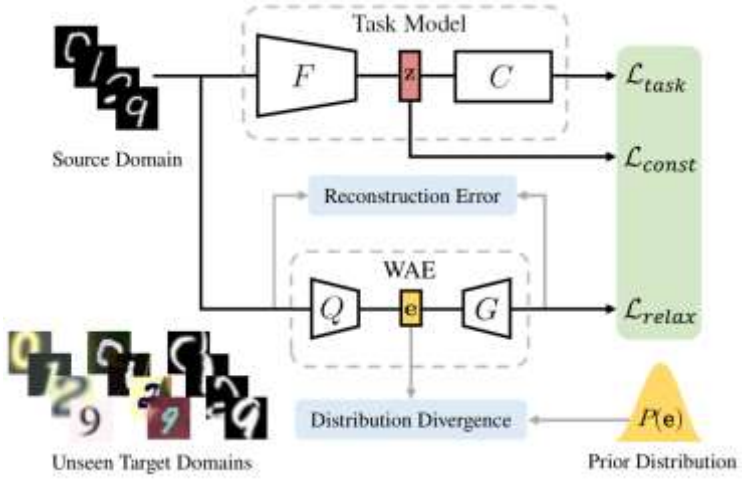
Generative adversarial net (GAN)



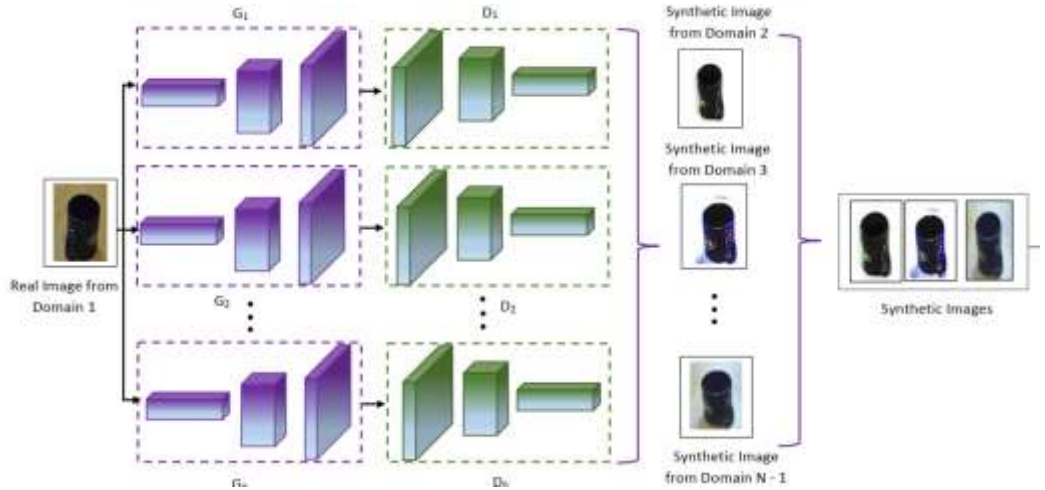
Mixup

- Kingma D P, Welling M. Auto-encoding variational bayes[J]. arXiv preprint arXiv:1312.6114, 2013.
- Goodfellow I, Pouget-Abadie J, Mirza M, et al. Generative adversarial nets[J]. Advances in neural information processing systems, 2014, 27.
- Zhang H, Cisse M, Dauphin Y N, et al. Mixup: Beyond empirical risk minimization[J]. arXiv preprint arXiv:1710.09412, 2017.

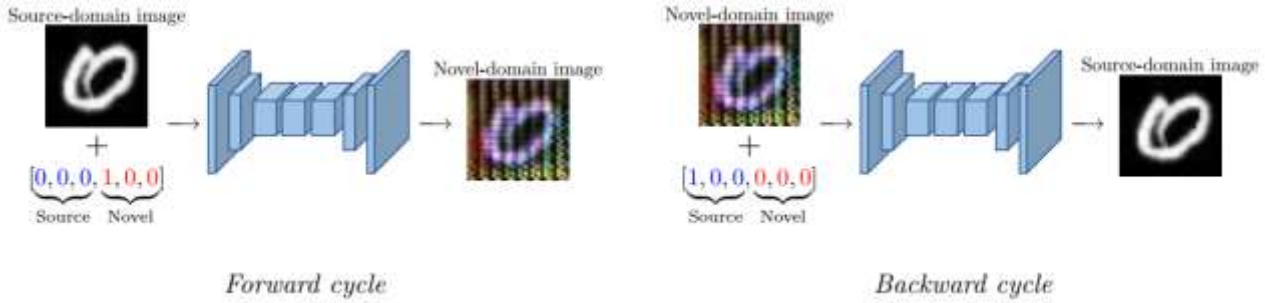
Data generation



VAE for generation



Multi-component generation



Conditional GAN for generation

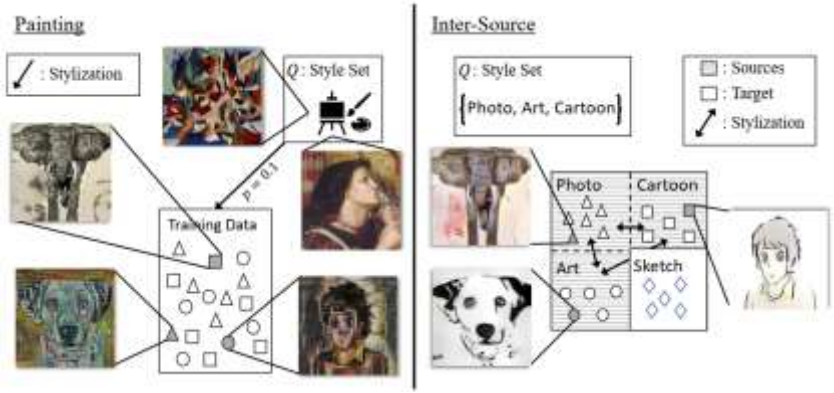
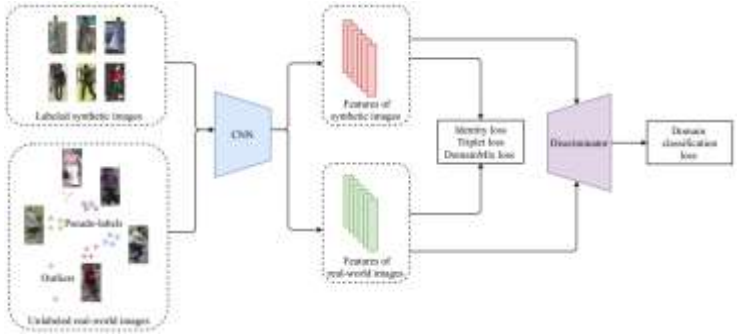


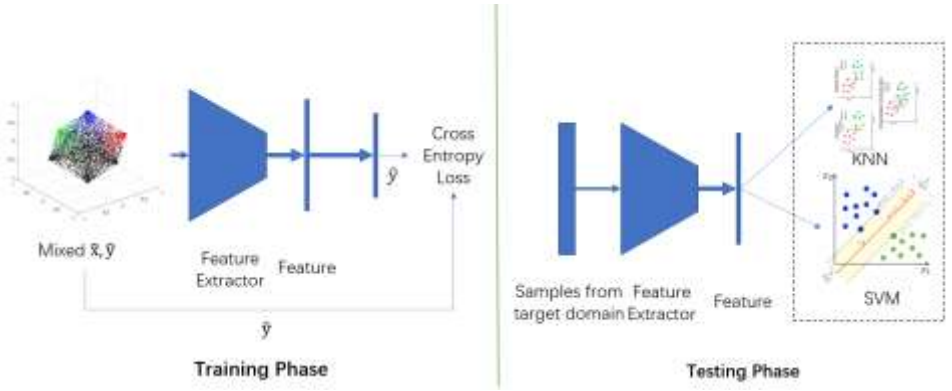
Image stylization

- Qiao et al. Learning to Learn Single Domain Generalization. CVPR 2020.
- Rahman et al. Multi-component Image Translation for Deep Domain Generalization. 2020.
- Zhou et al. Learning to Generate Novel Domains for Domain Generalization. ECCV 2020.
- Somavarapu et al. Frustratingly Simple Domain Generalization via Image Stylization. 2020.

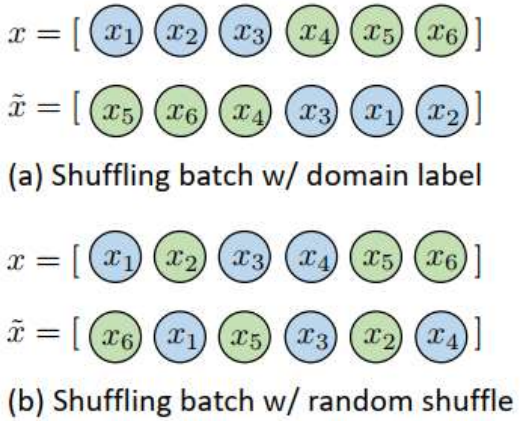
Mixup



DomainMix



MixAll

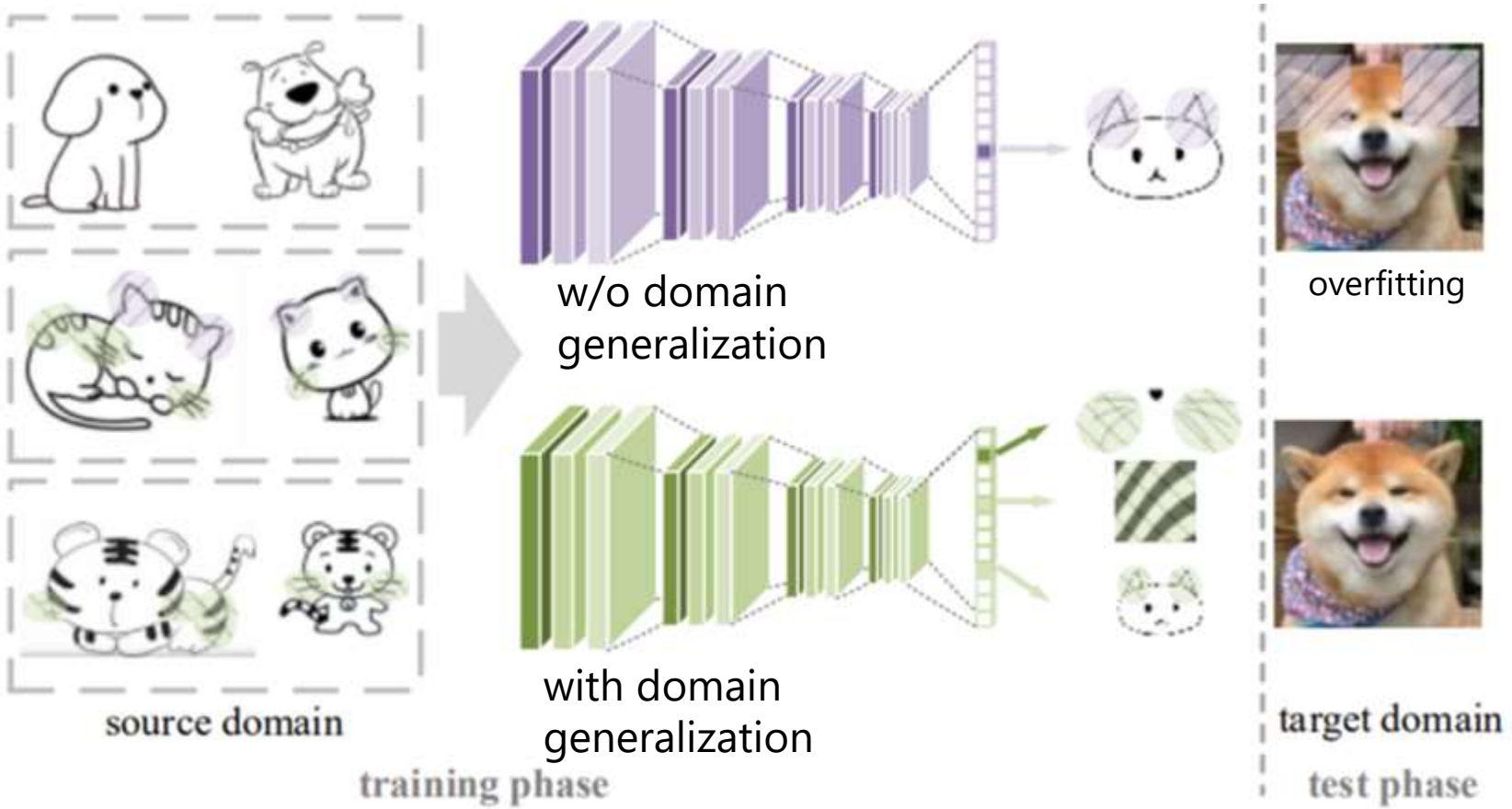


Style mixup

- Wang et al. DomainMix: Learning Generalizable Person Re-Identification Without Human Annotations. 2020.
- Wang et al. Heterogeneous domain generalization via domain mixup. ICASSP 2020.
- Zhou et al. Domain generalization with mixstyle. ICLR 2021.

Representation Learning

- Learning features which are expected to be better generalized to unseen target domain.



Kernel-based methods

- Using kernel methods to learn domain-invariant features
  - DICA: domain-invariant component analysis

$$\widehat{\mathbb{V}}_{\mathcal{H}}(\mathcal{BS}) = \mathrm{tr}(\widetilde{K}Q) = \mathrm{tr}(B^{\top}KQKB)$$

  - TCA: Transfer Component Analysis

$$\min_{W} tr(W^T KLKW) + \mu\, tr(W^T W),\ \text{s.t. } W^T KHKW = I.$$

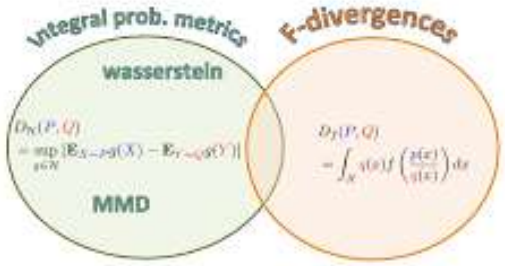
  - SCA: Scatter Component Analysis

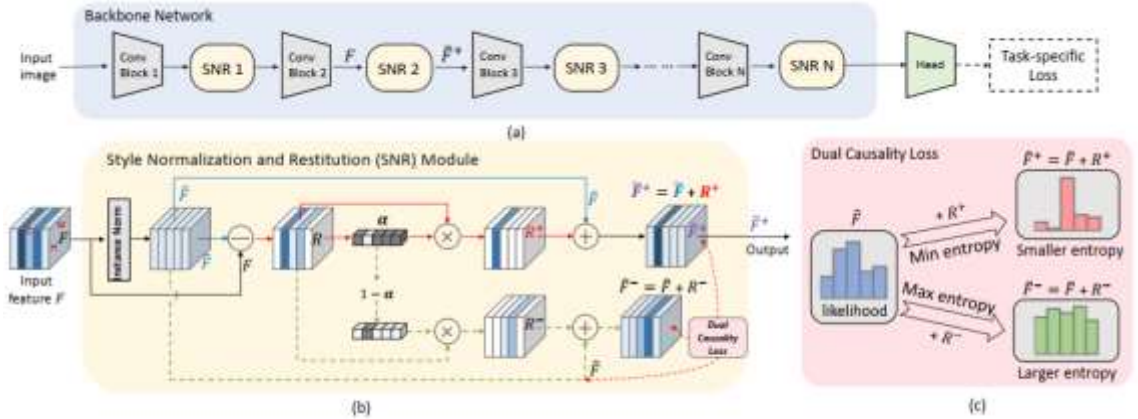
$$\Psi_{\phi}(\mathbb{P}) := \mathop{\mathbb{E}}_{x\sim\mathbb{P}}\left[\left\|\mu_{\mathbb{P}} - \phi(x)\right\|_{\mathcal{H}}^2\right] \qquad \sup \frac{\{\text{total scatter}\} + \{\text{between-class scatter}\}}{\{\text{domain scatter}\} + \{\text{within-class scatter}\}}$$

- Blanchard et al. Generalizing from Several Related Classification Tasks to a New Unlabeled Sample. NeurIPS 2011.
- Muandet et al. Domain Generalization via Invariant Feature Representation. ICML 2013.
- Grubinger et al. Domain Generalization Based on Transfer Component Analysis. IWANN 2015.
- Ghifary et al. Scatter Component Analysis: A Unified Framework for Domain Adaptation and Domain Generalization. TPAMI 2017.

Explicit feature alignment

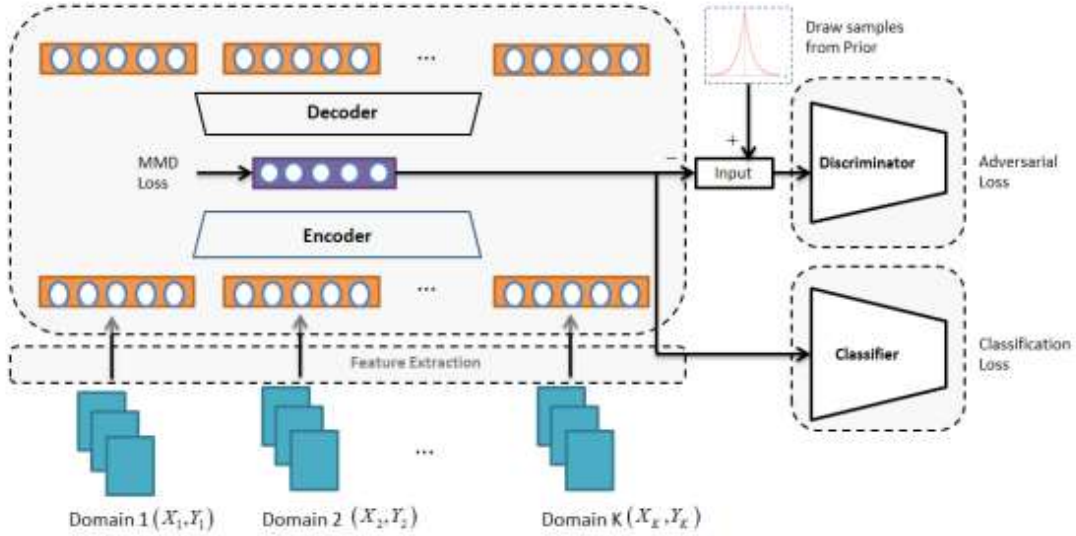
- Learning shareable information across domain
  - Maximum mean discrepancy: $\mathrm{MMD}(\mathcal{F}, P_X, P_Y) = \sup_{\|f\|_{\mathcal{H}} \leq 1} (\mathbb{E}_p(f(x)) - \mathbb{E}_p(f(y)))$
  - KL Divergence: $KL(q(\mathcal{Z}|\mathcal{X})||\mathcal{N} \sim (0,1))$
  - Correlation alignment: $\ell_{CORAL} = \frac{1}{4d^2}\|C_S - C_T\|_F^2$





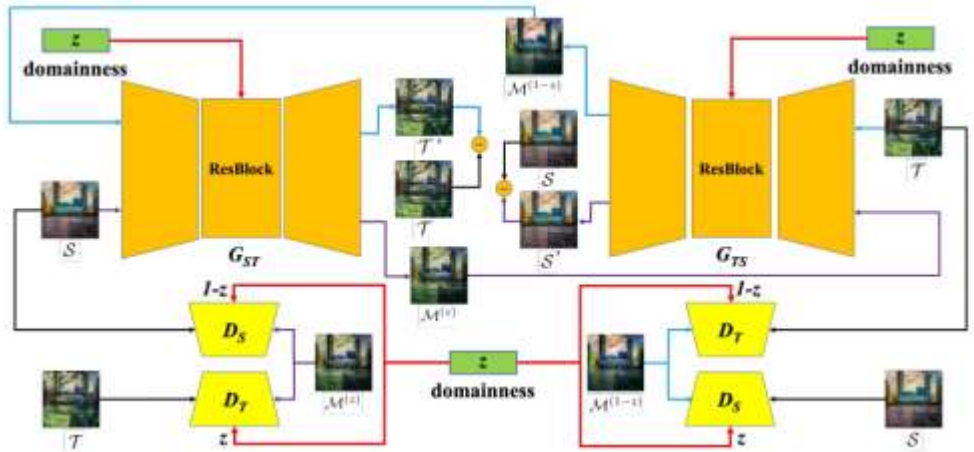
Ya Li, et al., Deep domain generalization via conditional invariant adversarial networks, ECCV 2018
Haoliang Li, et al., Domain Generalization for Medical Imaging Classification with Linear-Dependency, NeurIPS, 2020
Jin X, Lan C, Zeng W, et al. Style Normalization and Restitution for Domain Generalization and Adaptation, Arxiv, 2021.

Domain adversarial learning



MMD-AAE

$$\min_{Q,P} \max_{D} \mathcal{L}_{\text{ae}} + \lambda_1 \mathcal{R}_{\text{mmd}} + \lambda_2 \mathcal{J}_{\text{gan}}$$



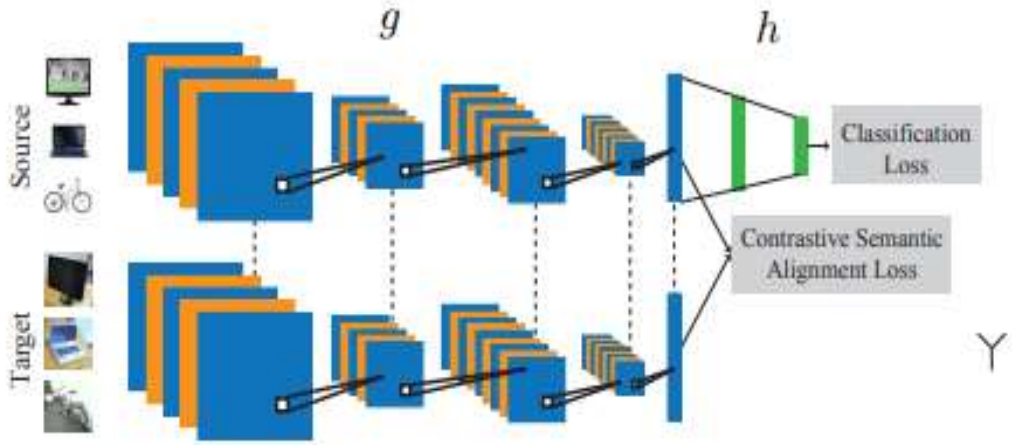
DLOW

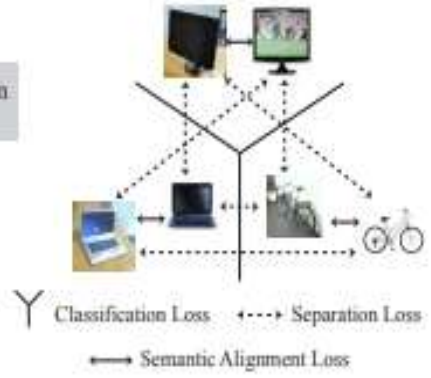
$$\begin{aligned}
\mathcal{L}_{adv}(\ G_{ST}\ , D_S) &= \mathbb{E}_{\mathbf{x}^s \sim P_S}\left[\log(D_S(\mathbf{x}^s))\right] \\
&+ \ \mathbb{E}_{\mathbf{x}^s \sim P_S}\left[\log(1 - D_S(G_{ST}(\mathbf{x}^s, z)))\right] \\
\mathcal{L}_{adv}(\ G_{ST}\ , D_T) &= \mathbb{E}_{\mathbf{x}^t \sim P_T}\left[\log(D_T(\mathbf{x}^t))\right] \\
&+ \ \mathbb{E}_{\mathbf{x}^s \sim P_S}\left[\log(1 - D_T(G_{ST}(\mathbf{x}^s, z)))\right].
\end{aligned}$$

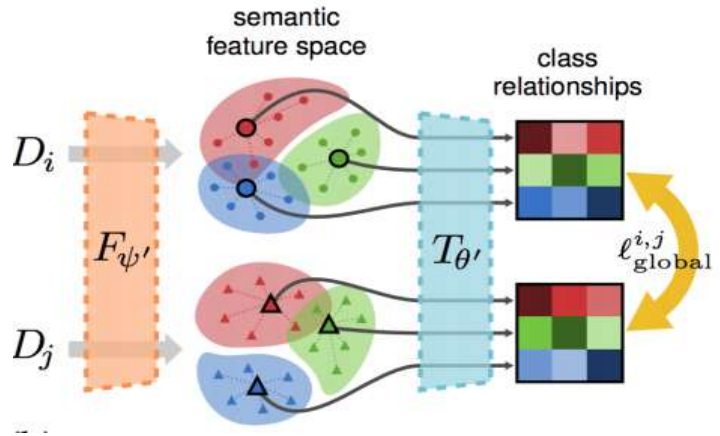
- Haoliang Li et al. Domain Generalization with Adversarial Feature Learning. CVPR 2018.
- Rui Gong et al. DLOW: Domain Flow for Adaptation and Generalization. CVPR 2019.

Contrastive Learning

Minimizing/Maximizing feature distance among samples from with same/different category information from different domains



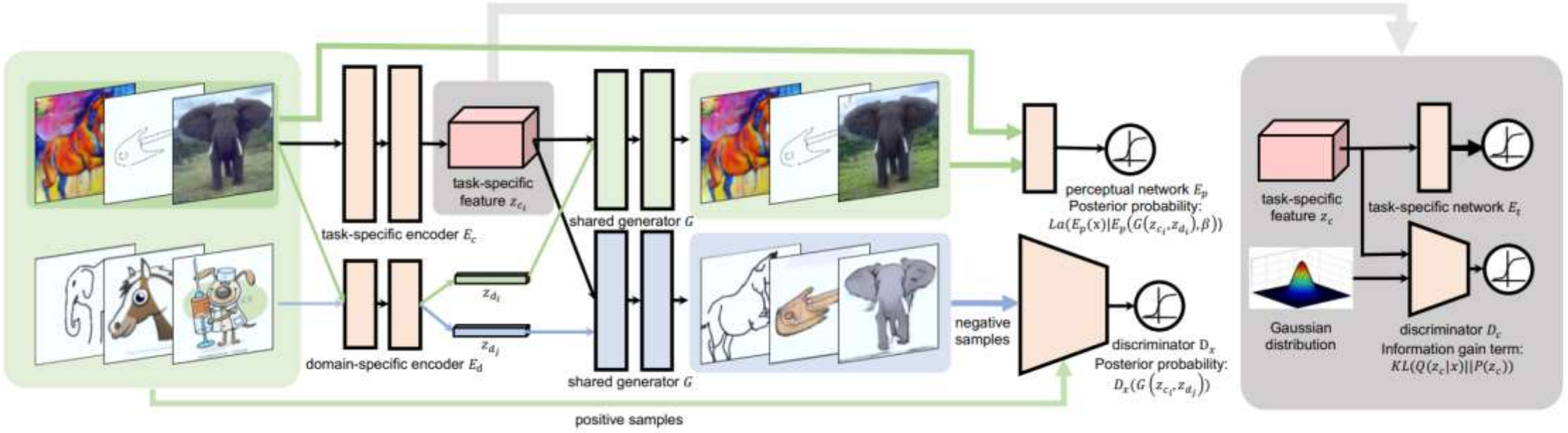




Motiian, et al., Unified Deep Supervised Domain Adaptation and Generalization, ICCV'17
Dou, et al., Domain Generalization via Model-Agnostic Learning of Semantic Features, NeurIPS'19

Feature disentanglement

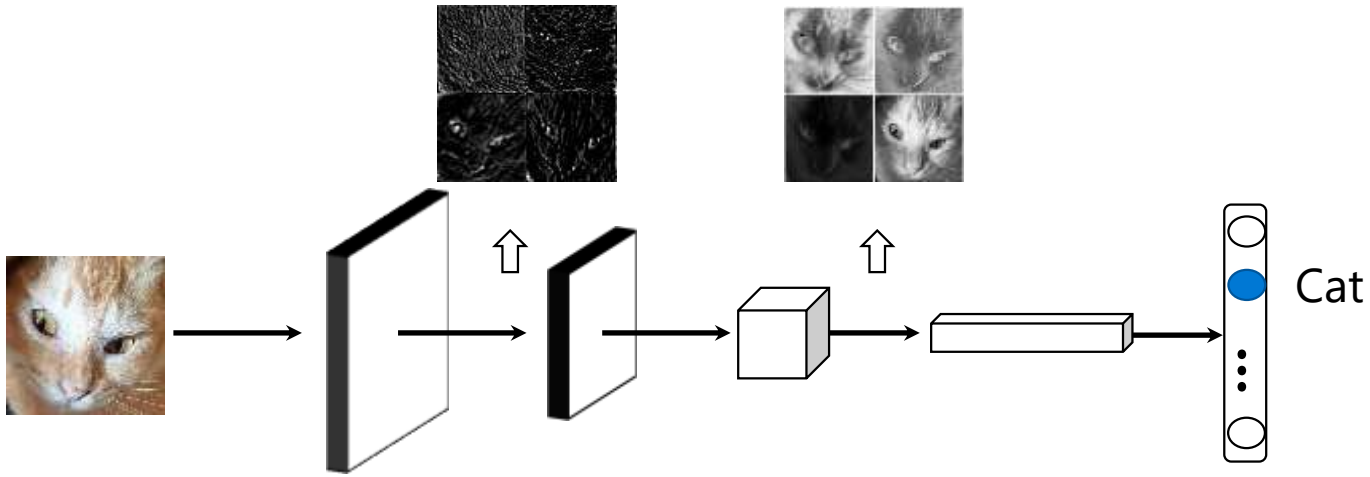
Invariant feature learning + style transfer



Yufei Wang, et al., Variational Disentanglement for Domain Generalization, Arxiv 2021

Multi-layer Feature Learning

- Deep features eventually transit from general to specific along the network.
- Shallow Layer extracts shareable information while deep layer extracts category specific information (with regularization).

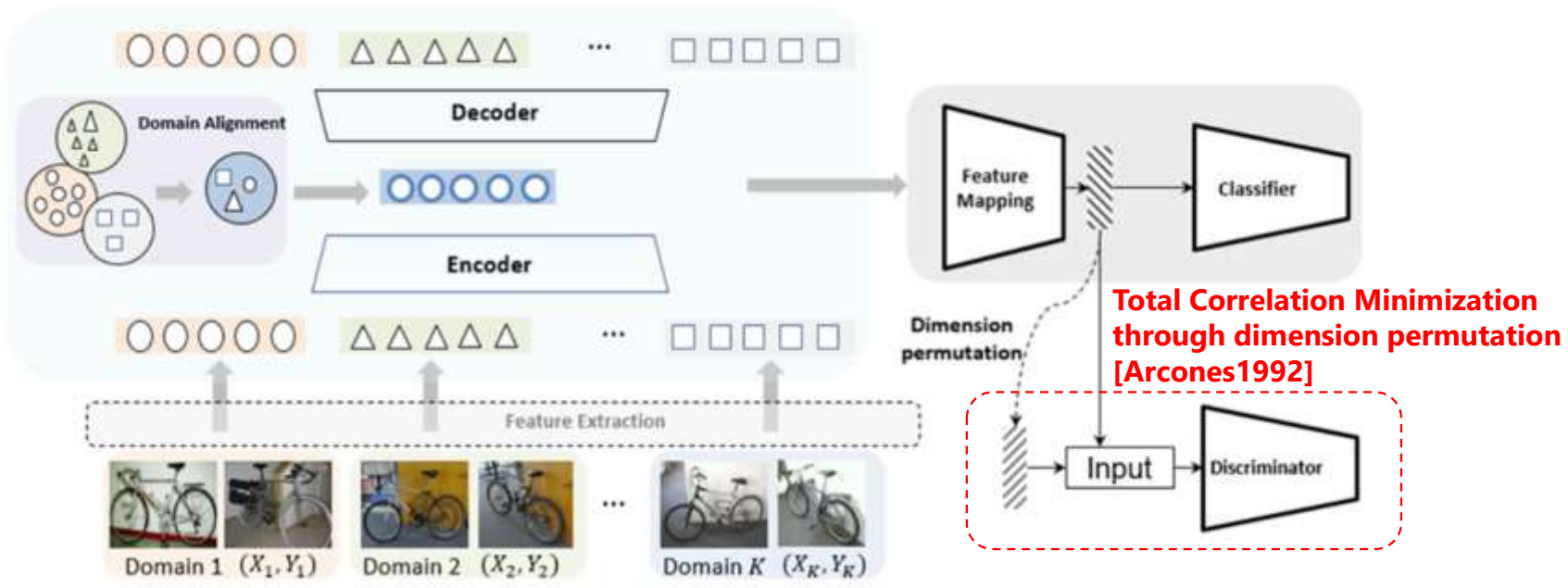


- Haoliang Li, *et.al.*, "GMFAD: Towards Generalized Visual Recognition via Multi-Layer Feature Alignment and Disentanglement", T-PAMI 2020

Multi-layer Feature Learning

- Feature disentanglement at deep layer.
  - Neuron independence regularization

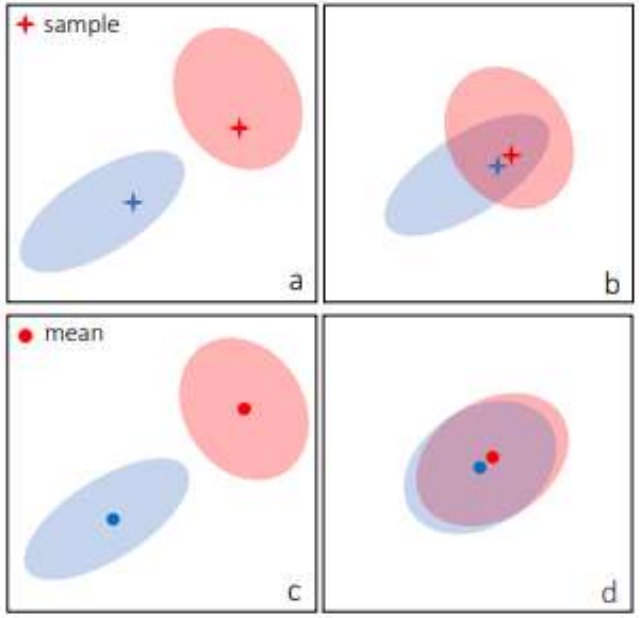
$$P\left(H^1, H^2, \ldots, H^{d'}\right) = P(H^1)P(H^2)\ldots P(H^d)$$



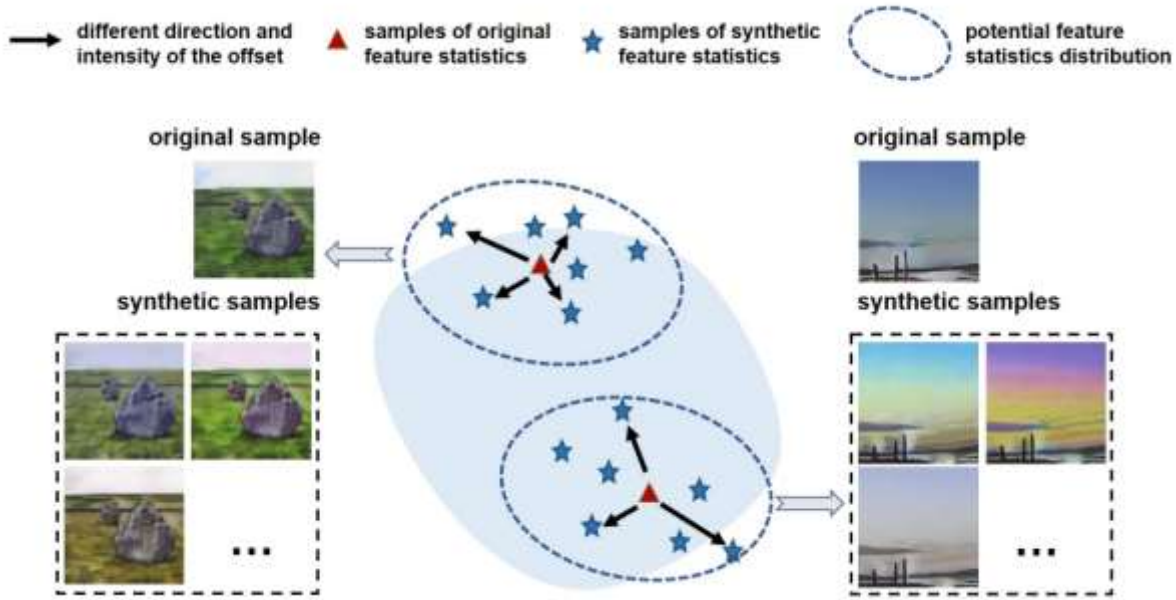
[Arcones1992] M. A. Arcones and E. Gine, "On the bootstrap of u and v statistics," The Annals of Statistics, pp. 655–674, 1992.

Domain-Invariant Learning with Uncertainty

- Uncertainty should be taken into account during domain-invariant learning.



Bayesian Neural Network



Uncertainty modeling through re-parameterization trick

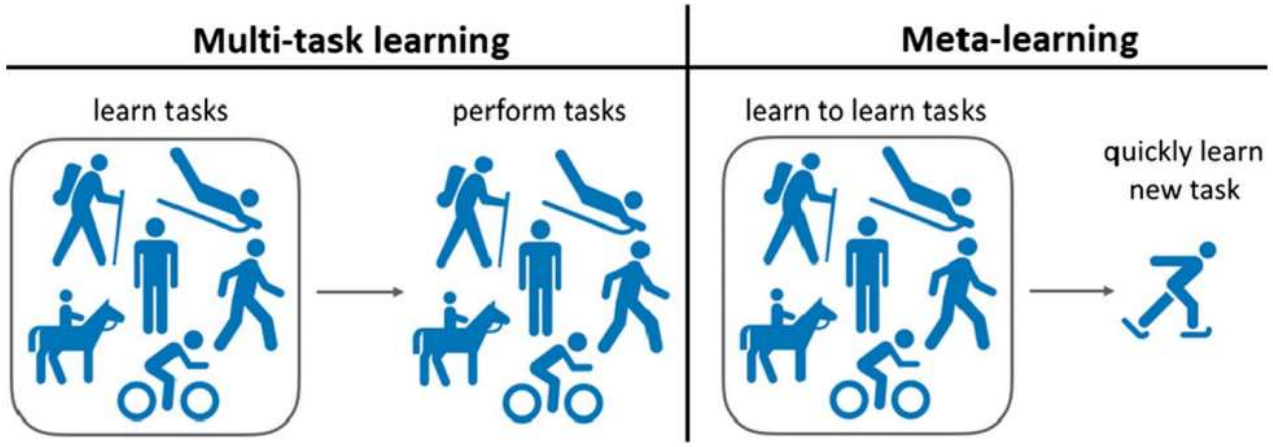
Zehan Xiao, et al., A Bit More Bayesian: Domain-Invariant Learning with Uncertainty , ICML'21
Xiaotong Li, et al., Uncertainty Modeling for Out-of-Distribution Generalization." ICLR'22.

Different learning strategy for DG

- Meta-learning
  - Divide domains into several tasks, then use meta-learning to learn general knowledge
- Ensemble learning
  - Design ensemble models
- Gradient operation
  - Alter the gradient interaction between domains
- Distributionally robust optimization
  - Acquire models that are better for worst-case distribution scenario
- Self-supervised learning
- Others

Meta-learning

- Learning to learn, or meta-learn the general knowledge
  - Instead of the original tasks, meta-learning wants to acquire knowledge about **new tasks**



Meta-knowledge acquisition

$$\phi^* = \arg\max_{\phi} \log P\left(\phi|\mathcal{D}_{\text{src}}\right)$$

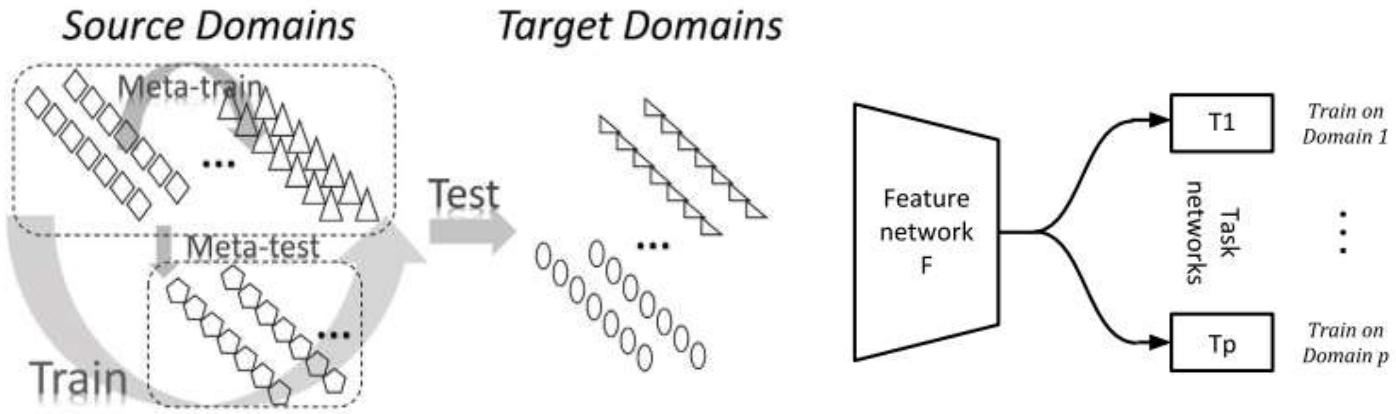
Meta-knowledge validation

$$\theta^{*(i)} = \arg\max_{\theta} \log P\left(\theta|\phi^*, \mathcal{D}_{\text{tar}}^{\text{train}\ (i)}\right)$$

Huisman M, Van Rijn J N, Plaat A. A survey of deep meta-learning[J]. Artificial Intelligence Review, 2021, 54(6): 4483-4541.

Meta-learning for DG

- How to adopt meta-learning for DG?
  - Key: Old tasks to new tasks in meta-learning → Old domains to new domains
- MLDG: Meta-learning for DG
- MetaReg: meta-learning for regularization



Algorithm 1 Meta-Learning Domain Generalization

procedure MLDG
  Input: Domains $\mathcal{S}$
  Init: Model parameters Θ. Hyperparameters α, β, γ.
  for ite **in** iterations **do**
    Split: $\bar{\mathcal{S}}$ and $\breve{\mathcal{S}} \leftarrow \mathcal{S}$
    Meta-train: Gradients $\nabla_\Theta = \mathcal{F}'_\Theta(\bar{\mathcal{S}}; \Theta)$
    Updated parameters $\Theta' = \Theta - \alpha \nabla_\Theta$
    Meta-test: Loss is $\mathcal{G}(\breve{\mathcal{S}}; \Theta')$.
    Meta-optimization: Update Θ

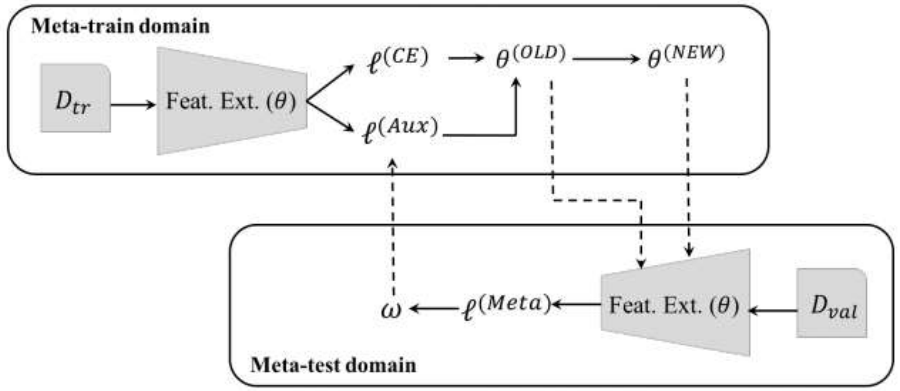
$$\Theta = \Theta - \gamma \frac{\partial(\mathcal{F}(\bar{\mathcal{S}}; \Theta) + \beta \mathcal{G}(\breve{\mathcal{S}}; \Theta - \alpha \nabla_\Theta))}{\partial \Theta}$$

  end for
end procedure

- Li D, Yang Y, Song Y Z, et al. Learning to generalize: Meta-learning for domain generalization. AAAI 2018.
- Balaji Y, Sankaranarayanan S, Chellappa R. Metareg: Towards domain generalization using meta-regularization. NeurIPS 2018.

Meta-learning for DG

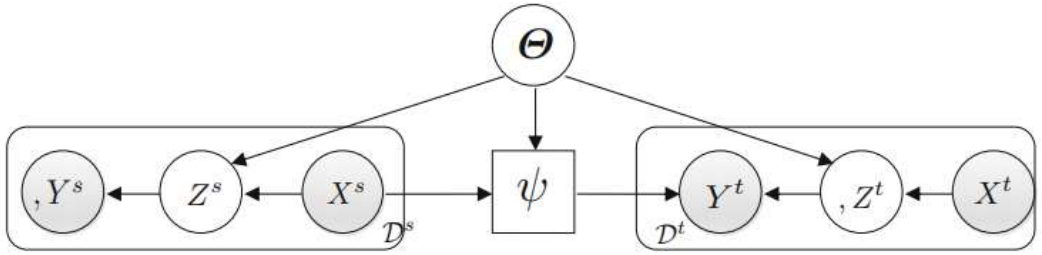
- Feature-critic training
  - Learning the regularization terms using meta-learning



$$\min_{\theta,\phi_j s} \sum_{D_j \in \mathcal{D}_{\text{trn}}} \sum_{d_j \in D_j} \ell^{(\text{CE})}(g_{\phi_j}(f_\theta(x^{(j)})), y^{(j)}) + \ell^{(\text{Aux})}$$

$$\max_{\omega} \sum_{D_j \in \mathcal{D}_{\text{val}}} \sum_{d_j \in D_j} \tanh(\gamma(\theta^{(\text{NEW})}, \phi_j, x^{(j)}, y^{(j)}) - \gamma(\theta^{(\text{OLD})}, \phi_j, x^{(j)}, y^{(j)}))$$

- Meta-VIB
  - Meta variational information bottleneck to model uncertainty between domain shifts



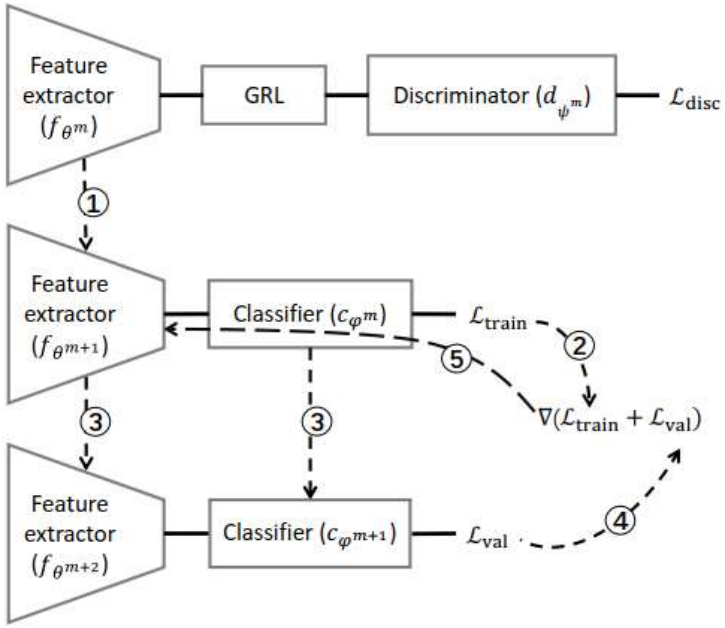
$$\tilde{\mathcal{L}}_{\text{MetaVIB}} = \frac{1}{N} \sum_{n=1}^{N} \int [p(\mathbf{z}_n|\mathbf{x}_n) p(\psi|D^s) \log q(\mathbf{y}_n|\mathbf{z}_n, \psi) - \beta\, p(\mathbf{z}_n|\mathbf{x}_n) \log \frac{p(\mathbf{z}_n|\mathbf{x}_n)}{q(\mathbf{z}_n|D^s)}] \mathrm{d}\mathbf{z}_n \mathrm{d}\psi.$$

Li Y, Yang Y, Zhou W, et al. Feature-critic networks for heterogeneous domain generalization. ICML 2019.

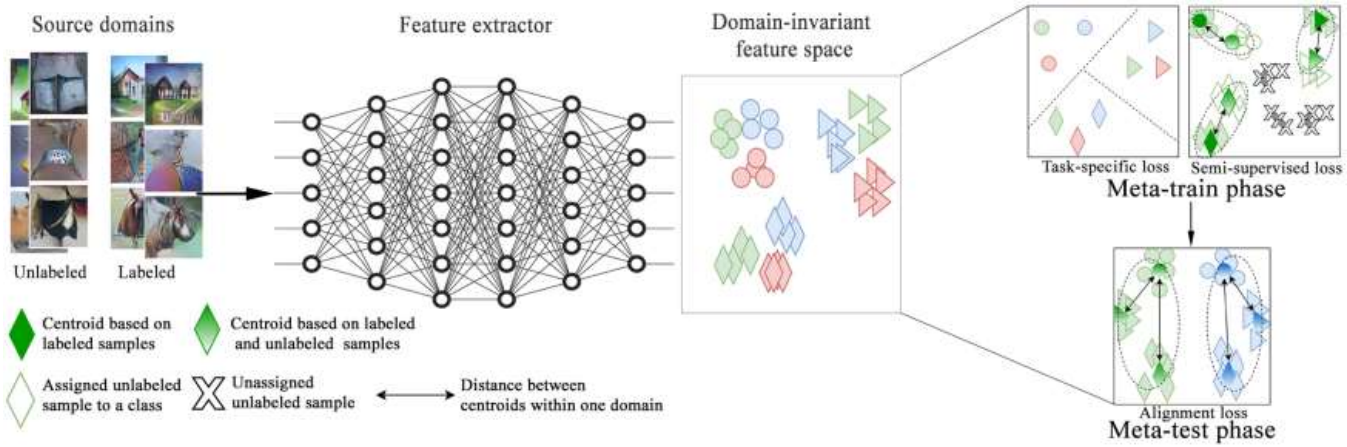
Du Y, Xu J, Xiong H, et al. Learning to learn with variational information bottleneck for domain generalization. ECCV 2020.

Meta-learning for DG

- DADG: MLDG with adversarial training



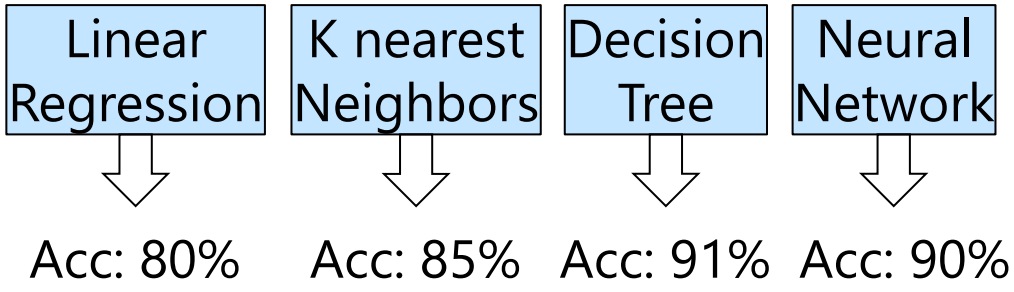
- DGSML: MLDG with semi-supervised learning

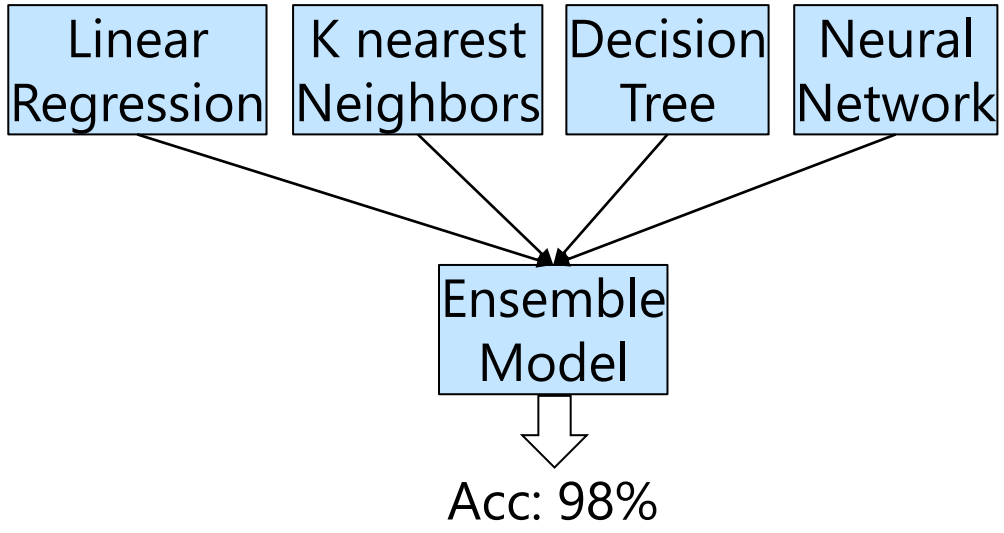


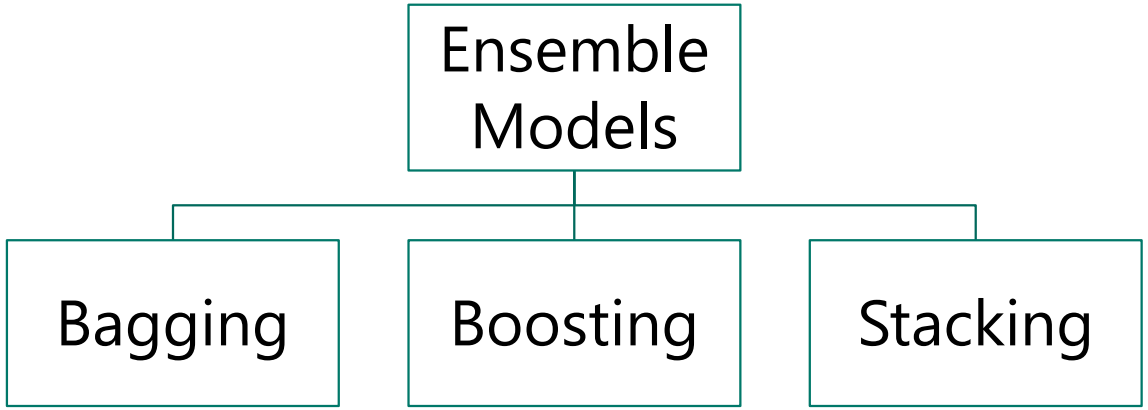
- Chen K, Zhuang D, Chang J M. Discriminative adversarial domain generalization with meta-learning based cross-domain validation. Neurocomputing 2022.
- Sharifi-Noghabi H, Asghari H, Mehrasa N, et al. Domain generalization via semi-supervised meta learning[J]. arXiv preprint arXiv:2009.12658, 2020.

Ensemble learning

- Is a single model or representation enough for generalization?



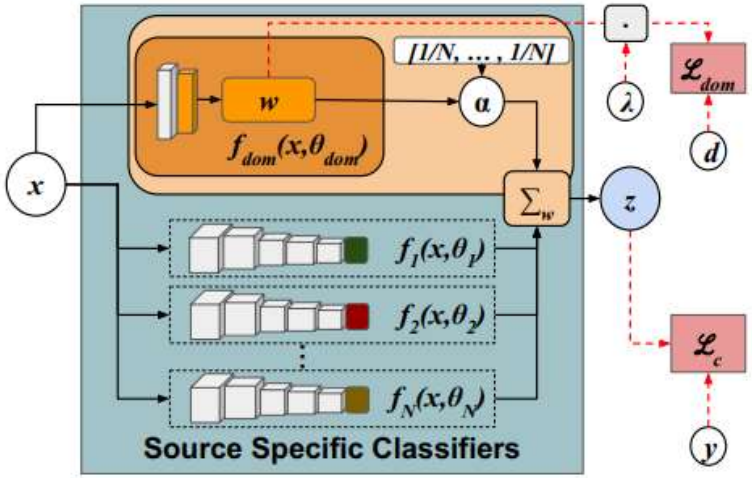




- Ensemble learning allows for more diversities in feature and classifier learning
- *The power from the crowd*

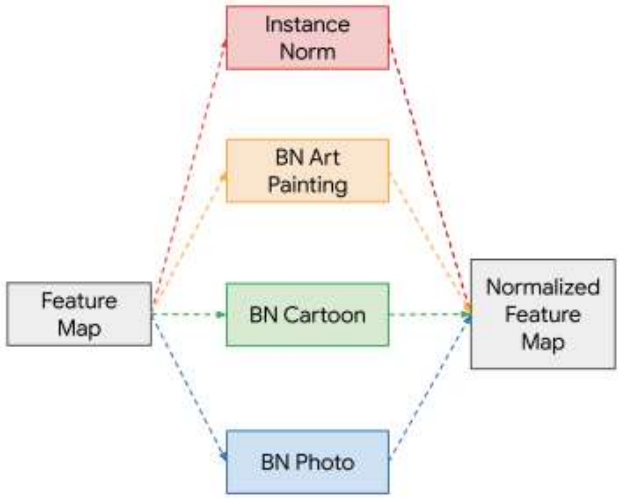
Ensemble learning for DG

- Ensemble-learned DG representations
  - Feature weighting

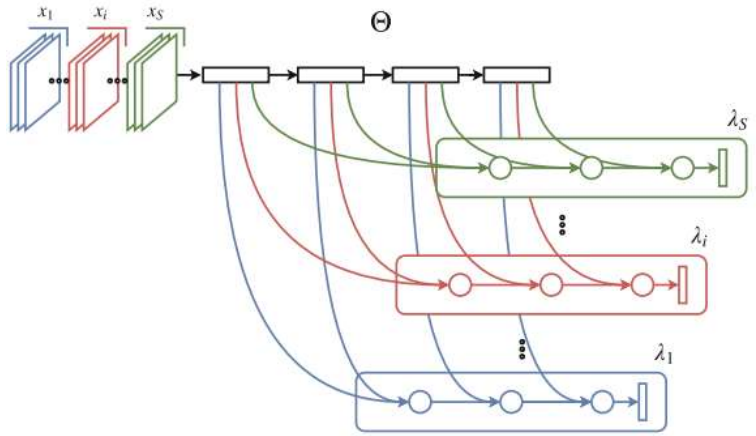


$$z_i = f(x_i, \Theta) = \sum_{j=1}^{N} w_{i,j} f_j(x_i, \theta_j)$$

Feature combination



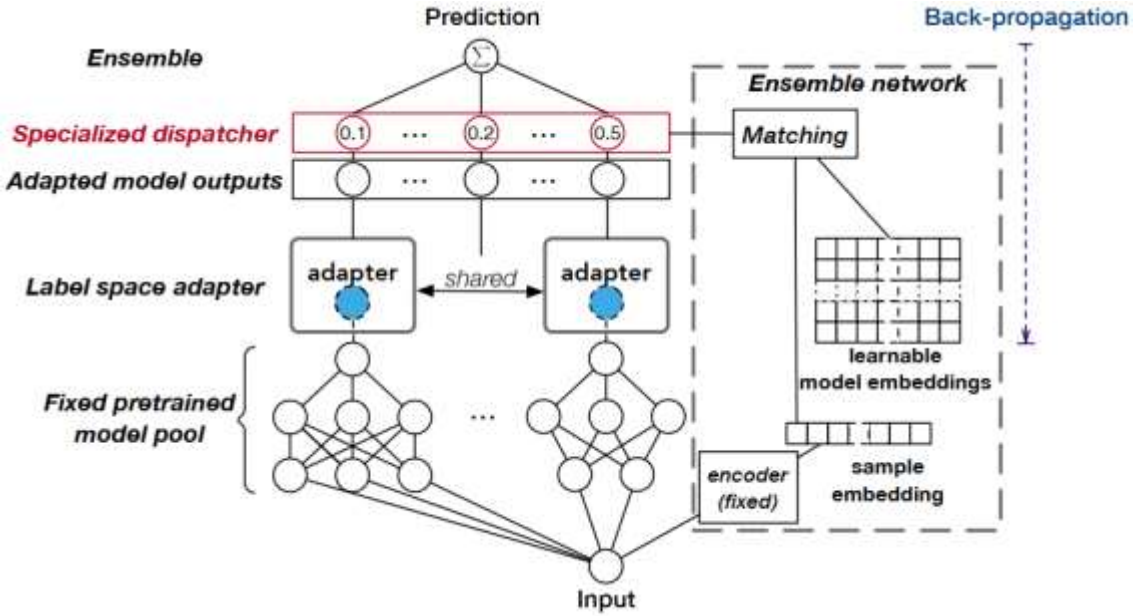
Feature attention



- Mancini M, Bulo S R, Caputo B, et al. Best sources forward: domain generalization through source-specific nets. ICIP 2018.
- Segu M, Tonioni A, Tombari F. Batch normalization embeddings for deep domain generalization[J]. arXiv preprint arXiv:2011.12672, 2020.
- D'Innocente A, Caputo B. Domain generalization with domain-specific aggregation modules[C]//German Conference on Pattern Recognition. Springer, Cham, 2018: 187-198.

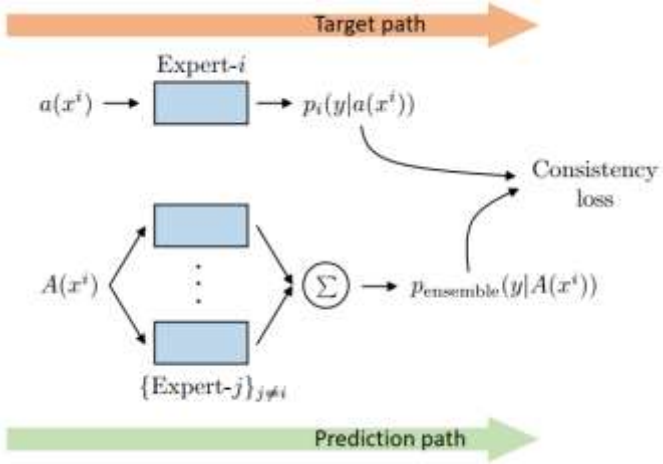
Ensemble learning for DG

- Ensemble learning for classifier learning
  - SEDGE: ensemble of pre-trained models for classifier learning



$$w_k = \frac{e^{(\zeta(\mathbf{W}(\mathbf{s})))_k}}{\sum_{j=1}^{K} e^{(\zeta(\mathbf{W}(\mathbf{s})))_j}}$$

DAEL: domain adaptive ensemble learning

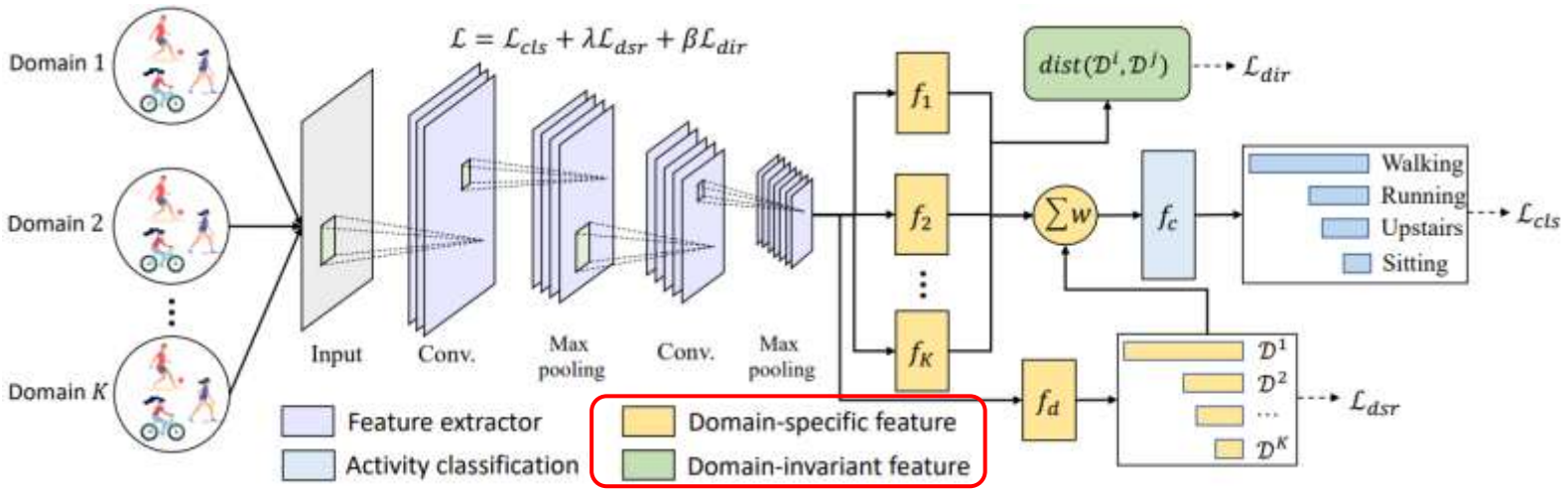


Li Z, Ren K, Jiang X, et al. Domain Generalization using Pretrained Models without Fine-tuning[J]. arXiv preprint arXiv:2203.04600, 2022.

Zhou K, Yang Y, Qiao Y, et al. Domain adaptive ensemble learning[J]. IEEE TIP 2021.

Ensemble learning for DG

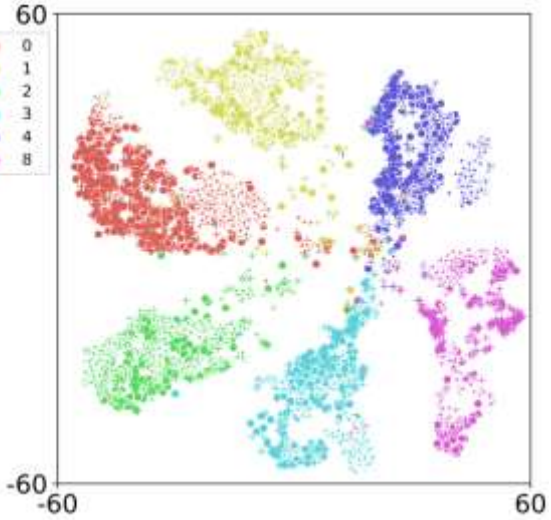
- Is ensemble learning enough for DG?
  - No. Ensemble → domain-specific knowledge
  - We also need a balance with domain-invariant knowledge
  - AFFAR: Adaptive Feature Fusion



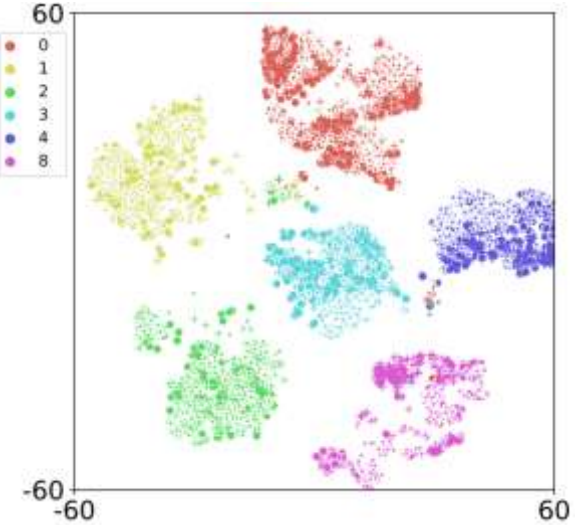
$$\mathcal{L} = \mathcal{L}_{cls} + \lambda\mathcal{L}_{dsr} + \beta\mathcal{L}_{dir}$$

Qin et al. Domain generalization for activity recognition via adaptive feature fusion. ACM TIST 2022.

Domain-specific features



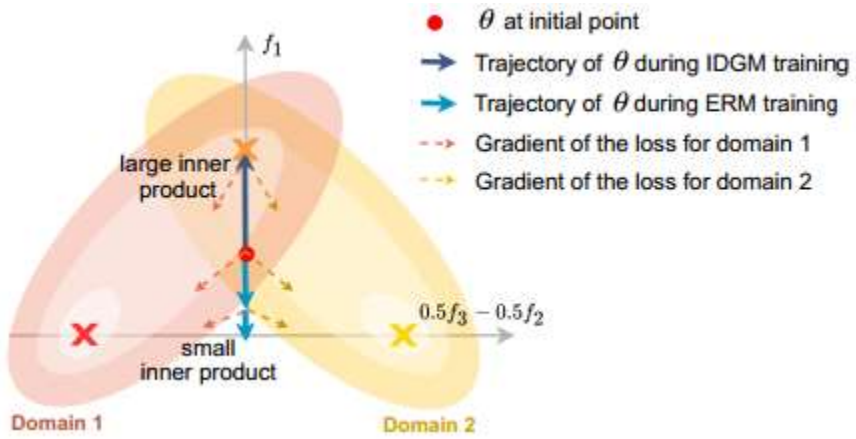
+Domain-invariant features



Gradient operation for DG

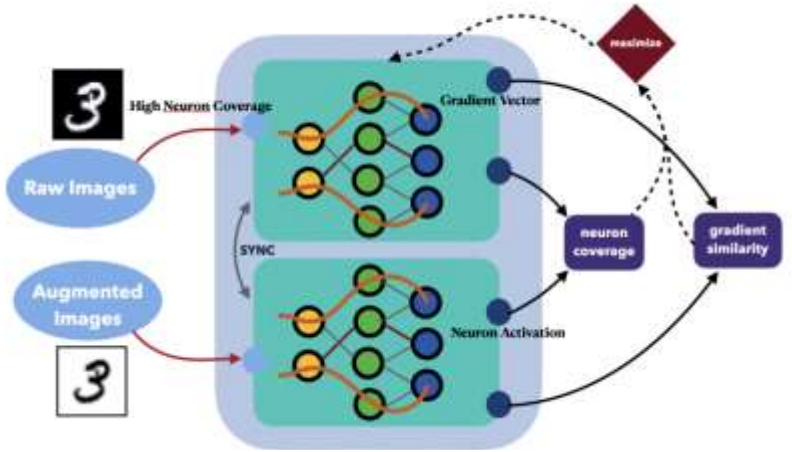
- Model the interactions between cross-domain gradients

Fish: gradient inner product



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

NCDG: gradient L2 norm
(with coverage regularization)



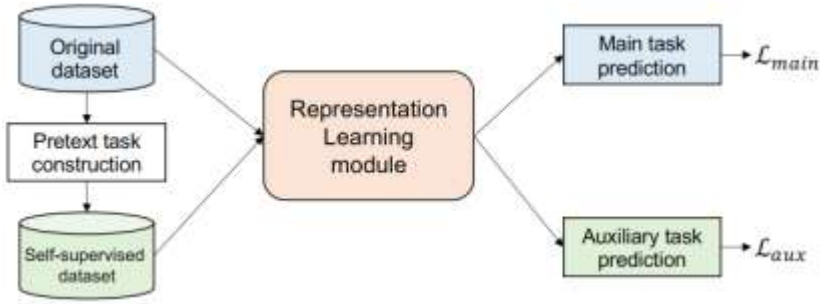
$$\mathcal{L}_{sim}(f) = \left\| \frac{\partial L_{cov}(f, \mathcal{D}_S)}{\partial \Theta} - \frac{\partial L_{cov}(f, \hat{\mathcal{D}}_S)}{\partial \Theta} \right\|_2$$

- Shi Y, Seely J, Torr P H S, et al. Gradient matching for domain generalization. ICLR 2022.
- Chris Xing Tian, Haoliang LI, et al. Neuron-coverage guided domain generalization. TPAMI 2023.

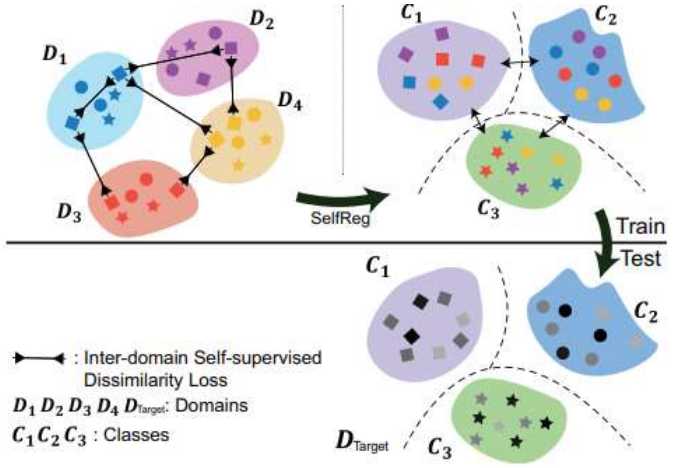
Self-supervised learning for DG

- Construct pretext tasks for general representation learning

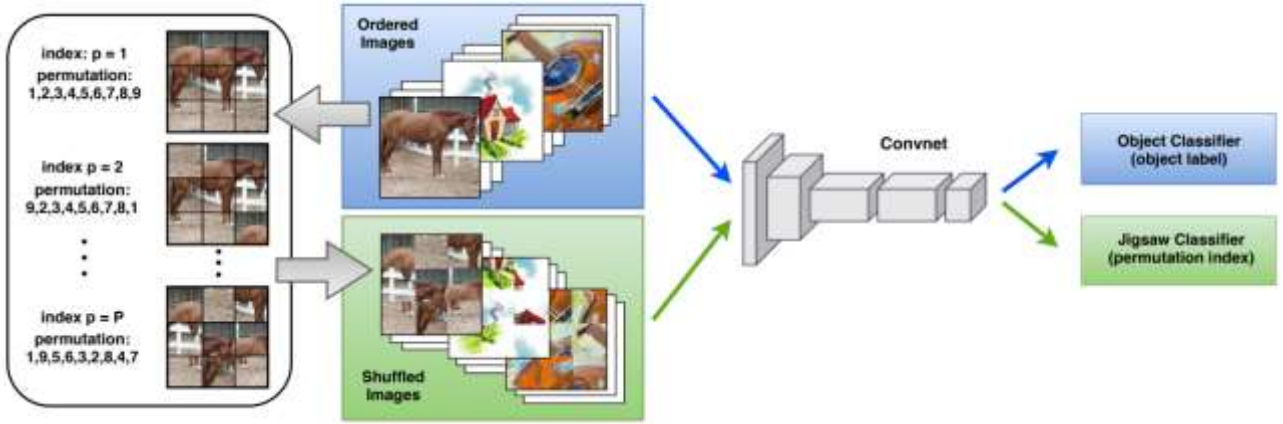
Self-supervised learning



Selfreg: self-supervised contrastive loss



JiGen: Jigsaw puzzle + DG

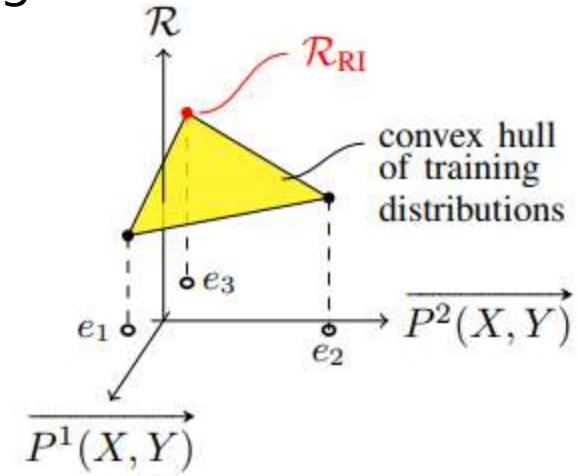


- Carlucci F M, D'Innocente A, Bucci S, et al. Domain generalization by solving jigsaw puzzles. CVPR 2019.
- Kim D, Yoo Y, Park S, et al. Selfreg: Self-supervised contrastive regularization for domain generalization. ICCV 2021.

Distributionally robust optimization for DG

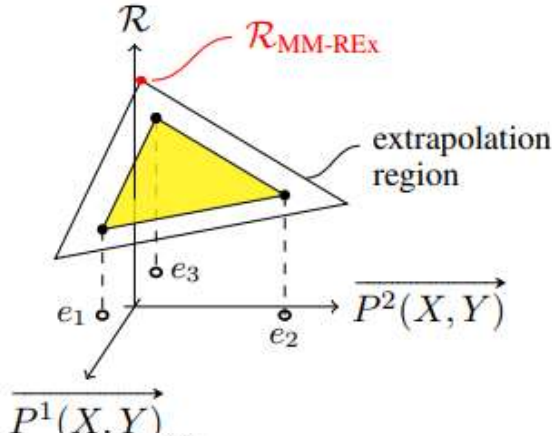
- Learn a model at worst-case distribution scenario

GroupDRO: Convex hull + Regularization



$$\min_{\theta\in\Theta} \sup_{q\in\Delta_m} \sum_{g=1}^{m} q_g \mathbb{E}_{(x,y)\sim P_g}[\ell(\theta;(x,y))]$$

VRex: Convex hull + Perturbation + Risk variance



$$\mathcal{R}_{\text{MM-REx}}(\theta) \doteq \max_{\substack{\Sigma_e \lambda_e = 1 \\ \lambda_e \geq \lambda_{\min}}} \sum_{e=1}^{m} \lambda_e \mathcal{R}_e(\theta) = (1 - m\lambda_{\min}) \max_e \mathcal{R}_e(\theta) + \lambda_{\min} \sum_{e=1}^{m} \mathcal{R}_e(\theta)$$

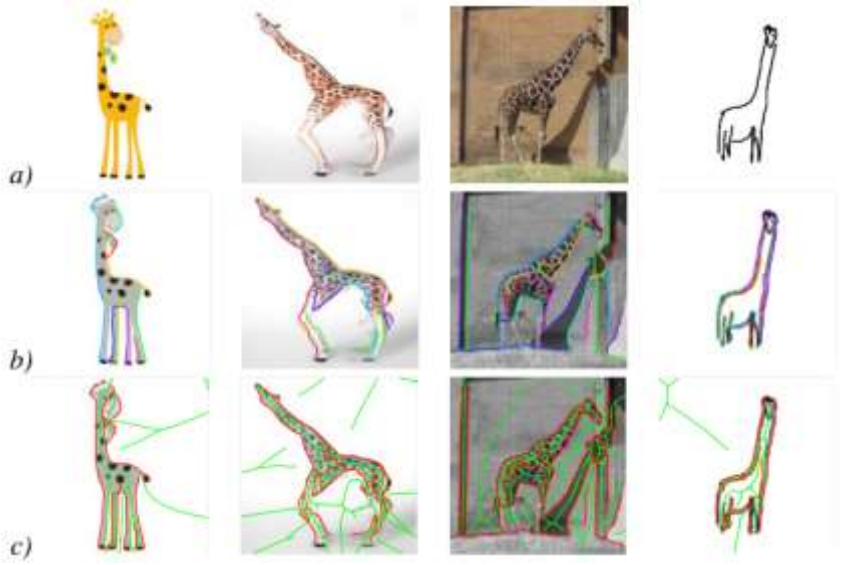
$$\mathcal{R}_{\text{V-REx}}(\theta) \doteq \beta \operatorname{Var}(\{\mathcal{R}_1(\theta), ..., \mathcal{R}_m(\theta)\}) + \sum_{e=1}^{m} \mathcal{R}_e(\theta)$$

- S. Sagawa, P. W. Koh, T. B. Hashimoto, and P. Liang, "Distributionally robust neural networks for group shifts: On the importance of regularization for worst-case generalization," in ICLR, 2020.
- D. Krueger, E. Caballero, J.-H. Jacobsen, A. Zhang, J. Binas, D. Zhang, R. Le Priol, and A. Courville, "Out-of-distribution generalization via risk extrapolation (rex)," in ICML, 2021, pp. 5815–5826.

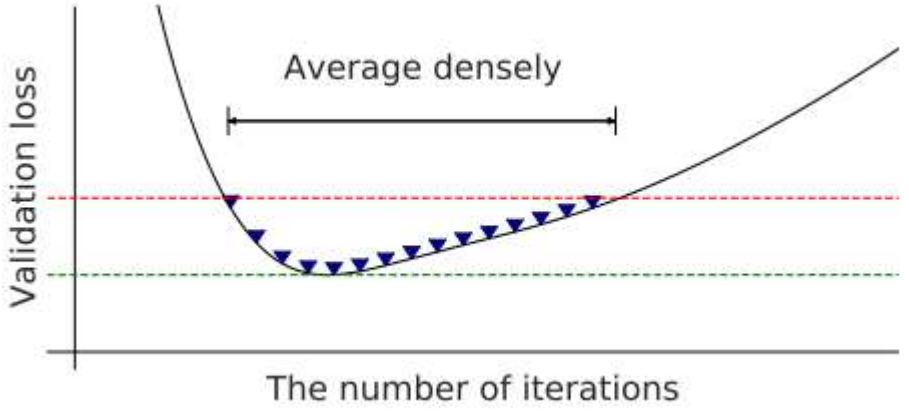
Other learning strategy

- Other interesting learning strategy for DG

Shapelet feature: invariant across domains



SWAD: Smooth training loss



- Narayanan M, Rajendran V, Kimia B. Shape-biased domain generalization via shock graph embeddings. ICCV 2021.
- Cha J, Chun S, Lee K, et al. Swad: Domain generalization by seeking flat minima. NeurIPS 2021.

Applications and benchmarks for DG

- Wide applications across CV, NLP, RL, and others

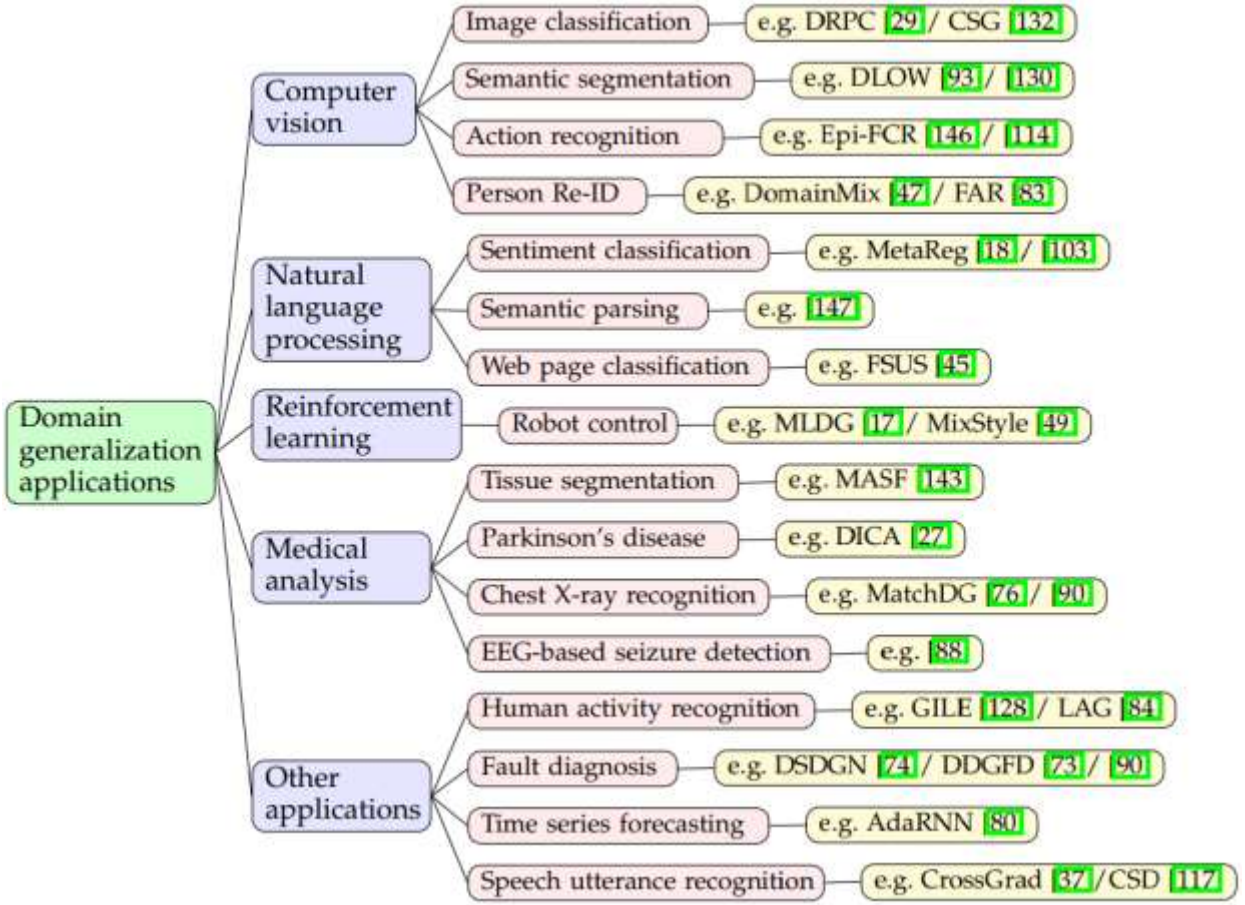
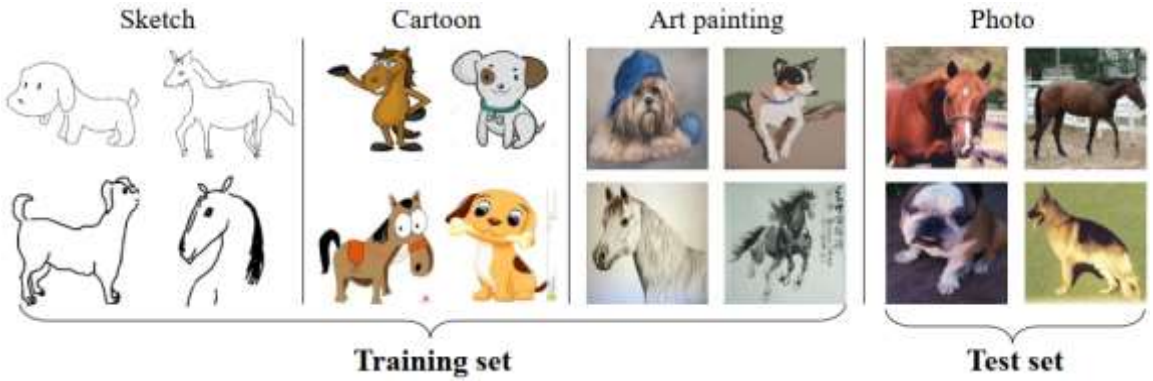


Figure credit: DG survey by Wang et al. (TKDE'22)

Wide applications of DG

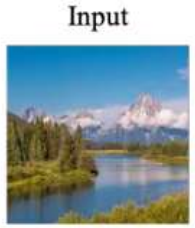
- Computer vision

Image classification

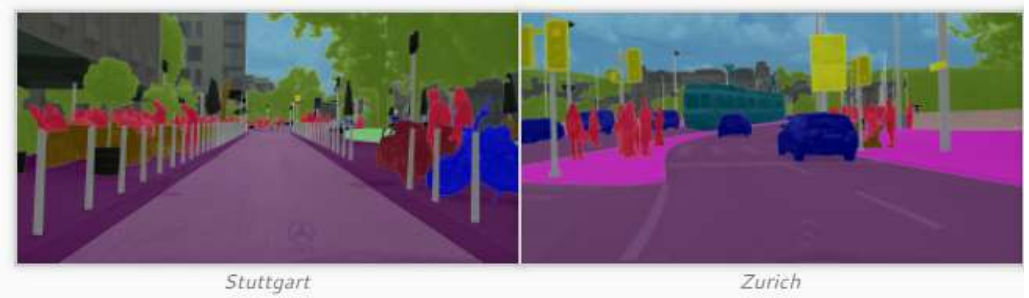


Action recognition

Style transfer



Semantic segmentation



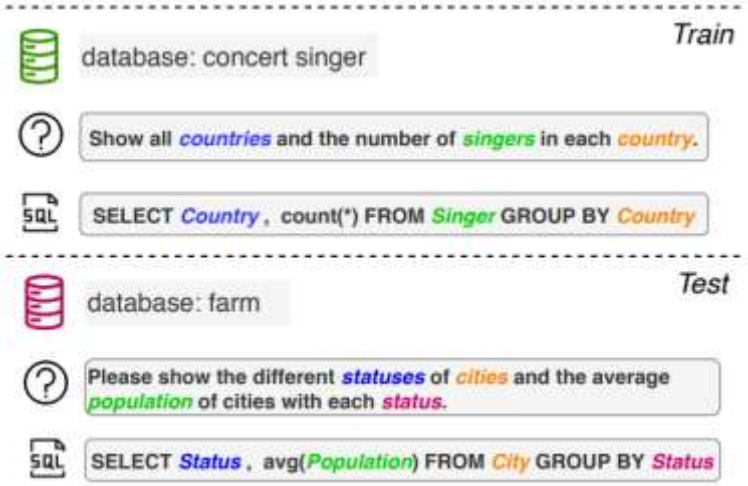
Person ReID

Wide applications of DG

- Natural language processing

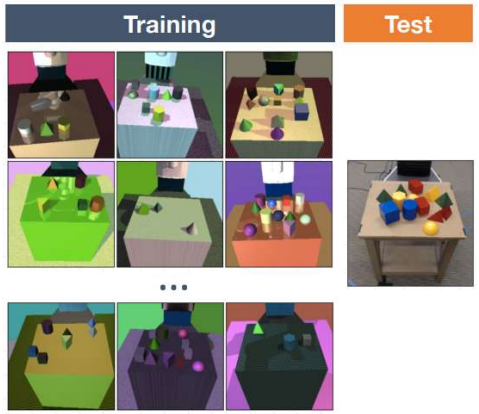
  Sentiment classification

Semantic parsing



- Reinforcement learning

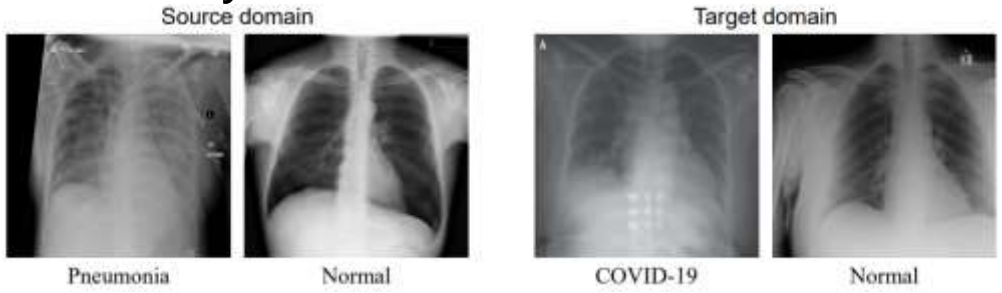
Sim-to-real Robot control



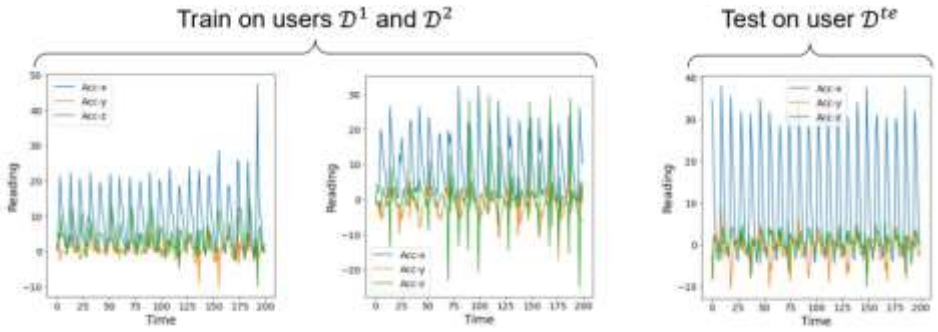
Wide applications of DG

- Medical applications

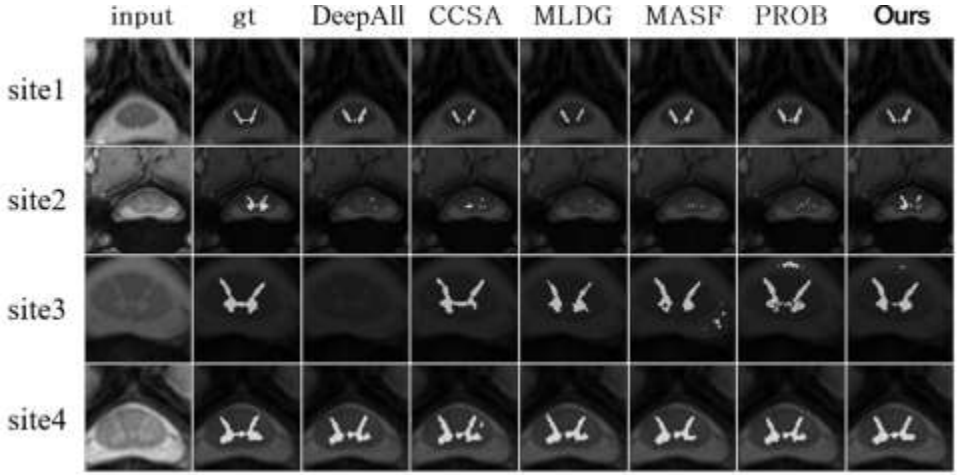
COVID X-ray classification



Parkinson's disease diagnosis



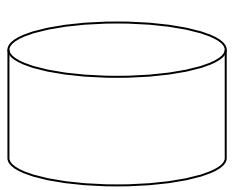
Tissue segmentation



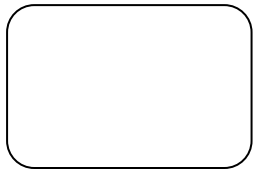
[Li, NeurIPS'20]

Benchmarks for DG

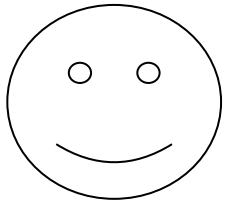
- Important consideration for DG benchmarks:



Which dataset?



Which codebase?



Which metric?

  - Popular datasets
  - Common benchmarks and codebases
  - Evaluation strategy: model selection

Note:

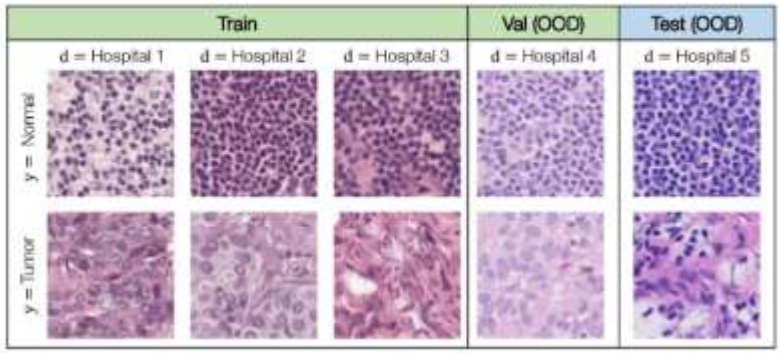
- Technically, ***any*** application settings that fits in DG scenario can be considered as a good test bed.
- There exists ***no*** "golden-standard" for benchmarking and evaluation.

Datasets for DG

- Common benchmarks

Camelyon17

WILDS



FMoW

Dataset	#Domain	#Class	#Sample	Description
Office-Caltech	4	10	2,533	Caltech, Amazon, Webcam, DSLR
Office-31	3	31	4,110	Amazon, Webcam, DSLR
PACS	4	7	9,991	Art, Cartoon, Photos, Sketches
VLCS	4	5	10,729	Caltech101, LabelMe, SUN09, VOC2007
Office-Home	4	65	15,588	Art, Clipart, Product, Real
Terra Incognita	4	10	24,788	Wild animal images taken at locations L100, L38, L43, L46
Rotated MNIST	6	10	70,000	Digits rotated from 0° to 90° with an interval of 15°
DomainNet	6	345	586,575	Clipart, Infograph, Painting, Quickdraw, Real, Sketch
iWildCam2020-wilds	323	182	203,029	Species classification across different camera traps
Camelyon17-wilds	5	2	45,000	Tumor identification across five different hospitals
RxRx1-wilds	51	1,139	84,898	Genetic perturbation classification across experimental batches
OGB-MolPCBA	120,084	128	400,000	Molecular property prediction across different scaffolds
GlobalWheat-wilds	47	bounding boxes	6,515	Wheat head detection across regions of the world
CivilComments-wilds	-	2	450,000	Toxicity classification across demographic identities
FMoW-wilds	80	62	118,886	Land use classification across different regions and years
PovertyMap-wilds	46	real value	19,669	Poverty mapping across different countries
Amazon-wilds	3920	5	539,502	Sentiment classification across different users
Py150-wilds	8,421	next token	150,000	Code completion across different codebases

Benchmark and codebase

- DomainBed
  - A unified benchmark for domain generalization

Available datasets

The currently available datasets are:

- RotatedMNIST (Ghifary et al., 2015)
- ColoredMNIST (Arjovsky et al., 2019)
- VLCS (Fang et al., 2013)
- PACS (Li et al., 2017)
- Office-Home (Venkateswara et al., 2017)
- A TerraIncognita (Beery et al., 2018) subset
- DomainNet (Peng et al., 2019)
- A SVIRO (Dias Da Cruz et al., 2020) subset
- WILDS (Koh et al., 2020) FMoW (Christie et al., 2018) about satellite images
- WILDS (Koh et al., 2020) Camelyon17 (Bandi et al., 2019) about tumor detection in tissues

Algorithm	CMNIST	RMNIST	VLCS	PACS	OfficeHome	TerraInc	DomainNet	Average
ERM	51.5 ± 0.1	98.0 ± 0.0	77.5 ± 0.4	85.5 ± 0.2	66.5 ± 0.3	46.1 ± 1.8	40.9 ± 0.1	66.6
IRM	52.0 ± 0.1	97.7 ± 0.1	78.5 ± 0.5	83.5 ± 0.8	64.3 ± 2.2	47.6 ± 0.8	33.9 ± 2.8	65.4
GroupDRO	52.1 ± 0.0	98.0 ± 0.0	76.7 ± 0.6	84.4 ± 0.8	66.0 ± 0.7	43.2 ± 1.1	33.3 ± 0.2	64.8
Mixup	52.1 ± 0.2	98.0 ± 0.1	77.4 ± 0.6	84.6 ± 0.6	68.1 ± 0.3	47.9 ± 0.8	39.2 ± 0.1	66.7
MLDG	51.5 ± 0.1	97.9 ± 0.0	77.2 ± 0.4	84.9 ± 1.0	66.8 ± 0.6	47.7 ± 0.9	41.2 ± 0.1	66.7
CORAL	51.5 ± 0.1	98.0 ± 0.1	78.8 ± 0.6	86.2 ± 0.3	68.7 ± 0.3	47.6 ± 1.0	41.5 ± 0.1	67.5
MMD	51.5 ± 0.2	97.9 ± 0.0	77.5 ± 0.9	84.6 ± 0.5	66.3 ± 0.1	42.2 ± 1.6	23.4 ± 9.5	63.3
DANN	51.5 ± 0.3	97.8 ± 0.1	78.6 ± 0.4	83.6 ± 0.4	65.9 ± 0.6	46.7 ± 0.5	38.3 ± 0.1	66.1
CDANN	51.7 ± 0.1	97.9 ± 0.1	77.5 ± 0.1	82.6 ± 0.9	65.8 ± 1.3	45.8 ± 1.6	38.3 ± 0.3	65.6
MTL	51.4 ± 0.1	97.9 ± 0.0	77.2 ± 0.4	84.6 ± 0.5	66.4 ± 0.5	45.6 ± 1.2	40.6 ± 0.1	66.2
SagNet	51.7 ± 0.0	98.0 ± 0.0	77.8 ± 0.5	86.3 ± 0.2	68.1 ± 0.1	48.6 ± 1.0	40.3 ± 0.1	67.2
ARM	56.2 ± 0.2	98.2 ± 0.1	77.6 ± 0.3	85.1 ± 0.4	64.8 ± 0.3	45.5 ± 0.3	35.5 ± 0.2	66.1
VREx	51.8 ± 0.1	97.9 ± 0.1	78.3 ± 0.2	84.9 ± 0.6	66.4 ± 0.6	46.4 ± 0.6	33.6 ± 2.9	65.6
RSC	51.7 ± 0.2	97.6 ± 0.1	77.1 ± 0.5	85.2 ± 0.9	65.5 ± 0.9	46.6 ± 1.0	38.9 ± 0.5	66.1

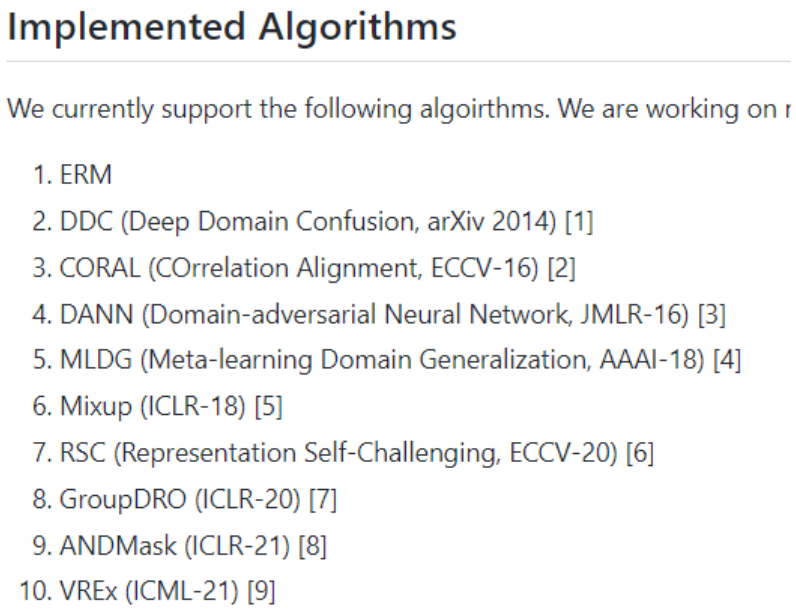
Model selection: training-domain validation set

Interesting results: DomainBed found that there ***are not*** significant improvements for recent DG algorithms. *Is it the case?*

https://github.com/facebookresearch/DomainBed

Benchmark and codebase

- DeepDG
  - Built by borrowing the knowledge from DomainBed, but *faster*, and *easier* to use



Implemented Algorithms

We currently support the following algoirthms. We are working on r

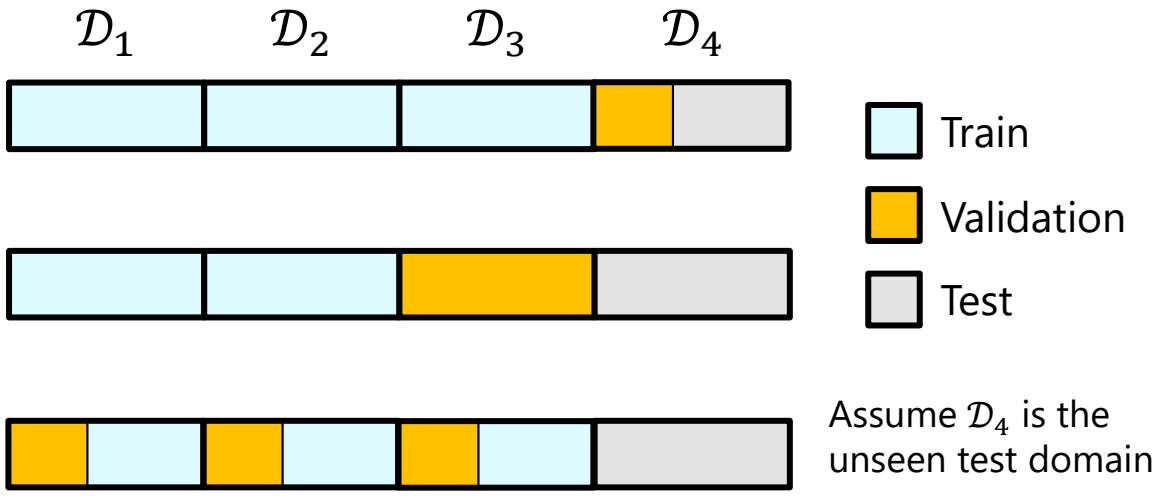
1. ERM
2. DDC (Deep Domain Confusion, arXiv 2014) [1]
3. CORAL (COrrelation Alignment, ECCV-16) [2]
4. DANN (Domain-adversarial Neural Network, JMLR-16) [3]
5. MLDG (Meta-learning Domain Generalization, AAAI-18) [4]
6. Mixup (ICLR-18) [5]
7. RSC (Representation Self-Challenging, ECCV-20) [6]
8. GroupDRO (ICLR-20) [7]
9. ANDMask (ICLR-21) [8]
10. VREx (ICML-21) [9]

- Avoids huge hyperparameter tuning
- More friendly interface
- Better customization

https://github.com/jindongwang/transferlearning/tree/master/code/DeepDG

Model selection

- Model selection in DomainBed
  - Test-domain validation set (oracle)
    - Use part of test domain as the validation
  - Leave-one-domain-out cross-validation
    - One domain as testing domain for validation
  - Training-domain validation set ***(popular)***
    - Leave some part of the training data as the validation set



- Q: is it reasonable to use training-domain validation for model selection?
- A: ***no***. Since the validation distribution cannot represent the test distribution.

Discussion about the performance of DG

- Performance should be restricted to certain applications
  - Cross-dataset human activity recognition[1]

Source	Target	DeepALL	DANN	CORAL	ANDMask	GroupDRO	RSC	Mixup	SDMix
1,2,3,4	0	41.52	45.45	33.22	47.51	27.12	46.56	**48.77**	47.50
0,2,3,4	1	26.73	25.36	25.18	31.06	26.66	27.37	34.19	**36.10**
0,1,3,4	2	35.81	38.06	25.81	39.17	24.34	35.93	37.49	**42.53**
0,1,2,4	3	21.45	28.89	22.32	30.22	18.39	27.04	29.50	**34.52**
0,1,2,3	4	27.28	25.05	20.64	29.90	24.82	29.82	29.95	**30.93**
AVG	-	30.56	32.56	25.43	35.57	24.27	33.34	35.98	**38.32**

  - Cross-dataset object detection[2]

Setting	Method	Cityscapes→Foggy Cityscapes								
		person	rider	car	truck	bus	train	mcycle	bicycle	mAP
DG	Faster R-CNN [52]	17.8	23.6	27.1	11.9	23.8	9.1	14.4	22.8	18.8
	SNR-Faster R-CNN	**20.3**	**24.6**	**33.6**	**15.9**	**26.3**	**14.4**	**16.8**	**26.8**	**22.3**
UDA	DA Faster R-CNN [72]	25.0	31.0	40.5	22.1	35.3	20.2	20.0	27.1	27.6
	SNR-DA Faster R-CNN	**27.3**	**34.6**	**44.6**	**23.9**	**38.1**	**25.4**	**21.3**	**29.7**	**30.6**

Hint: maybe we should develop application-oriented evaluation benchmarks?

[1] Lu et al. Semantic-discriminative mixup for generalizable cross-domain sensor-based human activity recognition. ACM IMWUT 2022.
[2] Jin X, Lan C, Zeng W, et al. Style normalization and restitution for domain generalization and adaptation. IEEE TMM 2021.

Theory

- Domain adaptation error bound
  - The error on target domain is bounded by:

$$\epsilon^t(h) \leq \epsilon^s(h) + d_{\mathcal{H}\Delta\mathcal{H}}\left(P_X^s, P_X^t\right) + \lambda_{\mathcal{H}}$$

Target risk | Source risk | Source-target distribution divergence | Complexity of $\mathcal{H}$

$\mathcal{H}$-divergence: $d_{\mathcal{H}}(P,Q) = 2 \sup_{h\in\mathcal{H}} |Pr_P[\mathbb{I}(h)] - Pr_Q[\mathbb{I}(h)]|$

$\mathcal{H}\Delta\mathcal{H}$-distance: $d_{\mathcal{H}\Delta\mathcal{H}}(P,Q) = 2 \sup_{h,h'\in\mathcal{H}} |Pr_{x\sim P}[h(x) \neq h'(x)] - Pr_{x\sim Q}[h(x) \neq h'(x)]|$

Discrepancy distance: $disc_L(P,Q) = \max_{h,h'\in\mathcal{H}} |L_P(h,h') - L_Q(h,h')|$

- Ben-David S, Blitzer J, Crammer K, et al. Analysis of representations for domain adaptation. NIPS 2016.
- Ben-David S, Blitzer J, Crammer K, et al. A theory of learning from different domains[J]. Machine learning, 2010, 79(1): 151-175.
- Mansour Y, Mohri M, Rostamizadeh A. Domain adaptation with multiple sources. NIPS 2009.

Theory

- Domain generalization error bound
  - There's no target in DG. How to measure the error?
  - Key: approximate target domain using the convex hull of source distributions

$$\Lambda := \{\textstyle\sum_{i=1}^{M} \pi_i P_X^i \mid \pi \in \Delta_M\}$$

Ideal joint risk (best source vs. target)

$$\epsilon^t(h) \leq \sum_{i=1}^{M} \pi_i^* \epsilon^i(h) + \frac{\gamma + \rho}{2} + \lambda_{\mathcal{H},(P_X^t, P_X^*)},$$

Target risk

Weighted source risk

$\gamma := \min_{\pi \in \Delta_M} d_{\mathcal{H}}(P_X^t, \sum_{i=1}^{M} \pi_i P_X^i)$

Distance between target and convex hull

$\rho := \sup_{P_X', P_X'' \in \Lambda} d_{\mathcal{H}}(P_X', P_X'')$

Diameter of Λ

  - Other theory: average source risk
  - *DG theory is in infant and on the go!*

Blanchard G, Lee G, Scott C. Generalizing from several related classification tasks to a new unlabeled sample. NIPS 2011.
Albuquerque I, Monteiro J, Darvishi M, et al. Adversarial target-invariant representation learning for domain generalization[J]. 2020.

Invariant risk minimization

- IRM
  - Do not match distributions; enforce optimal *classifier* on top of the representation space to be the same across all domains

$$\min_{\substack{g\in\mathcal{G},\\ f\in\bigcap_{i=1}^{M}\arg\min_{f'\in\mathcal{F}}\epsilon^i(f'\circ g)}} \sum_{i=1}^{M}\epsilon^i(f\circ g)$$

$$\min_{g\in\mathcal{G}} \sum_{i=1}^{M}\epsilon^i(g) + \lambda\left\|\nabla_f \epsilon^i(f\circ g)\big|_{f=1}\right\|^2$$

Arjovsky et al. Invariant risk minimization. *arXiv preprint arXiv:1907.02893*, 2019

New DG settings

- Some new DG settings

Setting	Description
Traditional domain generalization	The traditional setting
Evolving domain generalization	Domains gradually change
Test-time domain adaptation/generalization	Updating model by using target domain/data
Federated domain generalization	Training data cannot be accessed by central server
Open domain generalization	Training and test domains have different label spaces
Unsupervised domain generalization	Training domains are totally unlabeled

- Tiexin Qin, Shiqi Wang, and Haoliang Li, Generalizing to Evolving Domains with Latent Structure-Aware Sequential Autoencoder, ICML'22
- Chenyu Yi, Siyuan Yang, Yufei Wang, Haoliang Li, Yap-Peng Tan and Alex C. Kot, Temporal Coherent Test Time Optimization for Robust Video Classification, ICLR'23
- Zhang L, Lei X, Shi Y, et al. Federated Learning with Domain Generalization[J]. arXiv preprint arXiv:2111.10487, 2021.
- Shu Y, Cao Z, Wang C, et al. Open domain generalization with domain-augmented meta-learning. CVPR 2021.
- Qi L, Wang L, Shi Y, et al. Unsupervised Domain Generalization for Person Re-identification: A Domain-specific Adaptive Framework[J]. arXiv preprint arXiv:2111.15077, 2021.

New DG settings

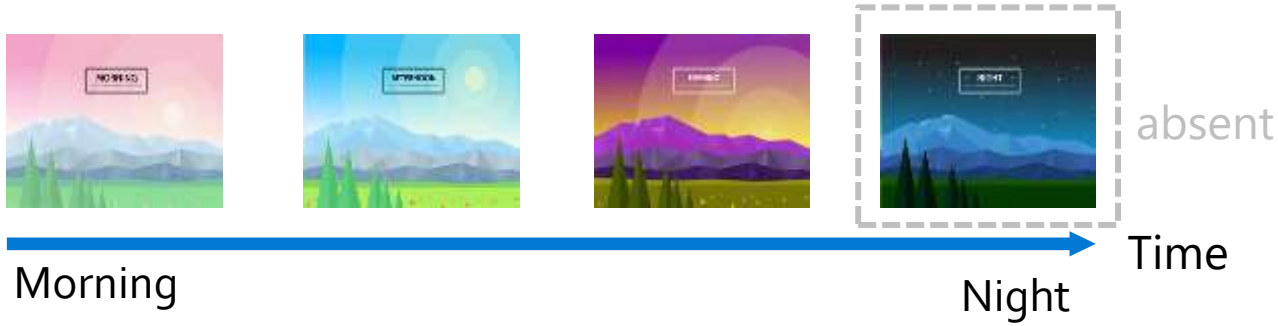
- Some new DG settings

Setting	Situation
Traditional domain generalization	The traditional setting
Evolving domain generalization	Domains gradually change
Test-time optimization	Updating model by using target domain/data
Federated domain generalization	Training data cannot be accessed by central server
Open domain generalization	Training and test domains have different label spaces
Unsupervised domain generalization	Training domains are totally unlabeled

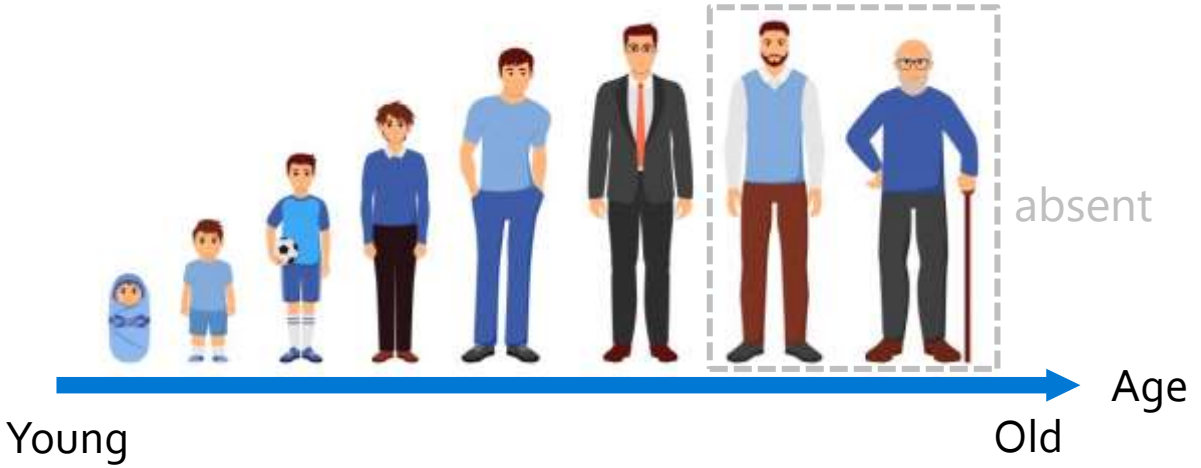
- Tiexin Qin, Shiqi Wang, and Haoliang Li, Generalizing to Evolving Domains with Latent Structure-Aware Sequential Autoencoder, ICML'22
- Chenyu Yi, Siyuan Yang, Yufei Wang, Haoliang Li, Yap-Peng Tan and Alex C. Kot, Temporal Coherent Test Time Optimization for Robust Video Classification, ICLR'23
- Zhang L, Lei X, Shi Y, et al. Federated Learning with Domain Generalization[J]. arXiv preprint arXiv:2111.10487, 2021.
- Shu Y, Cao Z, Wang C, et al. Open domain generalization with domain-augmented meta-learning. CVPR 2021.
- Qi L, Wang L, Shi Y, et al. Unsupervised Domain Generalization for Person Re-identification: A Domain-specific Adaptive Framework[J]. arXiv preprint arXiv:2111.15077, 2021.

Evolving Domain Generalization

Automated driving system

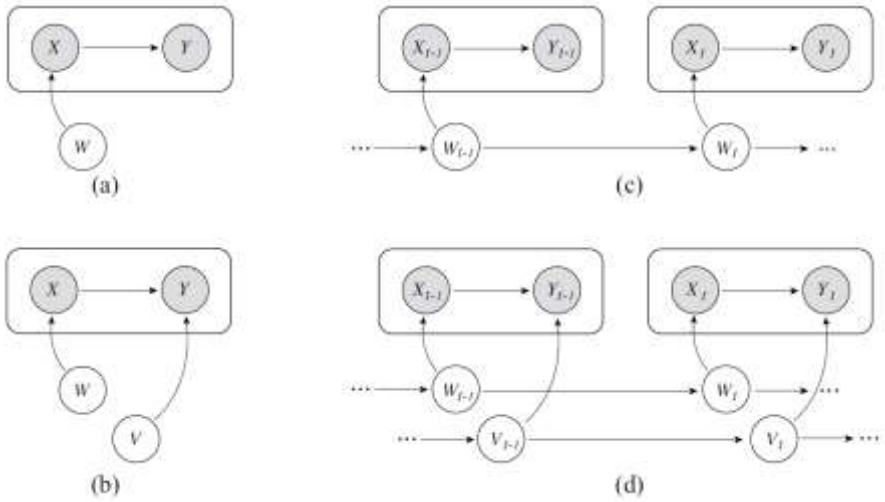


Disease diagnosis



Evolving Domain Generalization: Method

Latent Structure-aware Sequence AutoEncoder (LSSAE)



Bayesian rule:

$$P(X, Y) = P(X)P(Y|X)$$

Distribution shift:

(1) Covariate shift

$$P(X^s) \neq P(X^t)$$

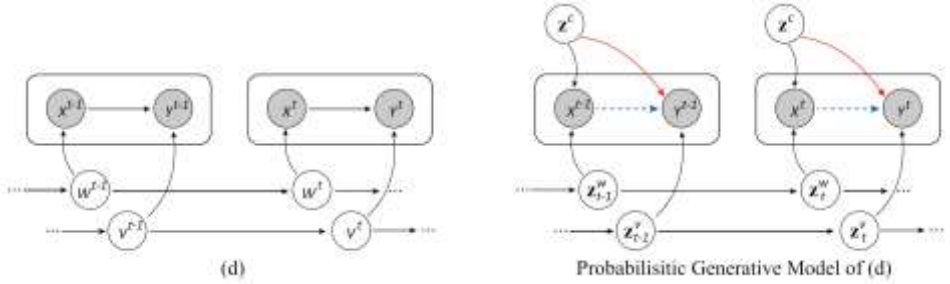
(2) Concept shift

$$P(Y^s|X^s) \neq P(Y^t|X^t)$$

- Tiexin Qin, Shiqi Wang, Haoliang Li. Generalizing to Evolving Domains with Latent Structure-Aware Sequential Autoencoder. ICML 2022.

Evolving Domain Generalization: Method

From casual diagram to probabilistic generative model



Definition: latent codes $(\boldsymbol{z}^c, \boldsymbol{z}^w_{1:T}, \boldsymbol{z}^v_{1:T})$
$\boldsymbol{z}^c$: static domain-invariant category information from X
$\boldsymbol{z}^w_{1:T}$: dynamic domain-specific information from X
$\boldsymbol{z}^v_{1:T}$: dynamic domain-specific category information from Y

(1) Markov chain of the latent codes:

$$p(\boldsymbol{z}^w_t) = p(\boldsymbol{z}^w_t|\boldsymbol{z}^w_{<t}) \text{ , } p(\boldsymbol{z}^v_t) = p(\boldsymbol{z}^v_t|\boldsymbol{z}^v_{<t})$$

$\boldsymbol{z}^c \sim \mathcal{N}(0,1)$ is a fixed Gaussian distribution

(2) Probabilistic generative model :

$$p(\mathbf{x}_{1:T}, \mathbf{y}_{1:T}, \boldsymbol{z}^c, \boldsymbol{z}^w_{1:T}, \boldsymbol{z}^v_{1:T}) = p(\mathbf{x}_{1:T}, \boldsymbol{z}^w_{1:T}, \boldsymbol{z}^c) p(\mathbf{y}_{1:T}, \boldsymbol{z}^w_{1:T}|\boldsymbol{z}^c)$$

(3) Variational Inference to approximate the prior

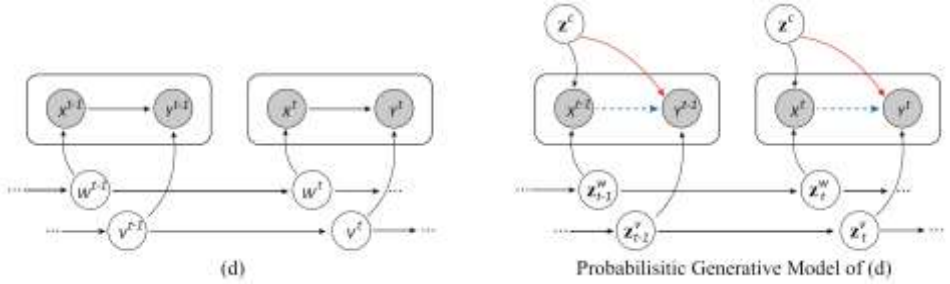
$$q(\boldsymbol{z}^c, \boldsymbol{z}^w_{1:T}, |\mathbf{x}_{1:T}), \; q(\boldsymbol{z}^v_{1:T}|\mathbf{y}_{1:T})$$

(4) Evidence lower bound (ELBO) for optimization:

$$\begin{aligned}\mathcal{L}_d = \sum_{t=1}^{T} \mathbb{E}_{q(z^c, z^w_t, z^v_t)}[\log p(\mathbf{x}_t|\boldsymbol{z}^c, \boldsymbol{z}^w_t) p(\boldsymbol{y}_t|\boldsymbol{z}^c, \boldsymbol{z}^v_t)] \\ -\lambda_1 \mathbb{D}_{KL}\big(q(\boldsymbol{z}^c|\mathbf{x}_{1:T}), p(\boldsymbol{z}^c)\big) \\ -\lambda_2 \mathbb{D}_{KL}\big(q(\boldsymbol{z}^w_t|\boldsymbol{z}^w_{<t}, \mathbf{x}_t), p(\boldsymbol{z}^w_t|\boldsymbol{z}^w_{<t})\big) \\ -\lambda_3 \mathbb{D}_{KL}\big(q(\boldsymbol{z}^v_t|\boldsymbol{z}^v_{<t}, \mathbf{x}_t), p(\boldsymbol{z}^v_t|\boldsymbol{z}^v_{<t})\big)\end{aligned}$$

Evolving Domain Generalization: Method

From casual diagram to probabilistic generative model



Definition: latent codes $(\boldsymbol{z}^c, \boldsymbol{z}^w_{1:T}, \boldsymbol{z}^v_{1:T})$
$\boldsymbol{z}^c$: static domain-invariant category information from X
$\boldsymbol{z}^w_{1:T}$: dynamic domain-specific information from X
$\boldsymbol{z}^v_{1:T}$: dynamic domain-specific category information from Y

(1) Markov chain of the latent codes:

$$p(\boldsymbol{z}^w_t) = p(\boldsymbol{z}^w_t|\boldsymbol{z}^w_{<t}) \text{ , } p(\boldsymbol{z}^v_t) = p(\boldsymbol{z}^v_t|\boldsymbol{z}^v_{<t})$$

$\boldsymbol{z}^c \sim \mathcal{N}(0,1)$ is a fixed Gaussian distribution

(2) Probabilistic generative model :

$$p(\mathbf{x}_{1:T}, \mathbf{y}_{1:T}, \boldsymbol{z}^c, \boldsymbol{z}^w_{1:T}, \boldsymbol{z}^v_{1:T}) = p(\mathbf{x}_{1:T}, \boldsymbol{z}^w_{1:T}, \boldsymbol{z}^c)p(\mathbf{y}_{1:T}, \boldsymbol{z}^w_{1:T}|\boldsymbol{z}^c)$$

(3) Variational Inference to approximate the prior

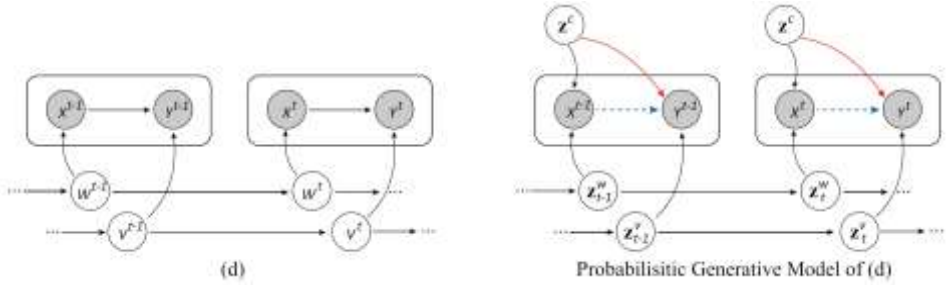
$$q(\boldsymbol{z}^c, \boldsymbol{z}^w_{1:T}, |\mathbf{x}_{1:T}), \; q(\boldsymbol{z}^v_{1:T}|\mathbf{y}_{1:T})$$

(4) Evidence lower bound (ELBO) for optimization:

$$\begin{aligned}\mathcal{L}_d = \sum_{t=1}^{T} \mathbb{E}_{q(z^c, z^w_t, z^v_t)}[\log p(\mathbf{x}_t|\boldsymbol{z}^c, \boldsymbol{z}^w_t)p(\boldsymbol{y}_t|\boldsymbol{z}^c, \boldsymbol{z}^v_t)] \\ -\lambda_1 \mathbb{D}_{KL}\big(q(\boldsymbol{z}^c|\mathbf{x}_{1:T}), p(\boldsymbol{z}^c)\big) \\ -\lambda_2 \mathbb{D}_{KL}\big(q(\boldsymbol{z}^w_t|\boldsymbol{z}^w_{<t}, \mathbf{x}_t), p(\boldsymbol{z}^w_t|\boldsymbol{z}^w_{<t})\big) \\ -\lambda_3 \mathbb{D}_{KL}\big(q(\boldsymbol{z}^v_t|\boldsymbol{z}^v_{<t}, \mathbf{x}_t), p(\boldsymbol{z}^v_t|\boldsymbol{z}^v_{<t})\big)\end{aligned}$$

Evolving Domain Generalization: Method

From casual diagram to probabilistic generative model



Definition: latent codes $(\boldsymbol{z}^c, \boldsymbol{z}^w_{1:T}, \boldsymbol{z}^v_{1:T})$
$\boldsymbol{z}^c$: static domain-invariant category information from X
$\boldsymbol{z}^w_{1:T}$: dynamic domain-specific information from X
$\boldsymbol{z}^v_{1:T}$: dynamic domain-specific category information from Y

(1) Markov chain of the latent codes:

$$p(\boldsymbol{z}^w_t) = p(\boldsymbol{z}^w_t|\boldsymbol{z}^w_{<t}) \text{ , } p(\boldsymbol{z}^v_t) = p(\boldsymbol{z}^v_t|\boldsymbol{z}^v_{<t})$$

$\boldsymbol{z}^c \sim \mathcal{N}(0,1)$ is a fixed Gaussian distribution

(2) Probabilistic generative model :

$$p(\mathbf{x}_{1:T}, \mathbf{y}_{1:T}, \boldsymbol{z}^c, \boldsymbol{z}^w_{1:T}, \boldsymbol{z}^v_{1:T}) = p(\mathbf{x}_{1:T}, \boldsymbol{z}^w_{1:T}, \boldsymbol{z}^c)p(\mathbf{y}_{1:T}, \boldsymbol{z}^w_{1:T}|\boldsymbol{z}^c)$$

(3) Variational Inference to approximate the prior

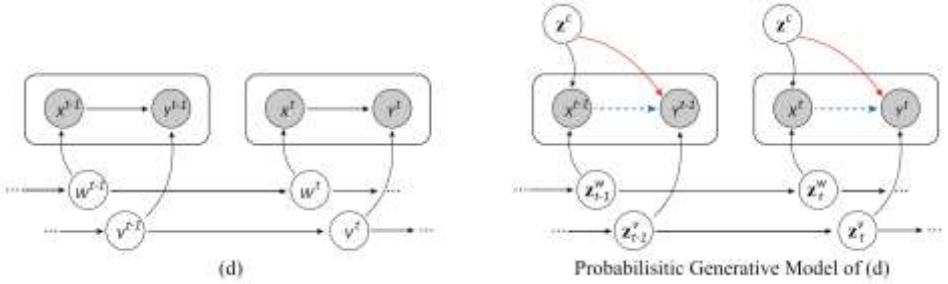
$$q(\boldsymbol{z}^c, \boldsymbol{z}^w_{1:T}, |\mathbf{x}_{1:T}), \; q(\boldsymbol{z}^v_{1:T}|\mathbf{y}_{1:T})$$

(4) Evidence lower bound (ELBO) for optimization:

$$\begin{aligned}\mathcal{L}_d = \sum_{t=1}^{T} \mathbb{E}_{q(z^c, z^w_t, z^v_t)}[\log p(\mathbf{x}_t|\boldsymbol{z}^c, \boldsymbol{z}^w_t)p(\boldsymbol{y}_t|\boldsymbol{z}^c, \boldsymbol{z}^v_t)] \\ -\lambda_1 \mathbb{D}_{KL}\big(q(\boldsymbol{z}^c|\mathbf{x}_{1:T}), p(\boldsymbol{z}^c)\big) \\ -\lambda_2 \mathbb{D}_{KL}\big(q(\boldsymbol{z}^w_t|\boldsymbol{z}^w_{<t}, \mathbf{x}_t), p(\boldsymbol{z}^w_t|\boldsymbol{z}^w_{<t})\big) \\ -\lambda_3 \mathbb{D}_{KL}\big(q(\boldsymbol{z}^v_t|\boldsymbol{z}^v_{<t}, \mathbf{x}_t), p(\boldsymbol{z}^v_t|\boldsymbol{z}^v_{<t})\big)\end{aligned}$$

Evolving Domain Generalization: Method

From casual diagram to probabilistic generative model



Definition: latent codes $(\boldsymbol{z}^c, \boldsymbol{z}^w_{1:T}, \boldsymbol{z}^v_{1:T})$
$\boldsymbol{z}^c$: static domain-invariant category information from X
$\boldsymbol{z}^w_{1:T}$: dynamic domain-specific information from X
$\boldsymbol{z}^v_{1:T}$: dynamic domain-specific category information from Y

(1) Markov chain of the latent codes:

$$p(\boldsymbol{z}^w_t) = p(\boldsymbol{z}^w_t|\boldsymbol{z}^w_{<t}) \text{ , } p(\boldsymbol{z}^v_t) = p(\boldsymbol{z}^v_t|\boldsymbol{z}^v_{<t})$$

$\boldsymbol{z}^c \sim \mathcal{N}(0,1)$ is a fixed Gaussian distribution

(2) Probabilistic generative model :

$$p(\mathbf{x}_{1:T}, \mathbf{y}_{1:T}, \boldsymbol{z}^c, \boldsymbol{z}^w_{1:T}, \boldsymbol{z}^v_{1:T}) = p(\mathbf{x}_{1:T}, \boldsymbol{z}^w_{1:T}, \boldsymbol{z}^c)p(\mathbf{y}_{1:T}, \boldsymbol{z}^w_{1:T}|\boldsymbol{z}^c)$$

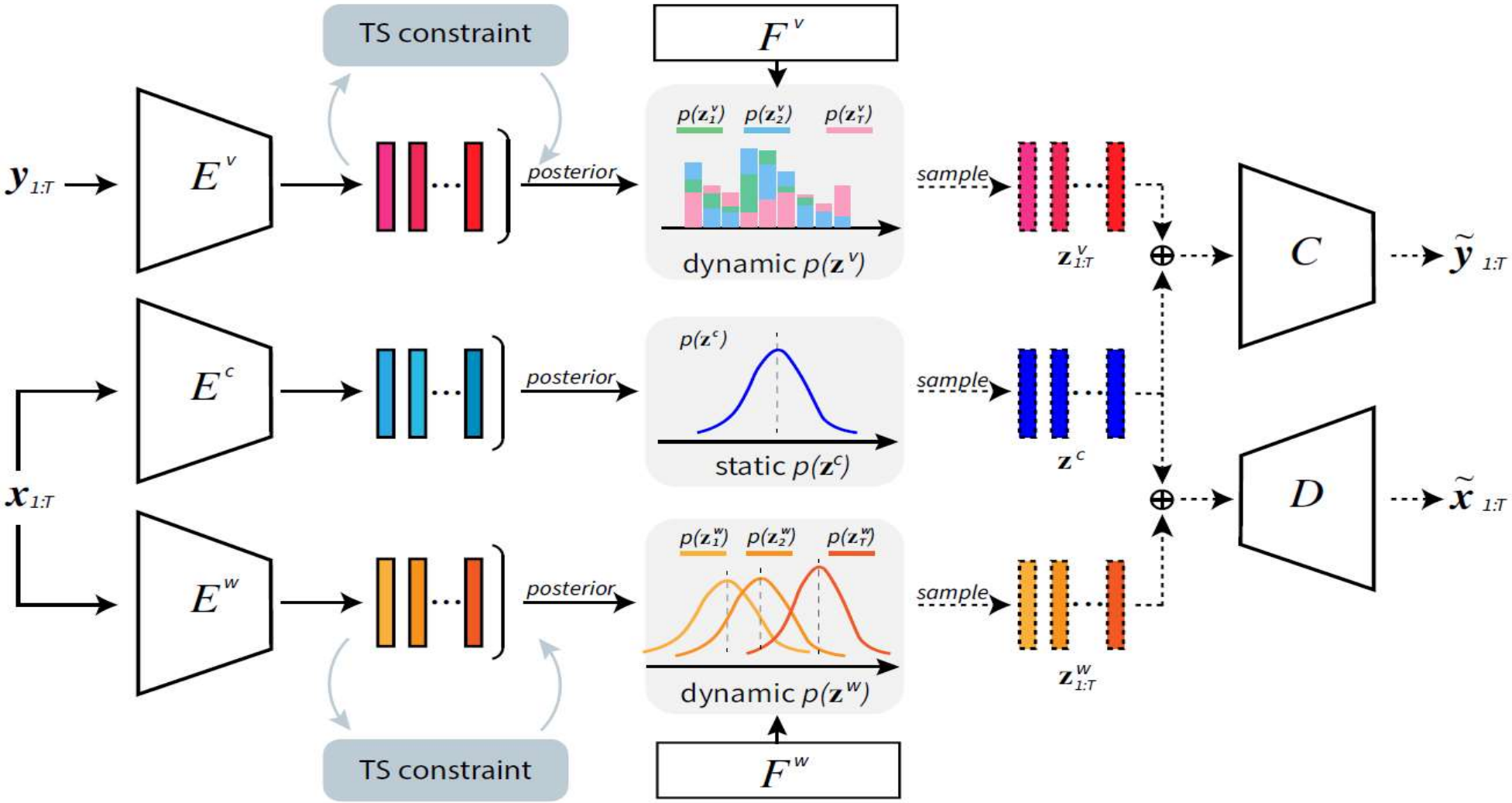
(3) Variational Inference to approximate the prior

$$q(\boldsymbol{z}^c, \boldsymbol{z}^w_{1:T}, |\mathbf{x}_{1:T}),\ q(\boldsymbol{z}^v_{1:T}|\mathbf{y}_{1:T})$$

(4) Evidence lower bound (ELBO) for optimization:

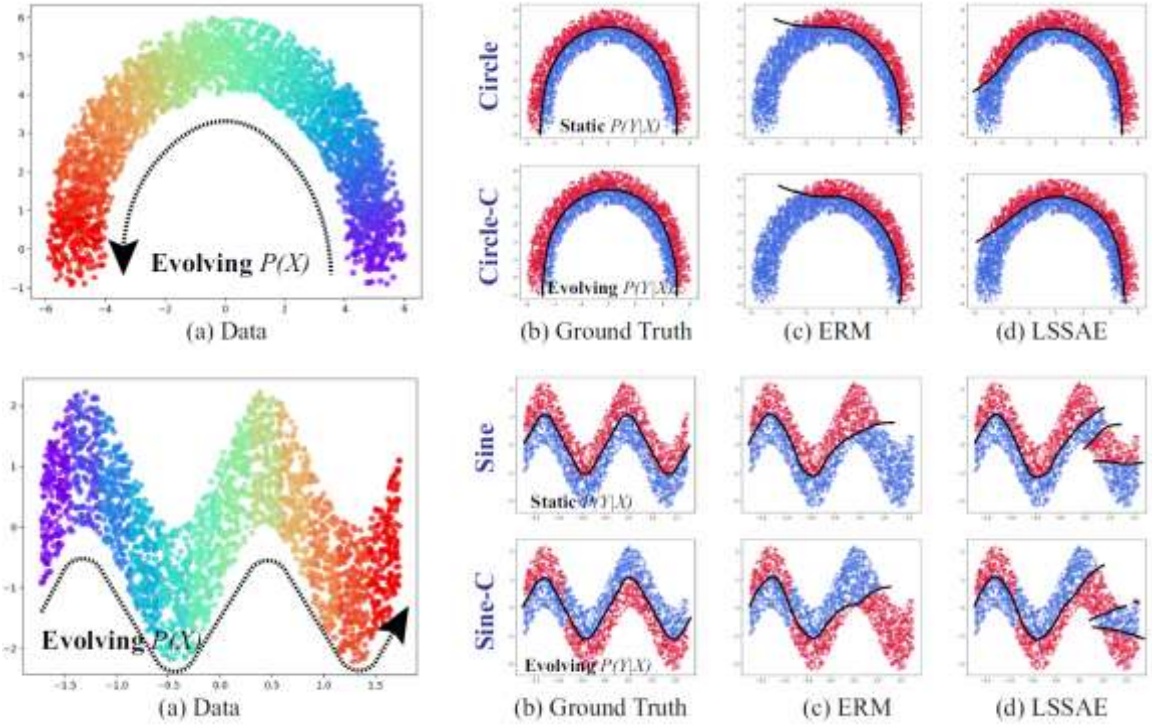
$$\begin{aligned}\mathcal{L}_d = \sum_{t=1}^{T} \mathbb{E}_{q(\boldsymbol{z}^c, \boldsymbol{z}^w_t, \boldsymbol{z}^v_t)}[\log p(\mathbf{x}_t|\boldsymbol{z}^c, \boldsymbol{z}^w_t)p(\boldsymbol{y}_t|\boldsymbol{z}^c, \boldsymbol{z}^v_t)] \\ -\lambda_1 \mathbb{D}_{KL}\big(q(\boldsymbol{z}^c|\mathbf{x}_{1:T}), p(\boldsymbol{z}^c)\big) \\ -\lambda_2 \mathbb{D}_{KL}\big(q(\boldsymbol{z}^w_t|\boldsymbol{z}^w_{<t}, \mathbf{x}_t), p(\boldsymbol{z}^w_t|\boldsymbol{z}^w_{<t})\big) \\ -\lambda_3 \mathbb{D}_{KL}\big(q(\boldsymbol{z}^v_t|\boldsymbol{z}^v_{<t}, \mathbf{x}_t), p(\boldsymbol{z}^v_t|\boldsymbol{z}^v_{<t})\big)\end{aligned}$$

Evolving Domain Generalization: Method



Evolving Domain Generalization: Results

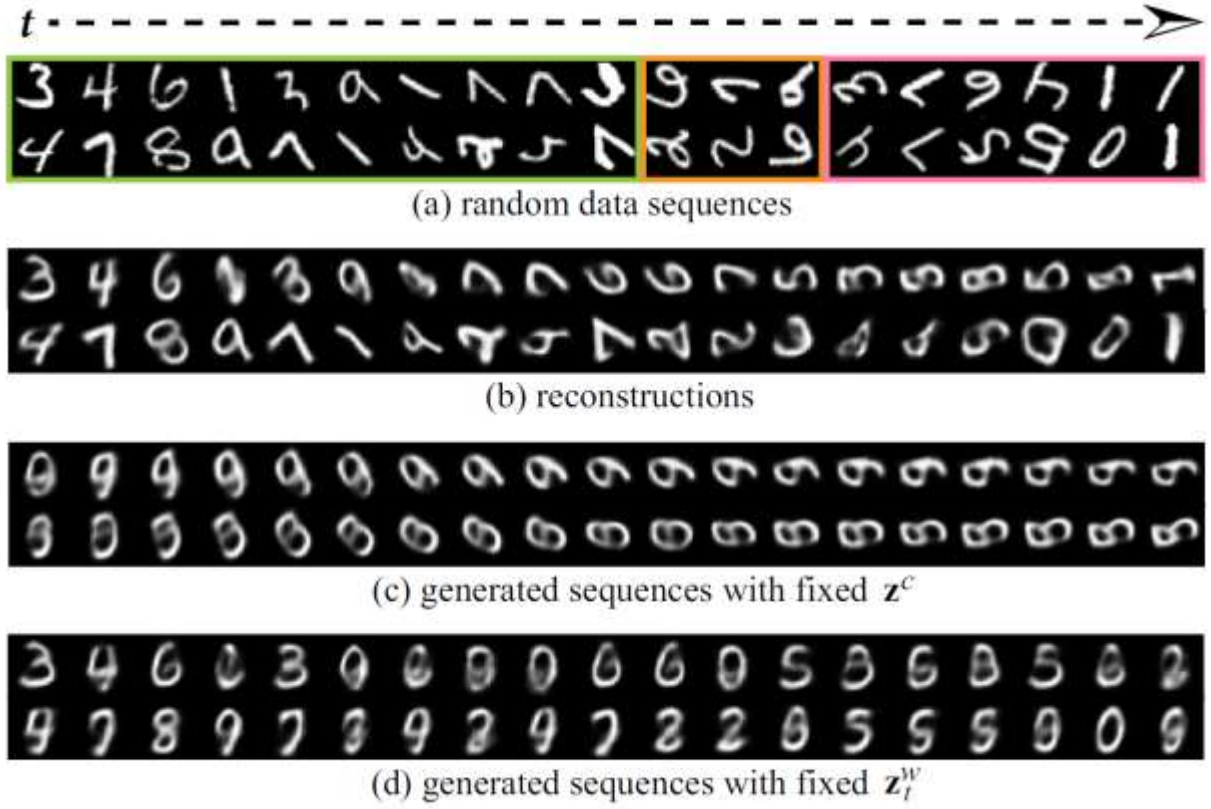
- Classification results on two toy datasets (More results can be found in our paper)
- Generation performance on RMNIST



1. LSSAE shows a desired generalization ability to unseen target domains
2. An accurate decision boundary for gradual concept shift but plain result for abrupt concept shift

Evolving Domain Generalization: Results

- Classification results on two toy datasets (More results can be found in our paper)
- Generation performance on RMNIST



LSSAE shows an ability of generating future unseen domains

New DG settings

- Some new DG settings

Setting	Description
Traditional domain generalization	The traditional setting
Evolving domain generalization	Domains gradually change
Test-time optimization	Updating model by using target domain/data
Federated domain generalization	Training data cannot be accessed by central server
Open domain generalization	Training and test domains have different label spaces
Unsupervised domain generalization	Training domains are totally unlabeled

- Tiexin Qin, Shiqi Wang, and Haoliang Li, Generalizing to Evolving Domains with Latent Structure-Aware Sequential Autoencoder, ICML'22
- Chenyu Yi, Siyuan Yang, Yufei Wang, Haoliang Li, Yap-Peng Tan and Alex C. Kot, Temporal Coherent Test Time Optimization for Robust Video Classification, ICLR'23
- Zhang L, Lei X, Shi Y, et al. Federated Learning with Domain Generalization[J]. arXiv preprint arXiv:2111.10487, 2021.
- Shu Y, Cao Z, Wang C, et al. Open domain generalization with domain-augmented meta-learning. CVPR 2021.
- Qi L, Wang L, Shi Y, et al. Unsupervised Domain Generalization for Person Re-identification: A Domain-specific Adaptive Framework[J]. arXiv preprint arXiv:2111.15077, 2021.

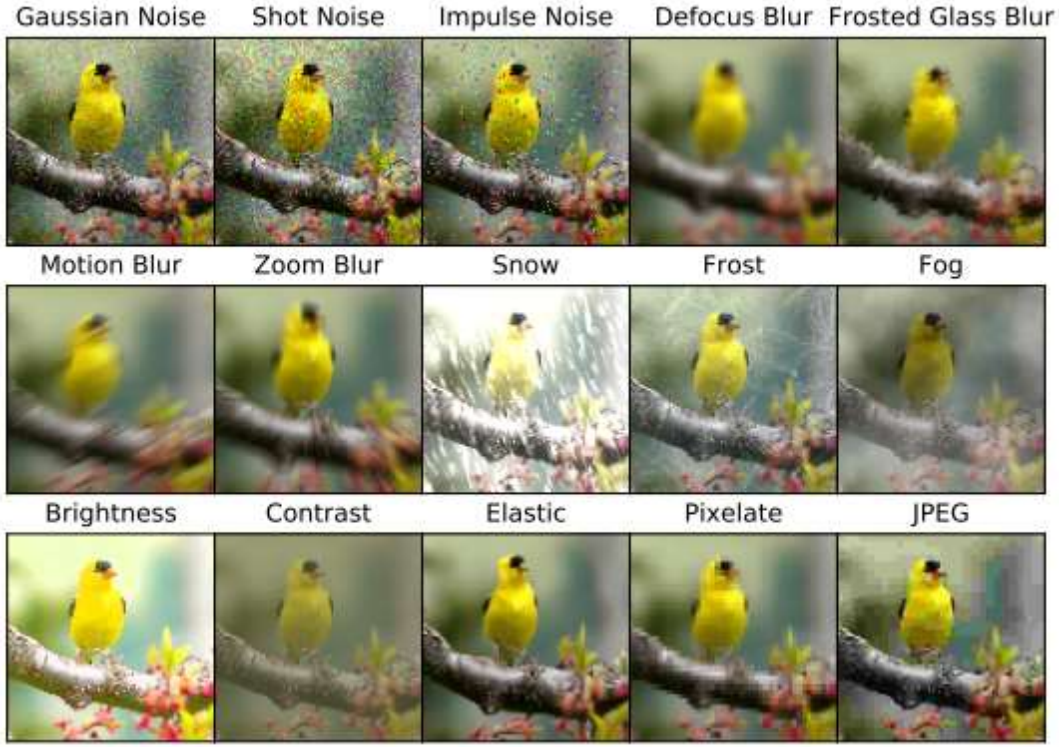
Test-time Optimization

- Test-time optimization is an effective method in improving model robustness

Setting	Source Data	Target Data	Train Loss	Test Loss
Fine Tune	No	Yes	Yes	No
Domain Adaption	Yes	Yes	Yes	No
Domain Generalization	Yes	No	Yes	No
Test-Time Optimization	No	Yes (batch/single sample)	No	Yes

Wang et al., Tent: Fully Test-Time Adaptation by Entropy Minimization,ICLR'21
Iwasawa et al., Test-Time Classifier Adjustment Module for Model-Agnostic Domain Generalization, NeurIPS'22

Test-time optimization for video data



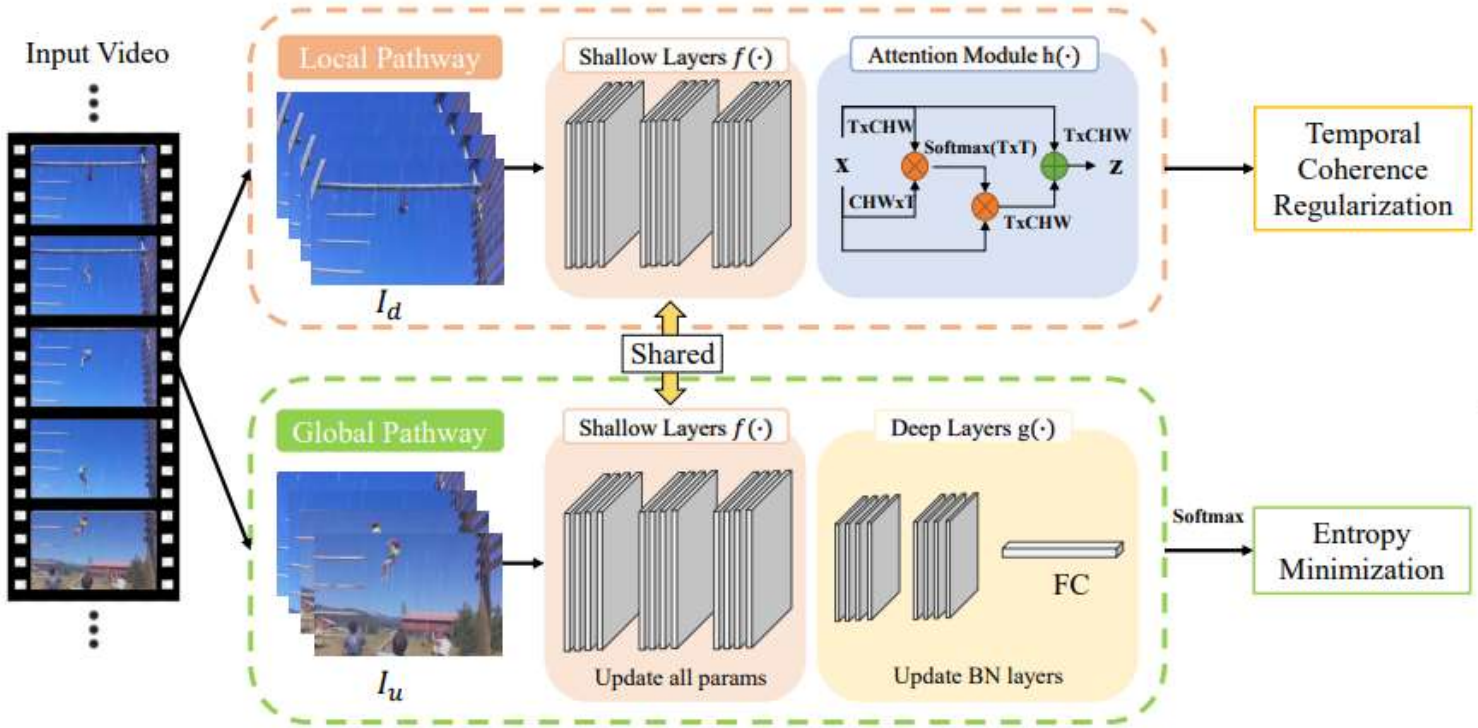
Hendrycks, ICLR'19

Yi, NeurIPS'21

Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. ICLR, 2019.
Chenyu Yi, et al., Benchmarking the Robustness of Spatial-Temporal Models Against Corruptions, NeurIPS 2021

Our Solution-Exploring temporal coherence

- TeCo – A Test-Time Optimization Framework for Robust Video Classification
  - Entropy Minimization on Global Pathway (Input from Uniform Sampling)
  - Apply Temporal Coherence Regularization on Local Path Way (Input from Dense Sampling)



$$L_{att-co} = \sum_{t=1} ||h(f(I_d))^t - h(f(I_d))^{t-i}||_1,$$

Overall Objective:

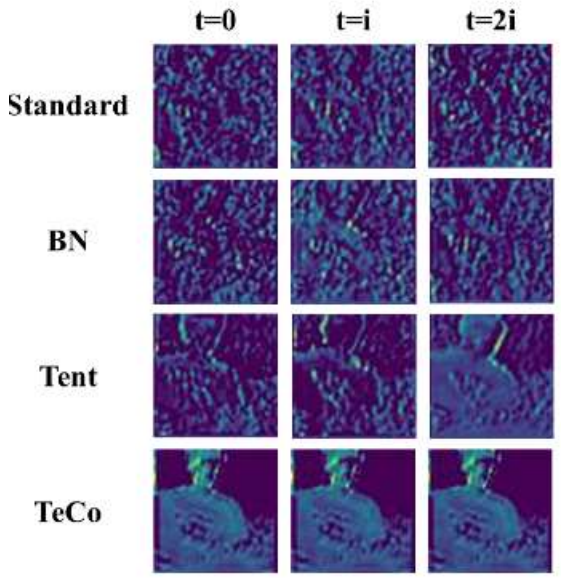
$$L = L_{ent}(I_u, f, g) + \beta L_{att-co}(I_d, f, h)$$

Results

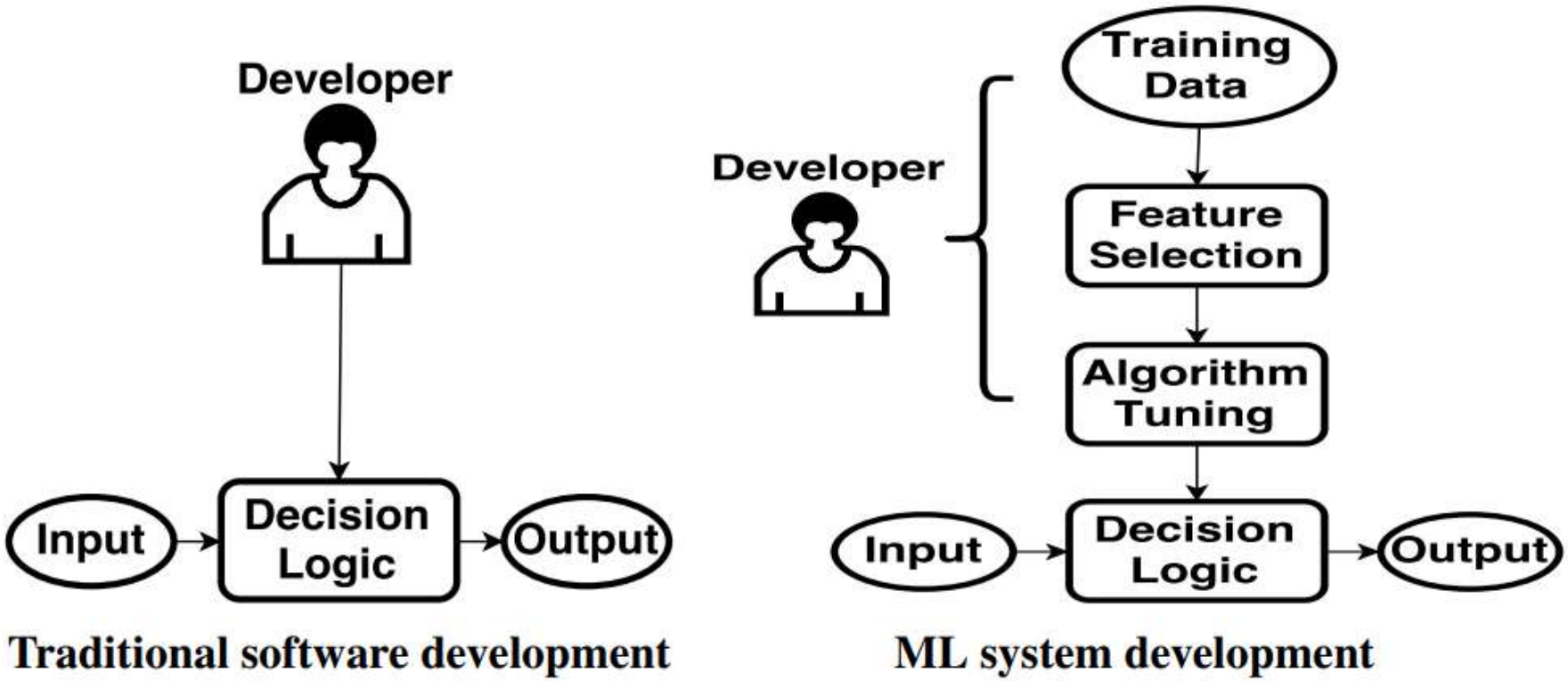
- TeCo outperforms other test-time optimization methods across architectures and datasets
- TeCo generates smoother feature maps

Table 1: mPC across architectures on Mini Kinetics-C and Mini SSV2-C. TeCo outperforms other baseline methods on different architectures and datasets. **Clean Acc** is the accuracy of model tested on clean data.

	Backbone	Clean Acc	Standard	BN	Tent	SHOT	TTT*	TeCo
Mini Kinetics-C	3D ResNet18	61.7	49.4	50.6	53.9	52.6	54.6	**56.9**
	ResNet18	66.0	51.6	53.0	55.6	53.2	59.5	**60.8**
	TAM-ResNet18	68.5	55.9	57.1	53.8	58.4	62.2	**63.4**
	MViTv2-S	84.4	77.9	78.0	79.2	78.2	78.0	**80.1**
Mini SSV2-C	3D ResNet18	52.2	39.3	40.0	39.9	42.4	41.5	**45.7**
	ResNet18	30.2	20.0	20.3	22.5	23.8	22.2	**24.5**
	TAM-ResNet18	55.5	44.2	45.0	44.5	45.2	46.7	**49.4**
	MViTv2-S	56.8	48.1	48.1	48.4	48.2	48.1	**48.5**



Deep Learning vs. Software Engineering



Ref from: [Pei, SOSP'17]

Software Quality Assurance

Code / Logic not covered during testing → Bugs may exist
Full Coverage Testing Needed !

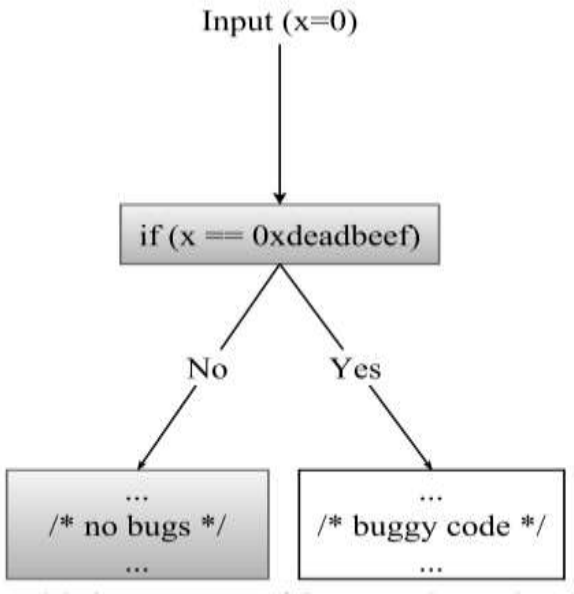
Data flow Analysis

- Code coverage
- Statement coverage
- Function coverage

Control flow Analysis

- Brach coverage
- Conditions coverage
- …….

Code Coverage



DNN Quality Assurance

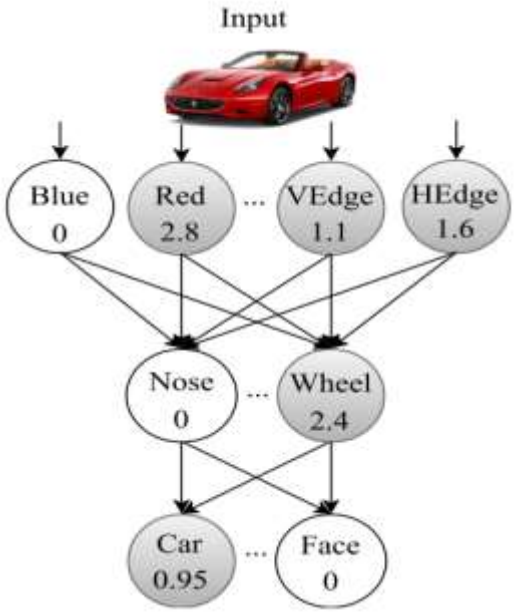
Accuracy based on standard test dataset is not sufficient.

DeepXplore: Automated Whitebox Testing of Deep Learning Systems [SOSP'17]

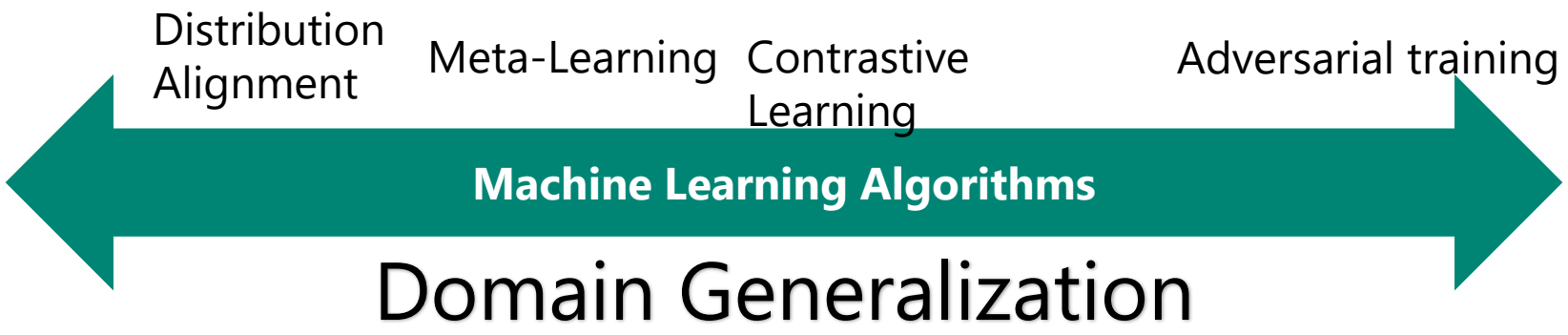
Neuron Coverage for DNN testing

- Deep-Gauge [FSE'18]
- Deep-Test [ICSE'18]
- Deep-Hunter [ISSTA'19]
- …….

Neuron Coverage

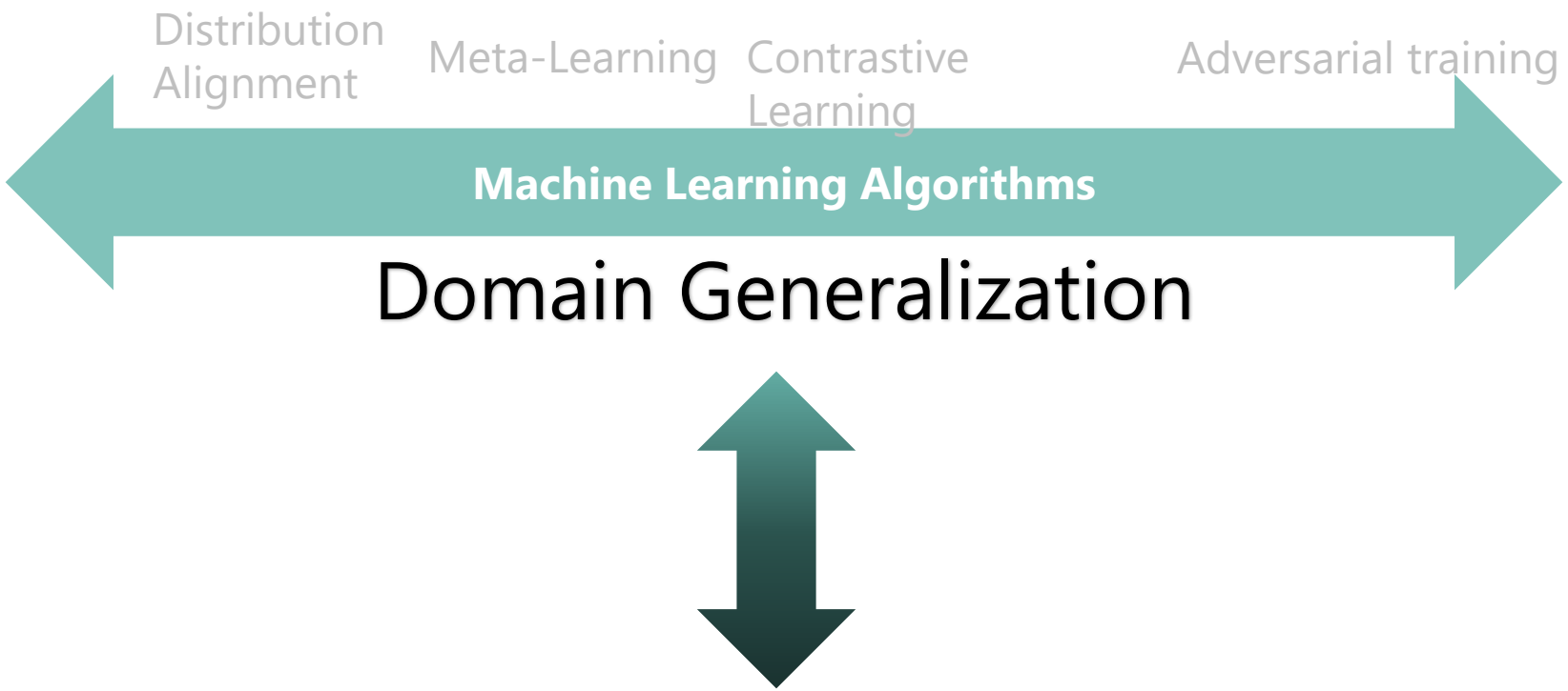


Neuron-Coverage Guided Domain Generalization



- C. X. Tian, Haoliang Li, et al., Neuron Coverage-Guided Domain Generalization, TPAMI2022

Neuron-Coverage Guided Domain Generalization



Treating the deep neural network as a software system

- C. X. Tian, Haoliang Li, et al., Neuron Coverage-Guided Domain Generalization, TPAMI2022

Neuron-Coverage Guided Domain Generalization

Neuron Coverage Guided Training

Neuron Coverage in DNN Testing → DNN Training

A neuron is inactive
during the **WHOLE** training process.

Once it get activated during evaluation,
errors may happen.

Data Flow in *software*
Code/Statement coverage

⇨

Data Flow in *DNN*
Neuron coverage

Neuron-Coverage Guided Domain Generalization

Neuron Coverage Loss

Activated Neuron: Normalized output value ≥ threshold t

When a New Epoch Starts

1. Set all neurons activation status as False
2. Each iteration: Keep track of inactivate neurons
3. Add up their outputs to form *Coverage Loss Term* (weighted by **λ**)
4. Maximize Coverage Loss Term in the next iteration.

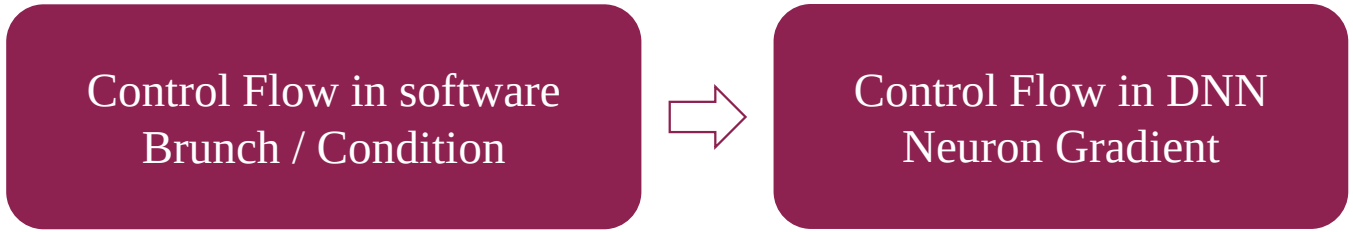
$$\mathscr{L}_{cov} = \mathscr{L}_{task} - \lambda \sum_i \sum_j out(\mathbf{x}, n_i^j)$$

Neuron-Coverage Guided Domain Generalization

Neuron Gradient

Neuron Coverage Isn’t Enough

- Neuron Activation → Binary Information (no layer interaction)



Neuron-Coverage Guided Domain Generalization

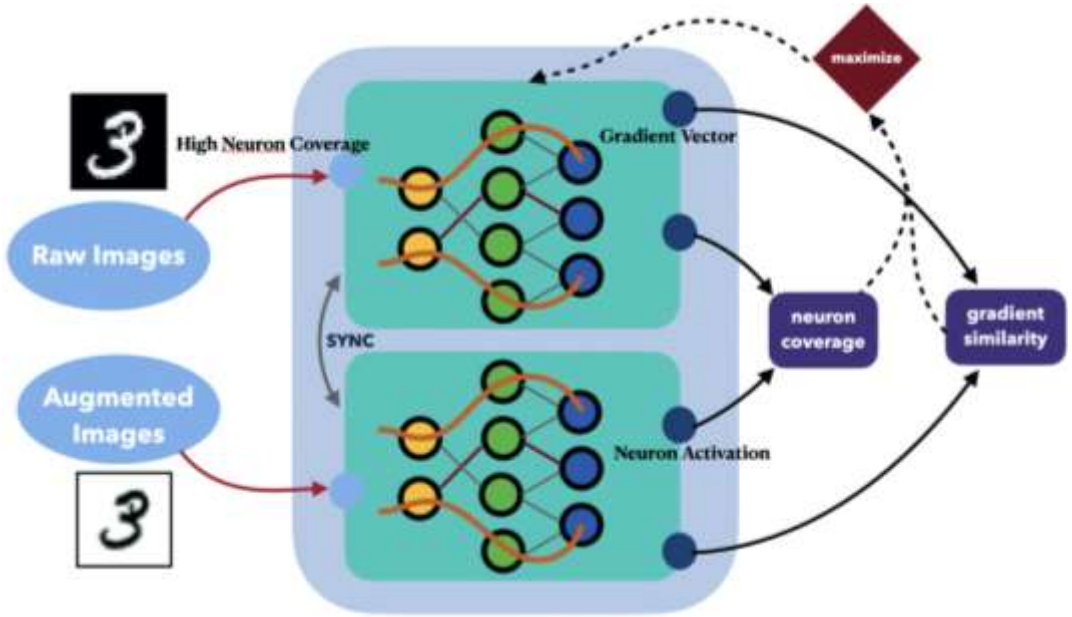
Neuron Gradient Similarity

Origin Sample: $\mathcal{D}_s$

Augmented Sample: $\hat{\mathcal{D}}_s$

Same semantics

Similar Control Flow



Similarity Loss Term (weighted by β)

$$\mathcal{L}_{sim}(f) = \left\| \frac{\partial L_{cov}(f, \mathcal{D}_S)}{\partial \Theta} - \frac{\partial L_{cov}(f, \hat{\mathcal{D}}_S)}{\partial \Theta} \right\|_2$$

Final Loss Function:

$$\mathcal{L}_{N-SimCov} = \mathcal{L}_{cov}(f, \mathcal{D}_S) + \mathcal{L}_{cov}(f, \hat{\mathcal{D}}_S) + \beta \mathcal{L}_{sim}(f, \mathcal{D}_S, \hat{\mathcal{D}}_S)$$

NCDG: Results

SSDG (Single Source Domain Generalization)

Object classification (PACS)

Source Domain	Method	Photo	Art_painting	Cartoon	Sketch	Avg.
Photo	DeepAll	/	66.0	26.7	35.0	42.5
	JiGen	/	64.1	23.9	32.9	40.3
	GUD	/	55.8	**33.3**	45.6	44.9
	M-ADA	/	64.3	29.8	35.2	43.1
	N-SimCov	/	**68.8**	29.8	**48.6**	**49.0**
Art_painting	DeepAll	**96.5**		60.6	52.5	69.9
	JiGen	95.5	/	60.1	50.2	68.6
	GUD	93.7	/	61.1	56.2	70.4
	M-ADA	95.0	/	61.5	47.6	68.0
	N-SimCov	95.0	/	**68.6**	**66.4**	**76.6**
Cartoon	DeepAll	87.4	67.6	/	68.3	74.5
	JiGen	85.1	65.5	/	65.7	72.1
	GUD	**86.5**	67.2	/	68.5	73.1
	M-ADA	83.1	66.4	/	66.3	71.9
	N-SimCov	85.8	**71.6**	/	**71.9**	**76.4**
Sketch	DeepAll	42.0	32.2	54.2	/	42.8
	JiGen	47.2	35.5	51.8	/	44.8
	GUD	32.9	23.1	37.5	/	31.2
	M-ADA	36.9	22.0	42.6	/	33.9
	N-SimCov	**47.7**	**41.3**	**60.4**	/	**49.8**

Segmentation (GTA5-Cityscape)

Performance comparisons on cross-domain semantic image segmentation on DeepLabv3+ with Resnet50 as backbone.

Method	Baseline	IBN	ISW
w/o coverage	29.7	33.9	36.6
w/ coverage	**34.7**	**35.6**	**37.2**

NCDG: Results

MSDG (Multiple Source Domain Generalization)

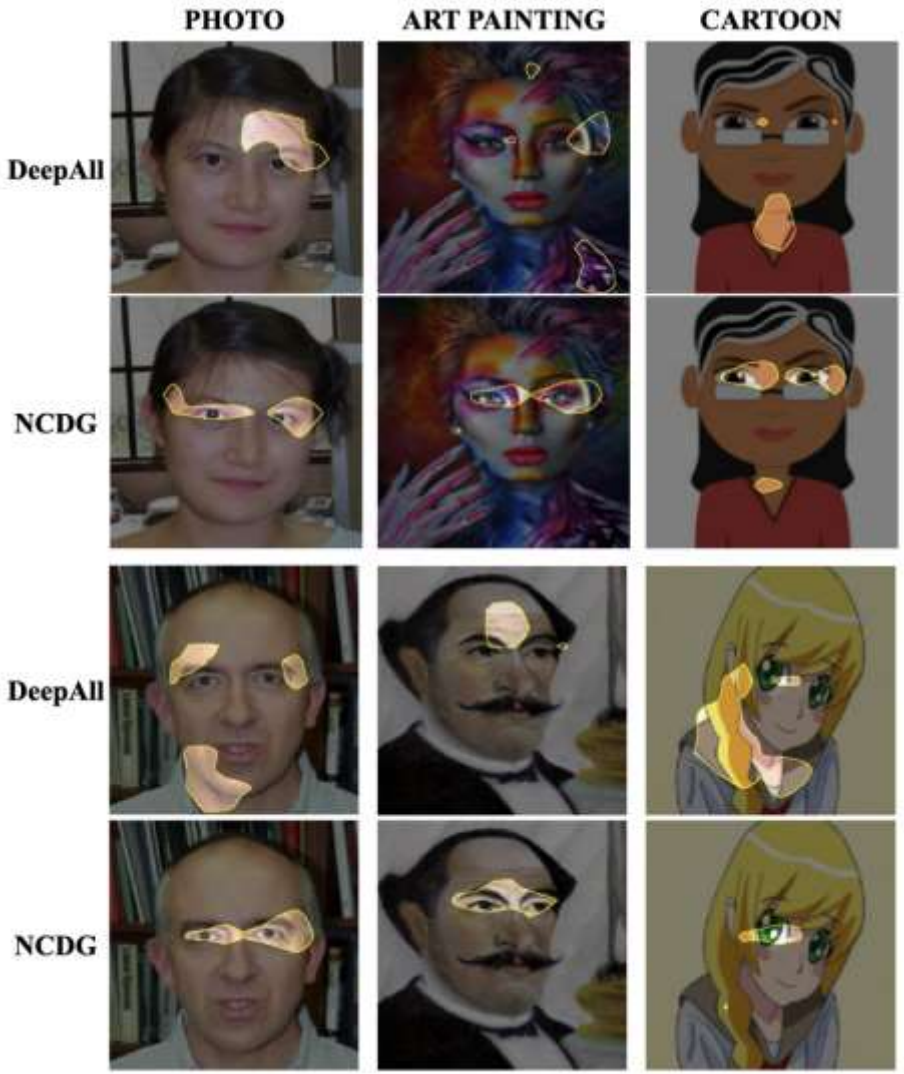
Leave-one-domain-out experimental protocol

Office-Home	Art	Clipart	Product	Real-World	Avg.
D-SAM	58.0	44.4	69.2	71.5	60.8
JiGen	53.0	47.5	71.5	72.8	61.2
L2A-OT	**60.6**	50.1	74.8	**77.0**	65.6
DDAIG	59.2	52.3	74.6	76.0	65.5
RSC	58.4	47.9	71.6	74.5	63.1
N-SimCov	59.8	**53.1**	**75.3**	76.3	**66.1**

VLCS	Caltech	LabelMe	Pascal	Sun	Avg.
TF	93.6	63.4	70.0	61.3	72.1
MMDS-AAE	94.4	62.6	67.7	64.4	72.3
D-SAM	91.8	57.0	58.6	60.8	67.0
JiGen	96.9	60.9	70.6	64.3	73.2
RSC	**97.6**	61.9	**73.9**	68.3	75.4
N-SimCov	97.2	**67.6**	70.7	**68.7**	**76.1**

PACS	Photo	Art painting	Cartoon	Sketch	Avg.
AlexNet					
D-SAM	85.6	63.9	69.4	64.7	71.2
JiGen	89.0	67.6	71.7	65.2	73.4
RSC	**90.9**	**71.6**	**75.1**	66.6	76.1
N-SimCov	89.0	68.9	74.7	**72.9**	**76.4**
Resnet-18					
D-SAM	95.3	77.3	72.4	77.8	80.7
JiGen	96.0	79.4	75.3	71.4	80.5
L2A-OT	**96.2**	83.3	78.2	73.6	82.8
DDAIG	95.3	**84.2**	78.1	74.7	83.1
RSC	96.0	83.4	80.3	80.9	85.2
N-SimCov	95.4	82.3	**82.3**	**82.1**	**86.2**

NCDG: Visualization



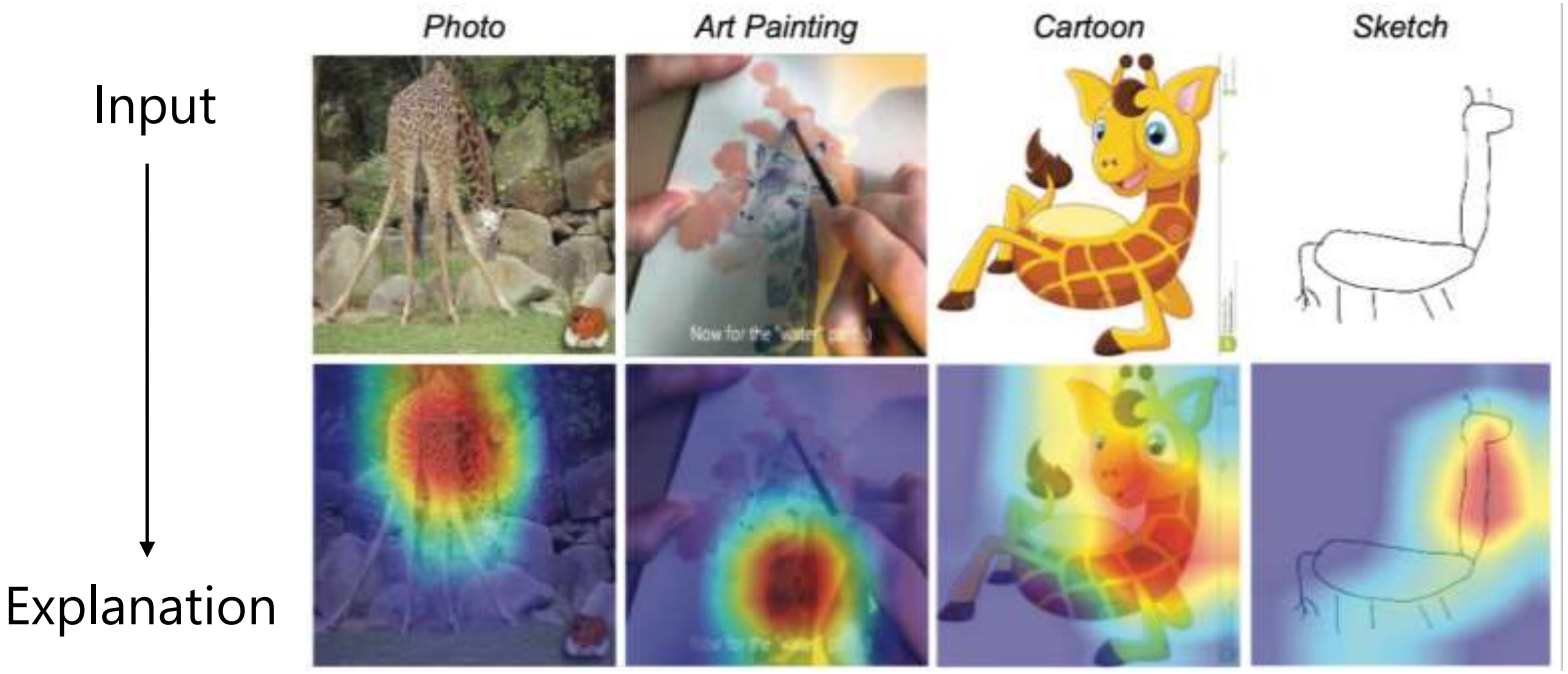
Software Coverage ⟷ **AI Explainability**

- Visualization through network dissection

[Bau2020] D. Bau, J.-Y. Zhu, H. Strobelt, A. Lapedriza, B. Zhou, and A. Torralba, "Understanding the role of individual units in a deep neural network," PNAS, 2020.

Explanation for Model Generalization

Shared causality across domains.

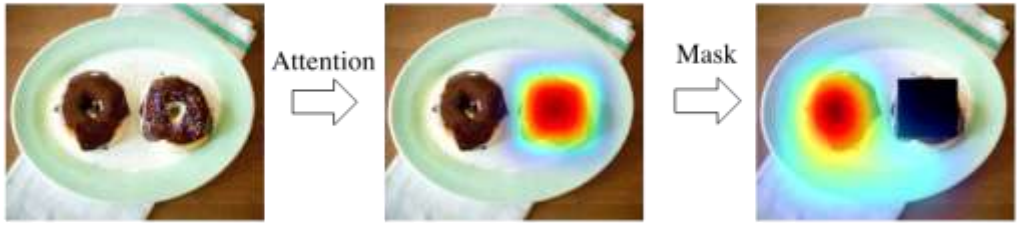


But can we believe such appealing visualizations?

Explainability through Attention

Do positive attention weights indicate contribution effects? No!

Question1: What are colorful pieces on the doughnut?



Pred: powder
(Confidence 16%↓) ✓

Question2: What is the girl eating?

Pred: donut
(Confidence 12%↑) ✗

How to evaluate the explanation faithfulness?

Evaluating two properties in explanation weights

– Importance Correlation:

Magnitude <-> Feature Importance

– Polarity Consistency:

Sign <-> Polarity of Feature Impact

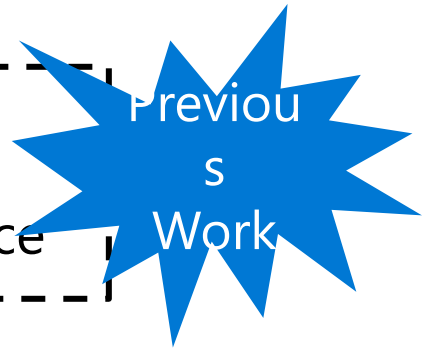
- Yibing Liu, Haoliang Li, et al., Rethinking Attention-model Explainability through Faithfulness Violation Test, ICML'22

How to evaluate the explanation faithfulness?

Evaluating two properties in explanation weights

– Importance Correlation:

Magnitude <-> Feature Importance



– Polarity Consistency:

Sign <-> Polarity of Feature Impact

- Yibing Liu, Haoliang Li, et al., Rethinking Attention-model Explainability through Faithfulness Violation Test, ICML'22

How to evaluate the explanation faithfulness?

Evaluating two properties in explanation weights

– Importance Correlation:

Magnitude <-> Feature Importance

– Polarity Consistency:

Sign <-> Polarity of Feature Impact

- Yibing Liu, Haoliang Li, et al., Rethinking Attention-model Explainability through Faithfulness Violation Test, ICML'22

Our Solution: Faithfulness Violation Test

Idea: measure the ratio of test samples violating polarity consistency.

Steps: given a test sample x and an explanation method $w(\cdot)$:

1. Find the most influential feature $x^* = \text{argmax}_{x_i \in x} ||w(x_i)||$.
2. Estimate the feature impact of x^* based on the perturbation test

$$\Delta \text{C}(x, x^*) = f(x)_{\hat{y}} - f(x \backslash x^*)_{\hat{y}}.$$

3. Check if the explanation weight aligns with the feature impact.

$$\text{Violation} = \mathbb{1}_{\text{sign}(w(x^*) \cdot \Delta \text{C}(x, x^*)) < 0}$$

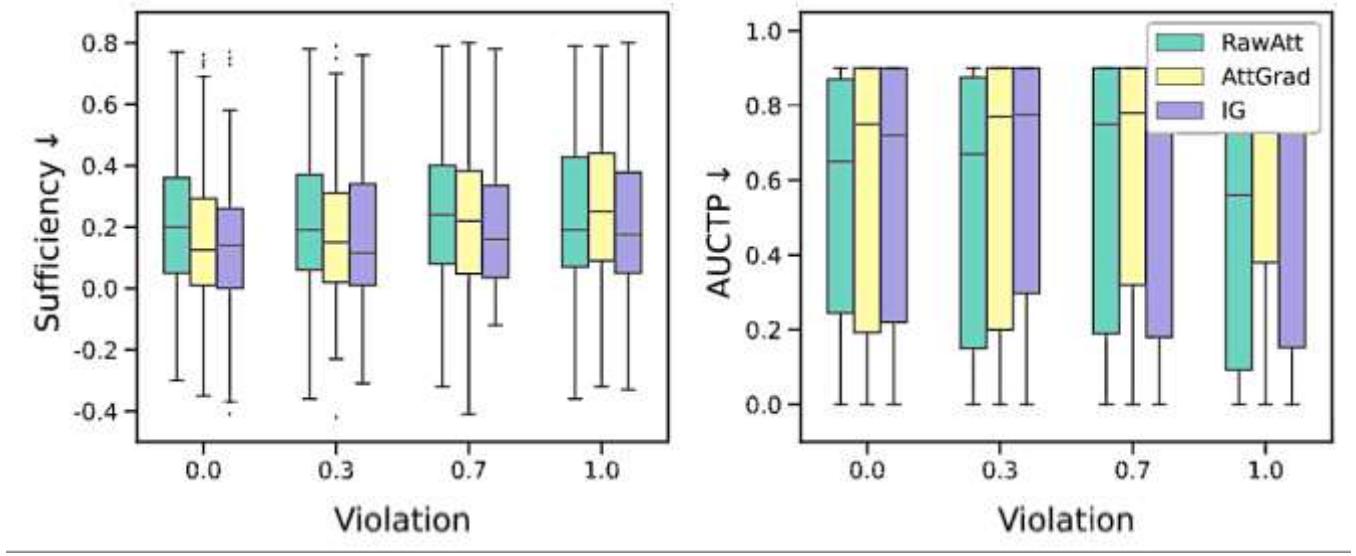
Experiments

- RQ1: Why we need the faithfulness violation test?
- RQ2: How existing methods perform on faithfulness?
- RQ3: What factors dominate the faithfulness violation issue?

Method	Denoted	Basis
Generic attention-based explanation methods		
Inherent Attention Explanation	RawAtt	α
Attention $\odot$ Gradient	AttGrad	$\alpha \odot \nabla\alpha$
Attention $\odot$ InputNorm	AttIN	$\alpha \odot \|\|v(x)\|\|$
Transformer-based explanation methods		
Partial LRP	PLRP	R^{α}
Attention Rollout	Rollout	α
Transformer Attention Attribution	TransAtt	$\nabla\alpha \odot \mathrm{R}^{\alpha}$
Generic Attention Attribution	GenAtt	$\alpha \odot \nabla\alpha$
Gradient-based attribution methods		
Input $\odot$ Gradient	InputGrad	$x \odot \nabla x$
Integrated Gradients	IG	$x \odot \nabla x$

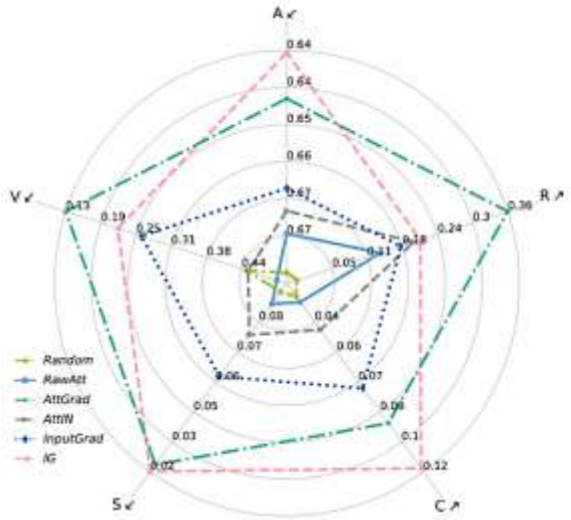
Comparison with Existing Metrics (RQ1)

Existing metrics are incapable of examining the polarity consistency!

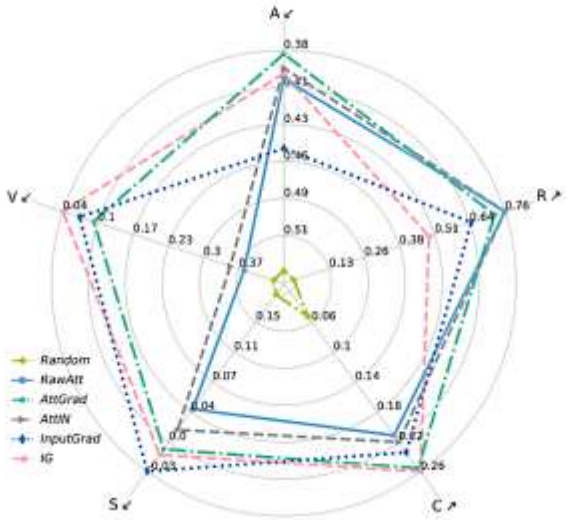


Sanity Faithfulness Evaluation (RQ2)

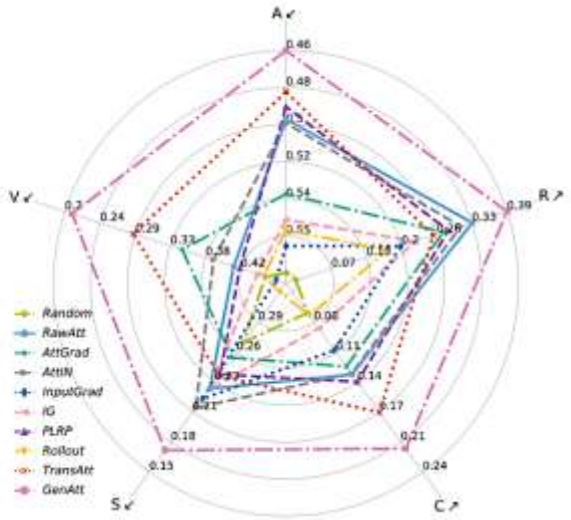
Most tested explanation methods suffer from the faithfulness violation issue regarding polarity consistency.



(a) LSTM+DotAtt on QQP dataset



(b) BUTD on VQA 2.0 dataset



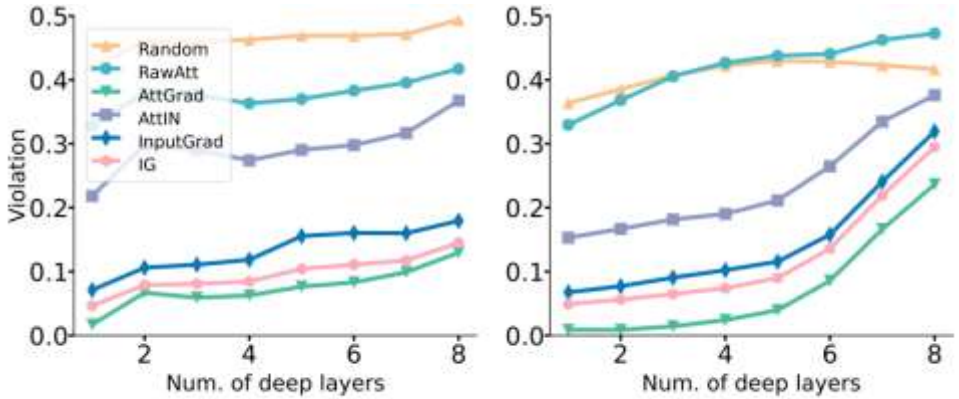
(c) LXMERT on GQA dataset

Factor Analysis (RQ3)

Two dominant factors

- The capability to identify polarity
- The complexity of model architectures

Method	Yelp	AgNews	VQA 2.0
α	0.31	0.28	0.40
$\alpha \odot \nabla\alpha$	**0.02**	**0.03**	**0.06**
$\alpha \odot \lvert\nabla\alpha\rvert$	0.15	0.07	0.25
$\alpha \odot \text{sign}(\nabla\alpha)$	0.16	0.18	0.27



(a) Yelp dataset (b) SST dataset

Challenges

- Continuous domain generalization
  - Continuous / online learning
- Generalize to novel categories
  - New categories instead of closed set
- Interpretable domain generalization
  - Learning to interpret: why it can generalize?
- Large-scale pre-training / self-learning and DG
  - The role of pre-training and self-learning with DG
- Performance evaluation
  - Develop more fair and application-driven evaluation standards

Conclusion

General ML —Non-IID→ Domain adaptation —Unseen target→ Domain generalization

DG Roadmap

- Introduction and background
- Relation with existing area: transfer learning, domain adaptation, multi-task learning…
- Algorithm
  - Data manipulation: augmentation, or generation
  - Representation learning: domain-invariant learning, disentanglement
  - Learning strategy: meta-learning, ensemble learning, gradient, DRO, SSL…
- Applications: CV, NLP, RL, medical…
- Datasets, benchmark, evaluation
- Theory, Connection to explainability, and future challenges

香港城市大學
City University of Hong Kong

Thank You

haoliang.li@cityu.edu.hk

專業 創新 胸懷全球
Professional · Creative
For The World