THE UNIVERSITY *of* EDINBURGH
informatics

SAMSUNG Research

Domain Adaptation for Visual Applications Part 4: Perspectives and Outlook

Timothy Hospedales

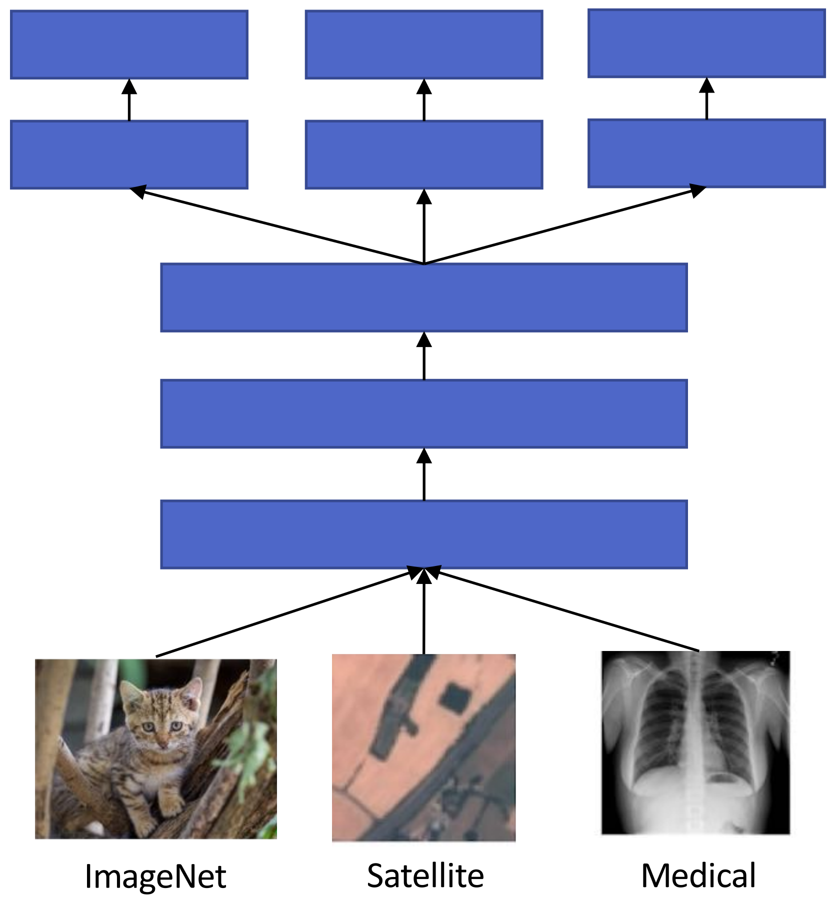
Machine Intelligence Group – University of Edinburgh

Machine Learning & Data Intelligence – Samsung AI Centre, Cambridge

Outline – Perspectives & Outlook

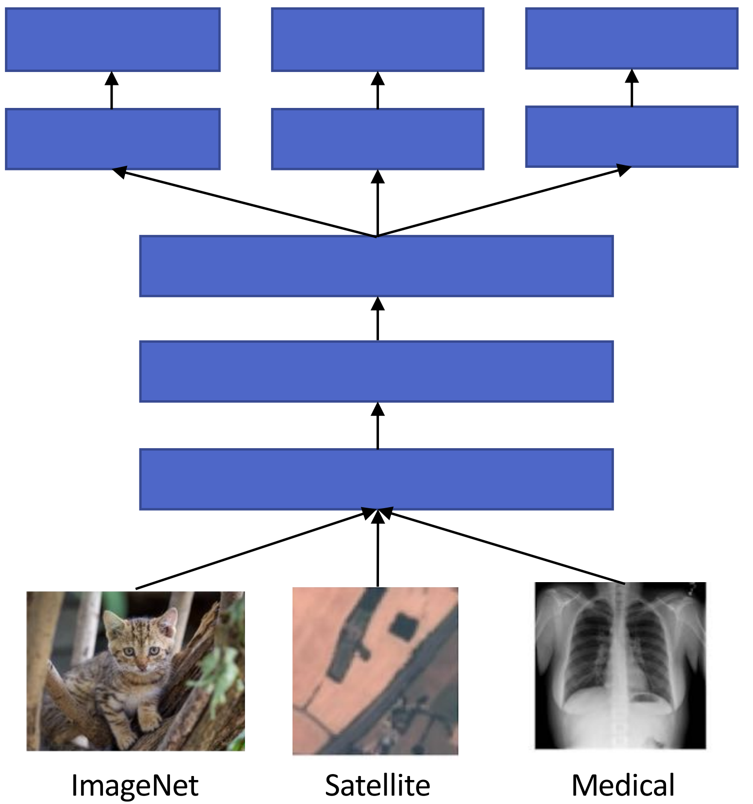
- Multi-Domain Learning & Tensor Methods
- Meta-Learning for DA and DG
- Emerging Problem Areas and Applications

What is Multi-Domain Learning?



Why Multi-Domain Learning?

- Save memory
- Multi-task effect
- Multi-source domain adaptation (?)
- Incremental learning (?)

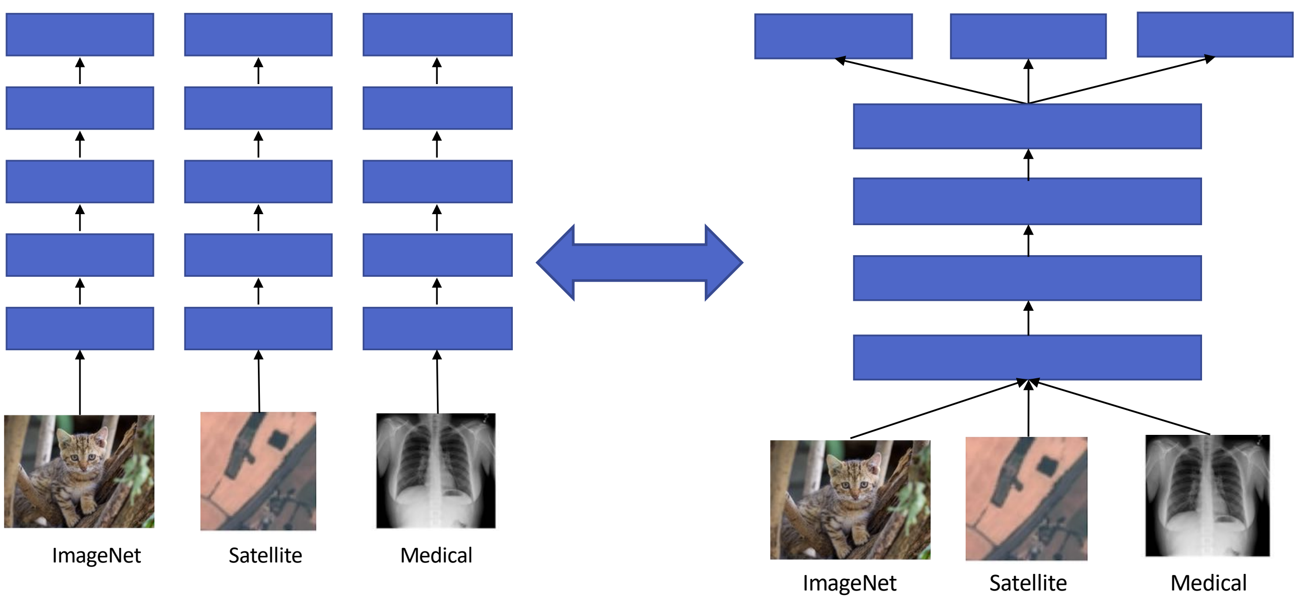


MDL Methods

How to manage sharing/separation?

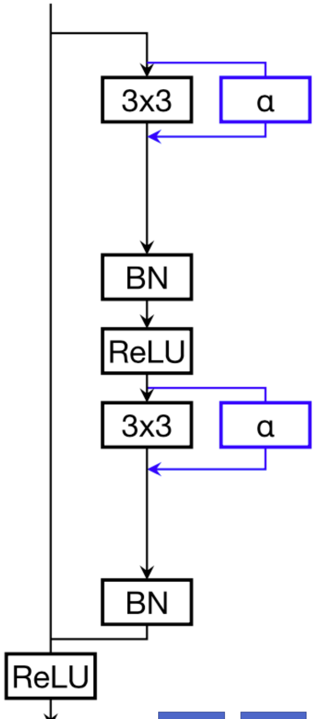
- Spectrum between:

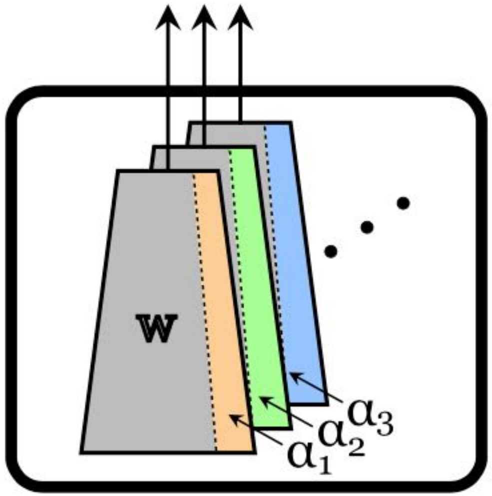
Independent Networks ⟷ Maximally Shared



Many Methods:
- Rebuffi, NIPS-17, CVPR-18
- Yang, ICLR-15, ICLR-17, ECCV-18
- Bulat, AAAI-20, CVPR-20
- Bragman, ICCV-19
- Kanakis, ECCV-20

Residual Adapters





+ Small number of parameters per domain
+ More knowledge transfer than single-task nets
+ More expressive than simple multi-head architecture

Residual Adapter:

$$\boldsymbol{y} = \boldsymbol{w} * \boldsymbol{x} + \boxed{\mathrm{diag}(\boldsymbol{\alpha}) * \boldsymbol{x}}$$

Next layer

Normal convolution

Input

1x1 convolution

ResNet Block w/ Adapter

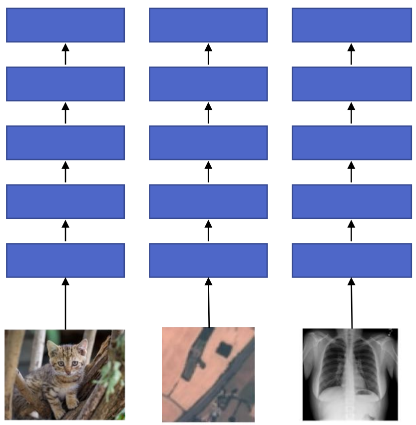
$$\boldsymbol{y} = \boldsymbol{x} + f_{\boldsymbol{w}}(\boldsymbol{x}) + \boxed{h_{\boldsymbol{\alpha}_d}(\boldsymbol{x})}$$

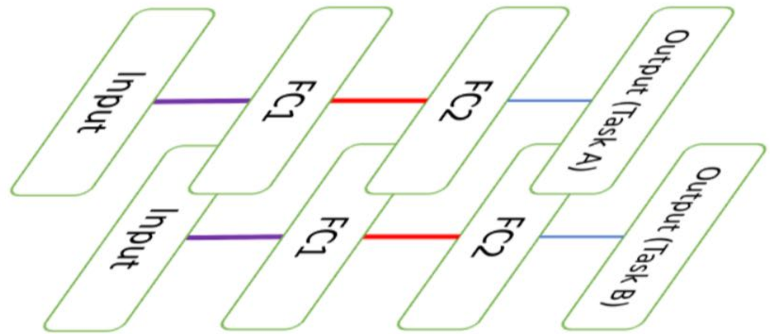
Normal block

Adapter module
For current domain *d*

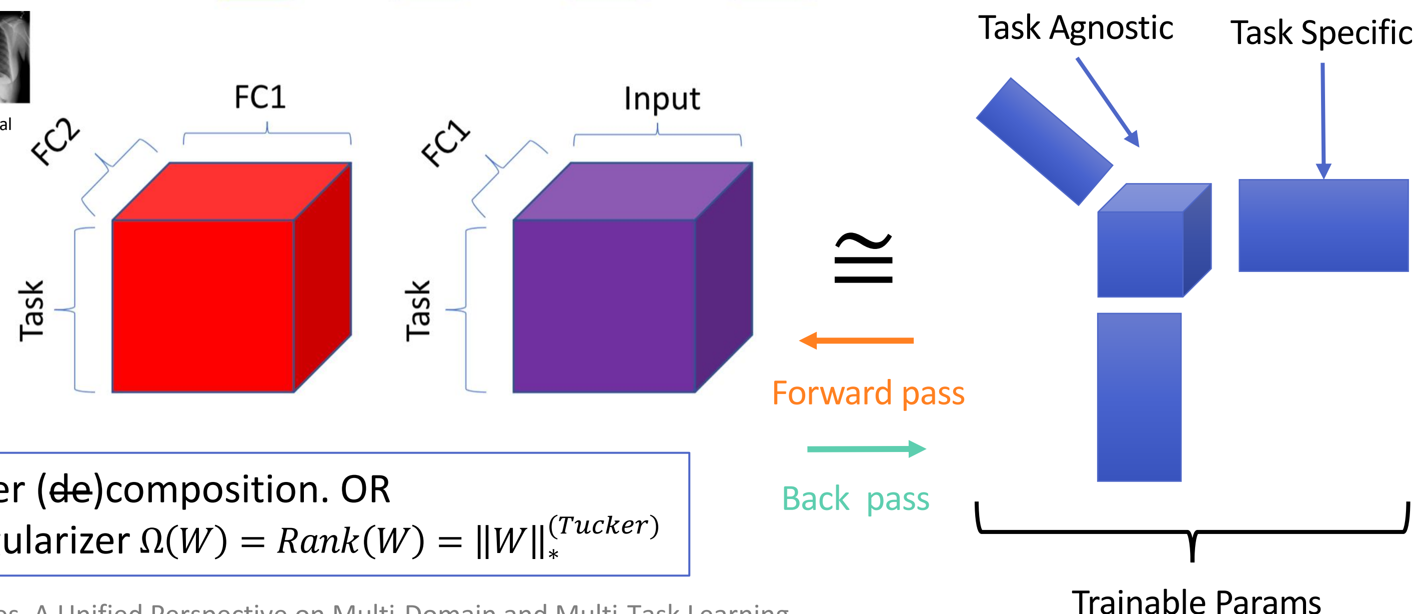
[Rebuffi, Learning multiple visual domains with residual adapters, NIPS-17;
Rebuffi, Efficient parametrization of multi-domain deep neural networks, CVPR-18]

Low-Rank Tensor Methods





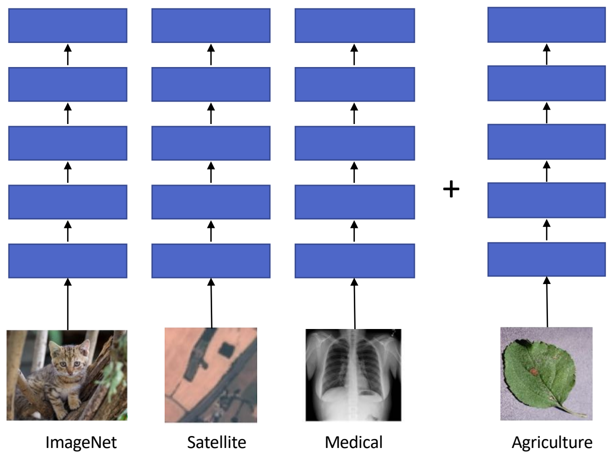
+ Few parameters per domain
+ More knowledge transfer than single-task nets
+ More expressive than simple multi-head architecture
+ Easy to control share strength



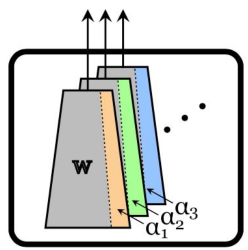
1. Explicit low-rank Tucker (~~de~~)composition. OR
2. Tensor trace norm regularizer $\Omega(W) = Rank(W) = \|W\|_*^{(Tucker)}$

[Yang & Hospedales, A Unified Perspective on Multi-Domain and Multi-Task Learning,
Yang & Hospedales, Deep Multi-task Representation Learning: A Tensor Factorisation Approach, ICLR-17]

Incremental Learning

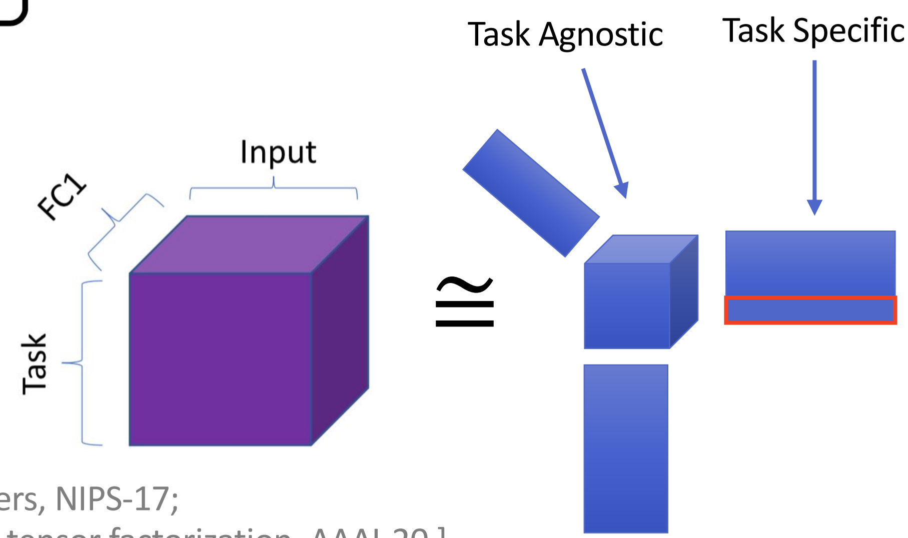


$$y = x + f_w(x) + h_{\alpha_d}(x)$$



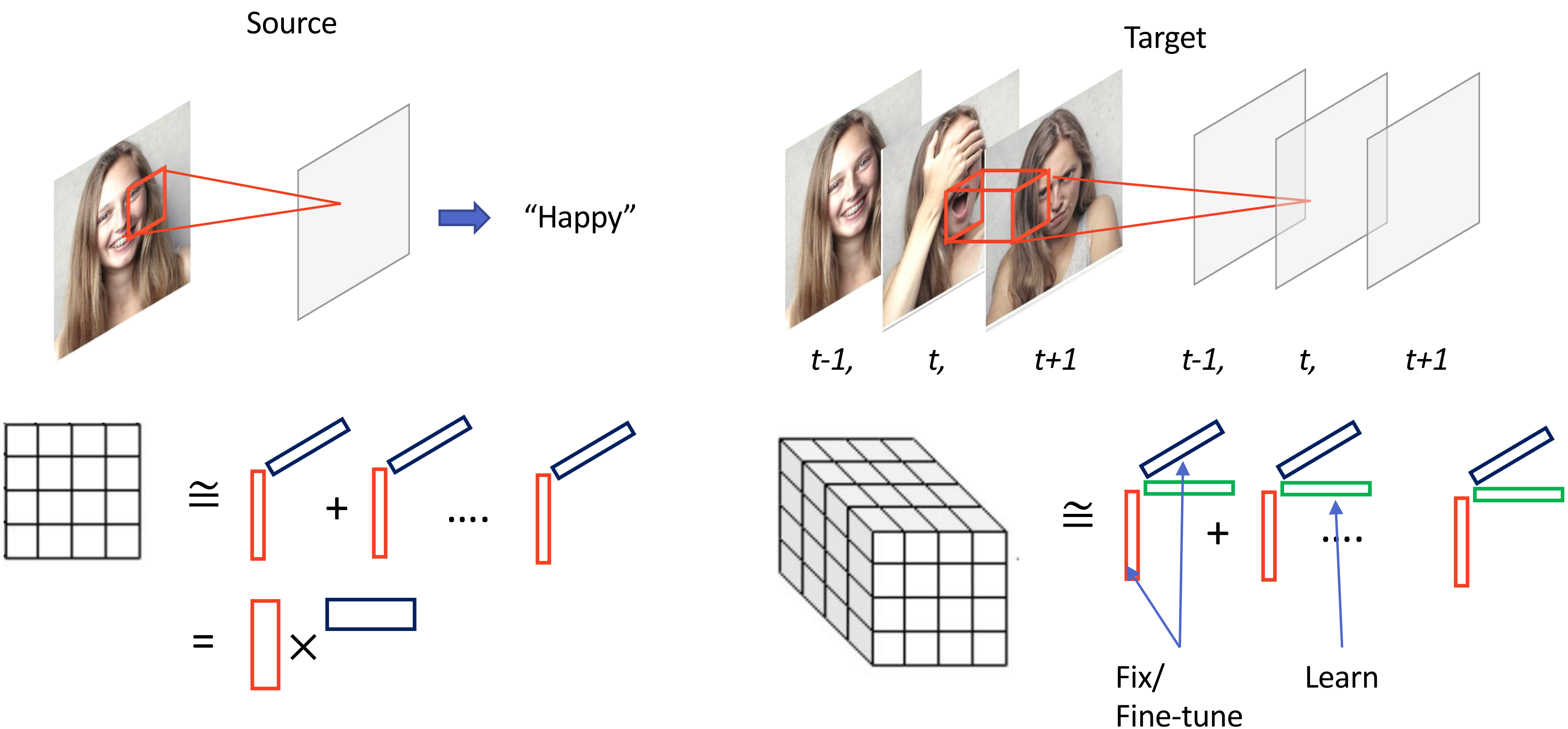
RA: Simply learn one new α_d vector. Keep others fixed.

Tensor: Simply learn one new task vector.
Keep others task-agnostic factors fixed.



[Rebuffi, Learning multiple visual domains with residual adapters, NIPS-17;
Bulat, Incremental multi-domain learning with network latent tensor factorization, AAAI-20]

Tensor Methods Example:
Adaptation from Image to Video Recognition



[Bulat, CVPR'20, Factorized Higher-Order CNNs..]

Latent Domain Learning

- MDL frameworks so far require domain labels.
- Open question: Data without domain annotation?
- Potential Solution: Dynamic Residual Adapters

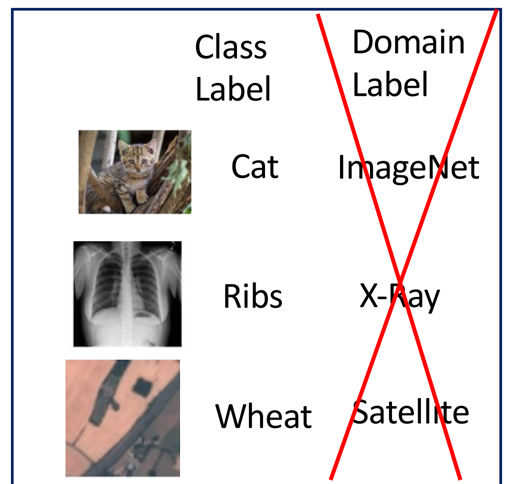
Residual Adapter

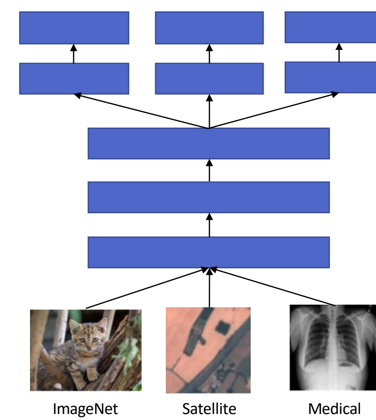
$$\boldsymbol{y} = \boldsymbol{x} + f_{\boldsymbol{w}}(\boldsymbol{x}) + h_{\boldsymbol{\alpha}_d}(\boldsymbol{x})$$

Dynamic R.A.

Recognize domain & activate adapter (sigmoid).

$$\boldsymbol{y} = \boldsymbol{x} + f_{\boldsymbol{w}}(\boldsymbol{x}) + \sum_d g_d(\boldsymbol{x})\, h_{\boldsymbol{\alpha}_d}(\boldsymbol{x})$$





[Deecke, arXiv:2006.00996, Latent Domain Learning with Dynamic Residual Adapters]

Summary & Outlook

Summary:

- Efficient knowledge sharing and incremental MDL.
- Domain-Incremental learning (without forgetting).

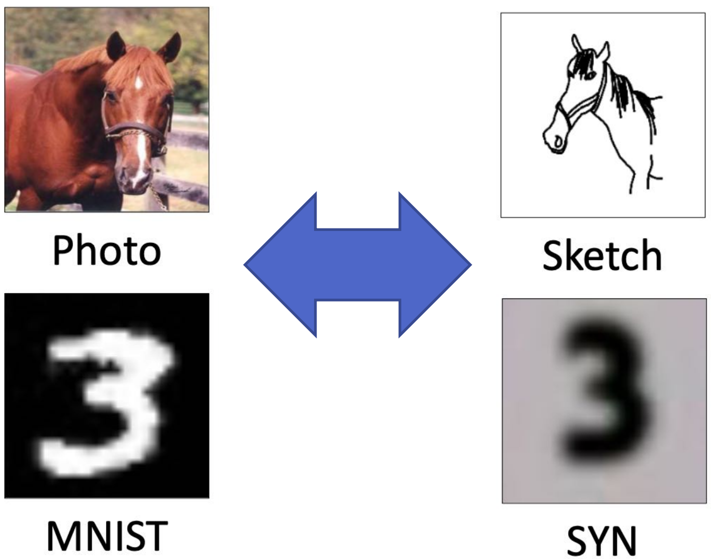
Open Questions:

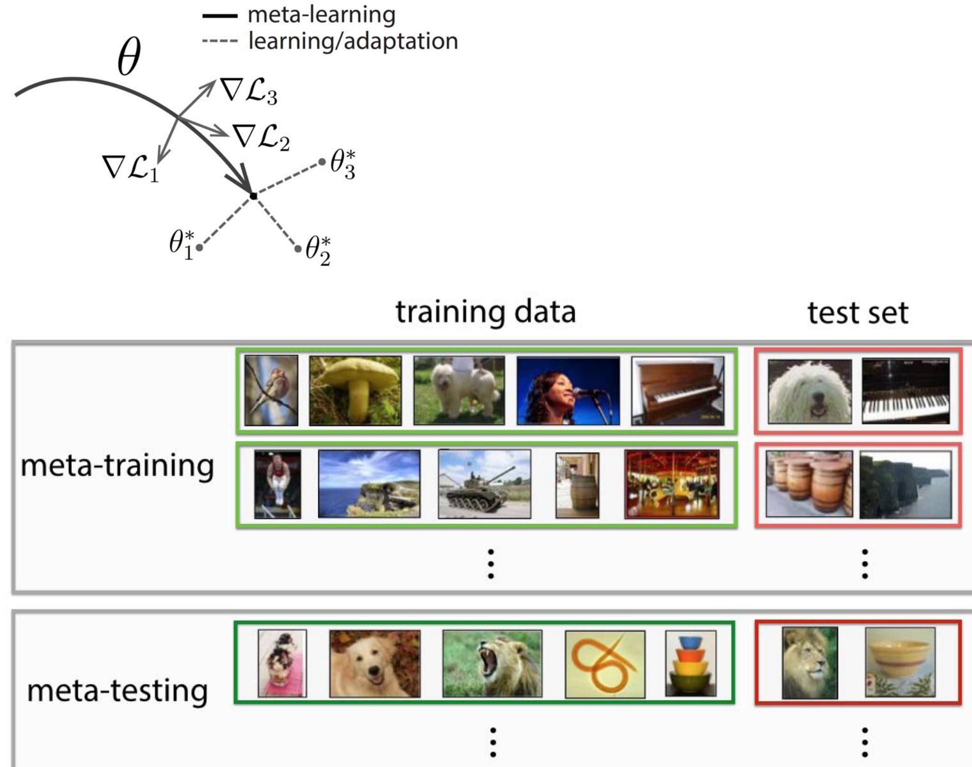
- Exploiting MDL in support of domain adaptation/generalisation
- Normalization layer design for MDL
- MDL (and multi-source DA) with latent domains

Outline – Perspectives & Outlook

- Multi-Domain Learning & Tensor Methods
- Meta-Learning for DA and DG
    - Meta-Learning Mini-Intro
    - Meta-Learning for DA and DG
- Emerging Problem Areas and Applications

What's the Connection?





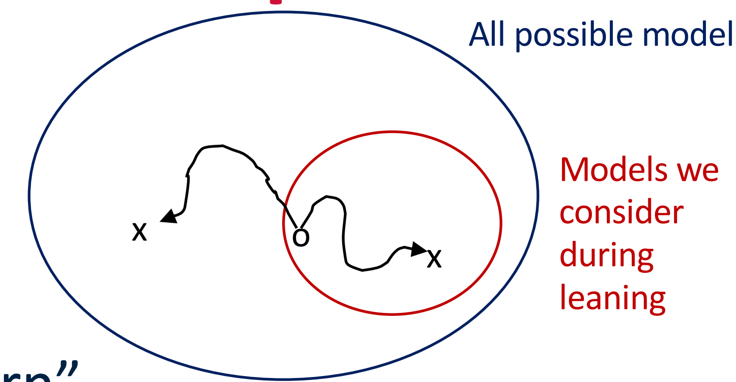
Domain Adaptation

[E.g. Csurka, Springer, 2017]

Meta-Learning

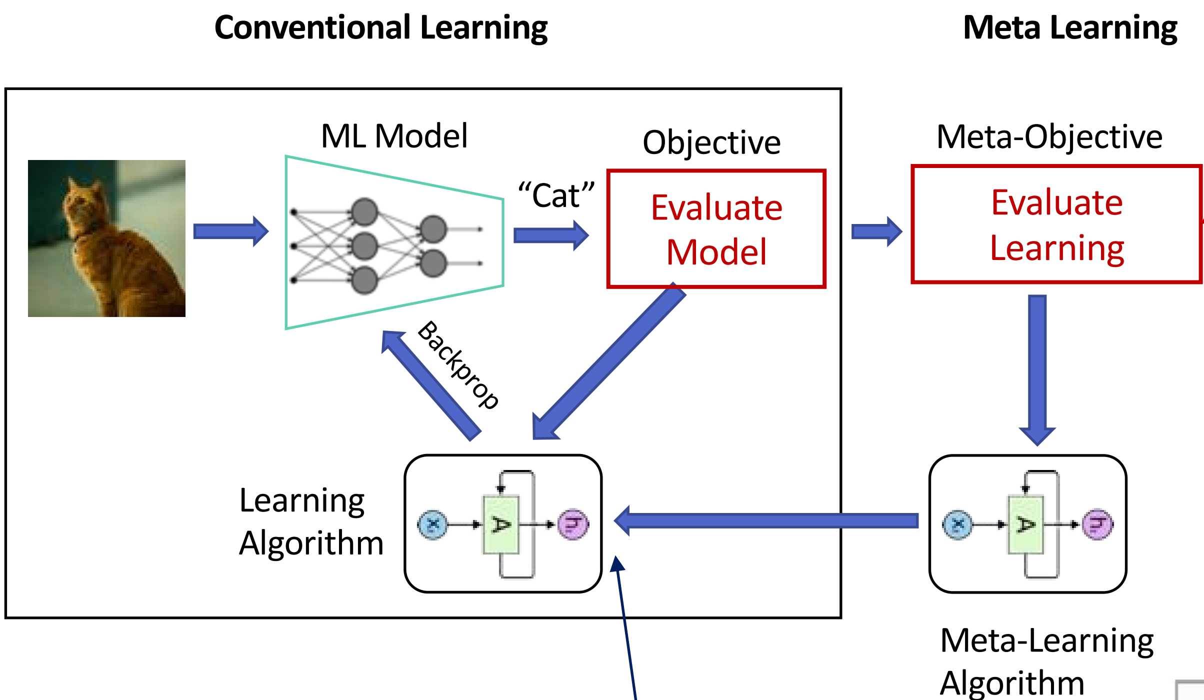
[Eg, Finn, MAML, ICML'17]

Meta-Learning Mini-Intro: Concept



- Meta-Learning Aka: "Learning-to-learn".
- Learning uses an `Inductive-Bias'
  - EG: Linearity, convolution, markov assumption.
  - EG: Regularization strength.
  - EG: Initial condition (non-convex optimization).
- Meta-learning:
  - => Searching for a good inductive bias

Meta-Learning Mini-Intro: Informal

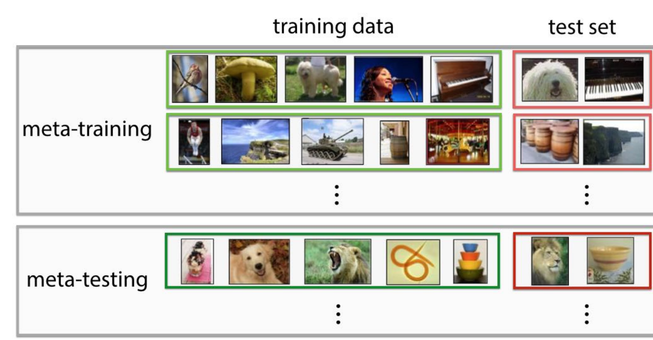


Common Meta-Objectives

- Validation performance
- Few-shot learning performance

Common "Meta Representation"

- Initial Condition
- Regularization Strength



Meta-Learning Mini-Intro: More Formal

- Conventional ML:
  - Dataset: $D = \{x_i, y_i\}_{i=1}^N$. Model: $y = f_\theta(x)$. Loss: $\mathcal{L}(D; \theta)$
  - Train: $\theta^* = \arg\min_\theta \mathcal{L}(D; \theta, \omega)$. Test: $y = f_{\theta^*}(x)$
- Meta-Learning:

Potential Task Distribution

Inductive Bias
Meta-Representation

$$\min_{\omega} \mathbb{E}_{D\sim p(D)} \mathcal{L}(D; \omega)$$

Meta-Train: $\omega^* = \arg\min_\omega \sum_t \mathcal{L}^+(D_t^{mtr}; \omega)$

Meta-Test: $\theta^* = \arg\min_\theta \mathcal{L}(D^{mte}; \theta, \omega^*)$

Test: $y = f_{\theta^*}(x)$

Meta-Train: Bilevel Optimization

$$\omega^* = \arg\min_\omega \sum_t \mathcal{L}^+\left(D_{t,val}^{mtr}; \theta_t^*, \omega\right)$$

$$\text{s.t. } \theta_t^* = \arg\min_\theta \mathcal{L}(D_{t,tr}^{mtr}; \theta_t, \omega)$$

Meta-Learning Mini-Intro: Schematic Algorithm

- Optimize validation loss $\mathcal{L}^{+}(D_{val};\omega)$ wrt meta representation ω :

Init: ω

Repeat:

Init: θ

Repeat:

Update Model θ
Wrt objective $\mathcal{L}$:

$$\theta = \theta - \alpha \nabla_{\theta} \mathcal{L}(D_{tr};\theta,\omega)$$

Update hyper-param ω
Wrt Meta-Objective $\mathcal{L}^{*}$

$$\omega = \omega - \beta \nabla_{\omega} \mathcal{L}^{+}(D_{val};\theta,\omega)$$

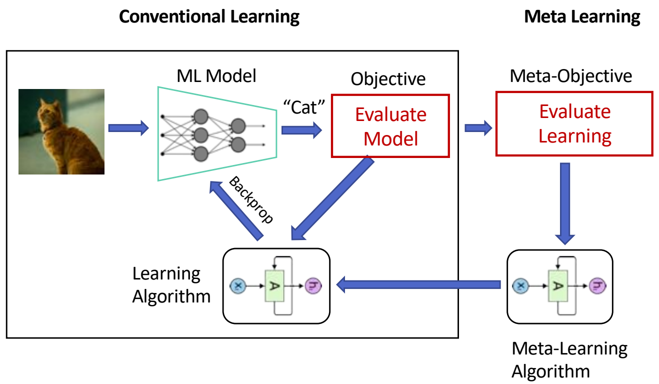
Depends on final state of θ from inner loop.
$\Rightarrow$ Backpropagation through inner optimization
$\Rightarrow$ Needs care to be tractable.

Summary

Define:

1. Meta-Representation ω to Learn
2. Base Objective & Data: $\mathcal{L}(D_{tr}; \theta, \omega)$
   - Optimize model θ conditional on meta knowledge ω.
3. Meta Objective & Data: $\mathcal{L}^{+}(D_{val};\ \omega)$
   - Optimize ω to achieve best model learning

For more information on meta-learning, see: Hospedales et al, "*Meta-Learning in Neural Networks: A Survey*", arXiv:2004.05439, 2020.

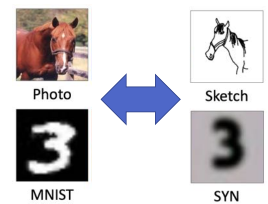


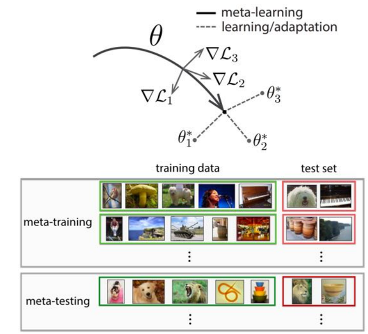
Outline – Perspectives & Outlook

- Multi-Domain Learning & Tensor Methods
- Meta-Learning for DA and DG
  - Meta-Learning Mini-Intro
  - Meta-Learning for DA and DG
- Emerging Problem Areas and Applications

Meta-Learning for Domain Adaptation/Generalisation

- Design Parameters of your favorite deep DA/DG algorithms?
  - Initial Condition (almost always)
  - Regularizer (often)
  - Neural architecture (often)
  - Learning rate (often)
  - Multiple-loss weighting (sometimes)





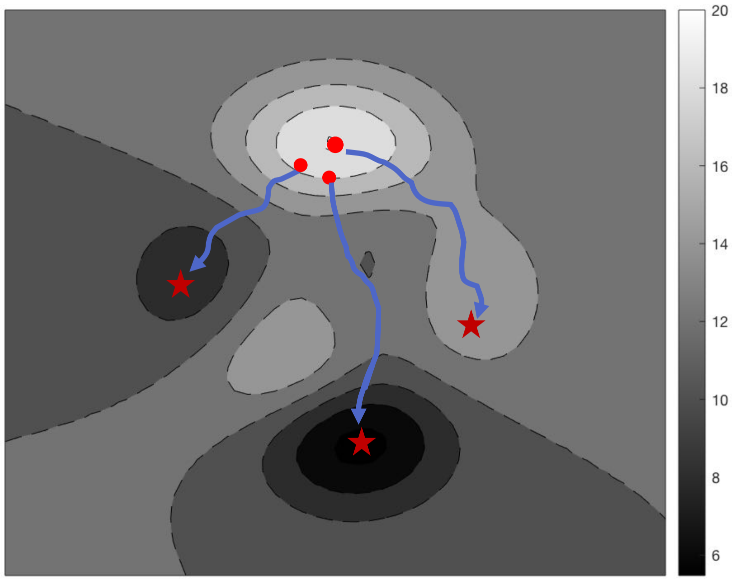
<u>Goal</u>:
Can we optimize these by meta-learning to improve DA/DG performance?

<u>Challenges</u>:
- How to define meta-losses for DA/DG?
- How to tractably train design parameters?

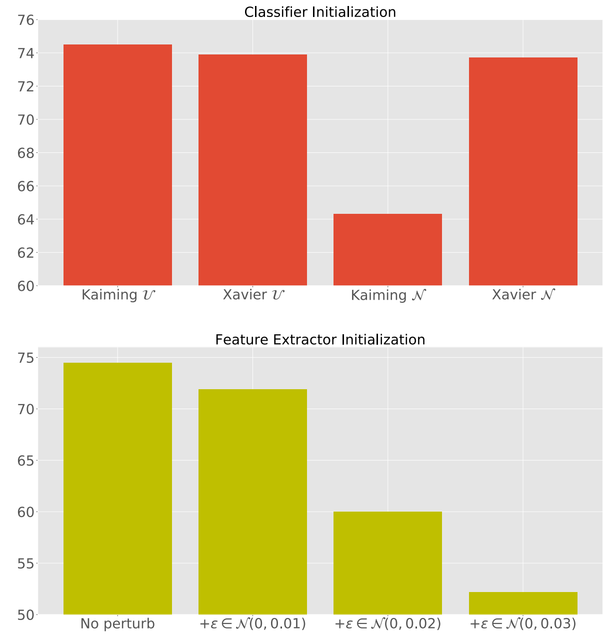
Meta-Learning for DA (ECCV-20): Concept

- Many popular DA algorithms are initialization dependent.
  - => Can we meta-learn a good <u>initialization</u>?

DA Algorithm Loss Surface



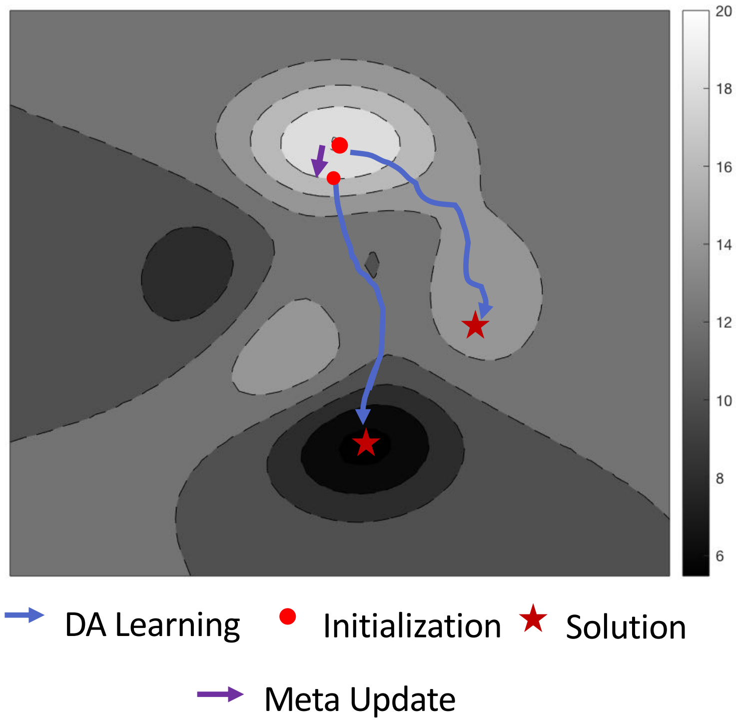
DA Learning   Initialization   Solution



MSDA [ICCV-19]
MCD [CVPR-18]
DANN [JMLR-16]
MME [ICCV-19]
CCSA [ICCV-17]
JiGen [CVPR-19]

[Li & Hospedales, Meta Learning for Domain Adaptation, ECCV-20]

Meta-DA: Implementation: Semi-Supervised DA



Initial condition to start minimizing from = hyperparameter ω

Data to use

"Find initial condition θ that leads to best target domain performance after adapt from src":

Notation: $\mathcal{L}(\theta, D)$

$$\Theta = \underbrace{\underset{\Theta}{\operatorname{argmin}} \; \mathcal{L}_{\text{outer}}(\overbrace{\mathcal{L}_{\text{inner}}(\Theta, \mathcal{D}_{\text{tr}})}^{\text{Inner-level}}, \mathcal{D}_{\text{val}})}_{\text{Outer-level}}$$

Semi-supervised domain adaptation:

Bar denotes unlabeled

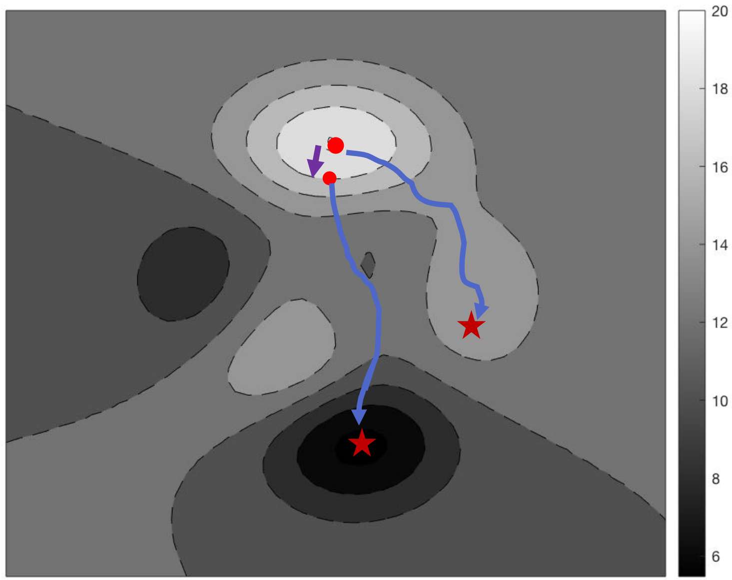
$$\Theta_0 = \underset{\Theta_0}{\operatorname{argmin}} \sum \mathcal{L}_{\text{sup}}(\underbrace{\mathcal{L}_{\text{uda}}(\mathcal{D}_{\text{S}}, \overline{\mathcal{D}}_T; \Theta_0)}, \mathcal{D}_T)$$

Unsupervised DA with unlab source data

Validate with labeled target data

[Li & Hospedales, Meta Learning for Domain Adaptation, ECCV-20]

Meta-DA: Implementation: Multi-Source Unsupervised DA



DA Learning · Initialization · Solution

Meta Update

“Find initial condition θ that leads to best target domain performance after adapt from src”:

$$\Theta = \underbrace{\arg\min_{\Theta} \; \mathcal{L}_{\mathrm{outer}}(\overbrace{\mathcal{L}_{\mathrm{inner}}(\Theta, \mathcal{D}_{\mathrm{tr}})}^{\text{Inner-level}}, \mathcal{D}_{\mathrm{val}})}_{\text{Outer-level}}$$

<u>Multi-source unsupervised domain adaptation</u>:

Bar denotes unlabeled

$$\Theta_0 = \arg\min_{\Theta_0} \sum_{\mathcal{D}_S^{\mathrm{mtr}}, \mathcal{D}_S^{\mathrm{mte}} \sim \mathcal{D}_S} \mathcal{L}_{\mathrm{sup}}(\mathcal{L}_{\mathrm{uda}}(\underbrace{\mathcal{D}_S^{\mathrm{mtr}}, \overline{\mathcal{D}}_S^{\mathrm{mte}}; \Theta_0)}_{}, \mathcal{D}_S^{\mathrm{mte}})$$

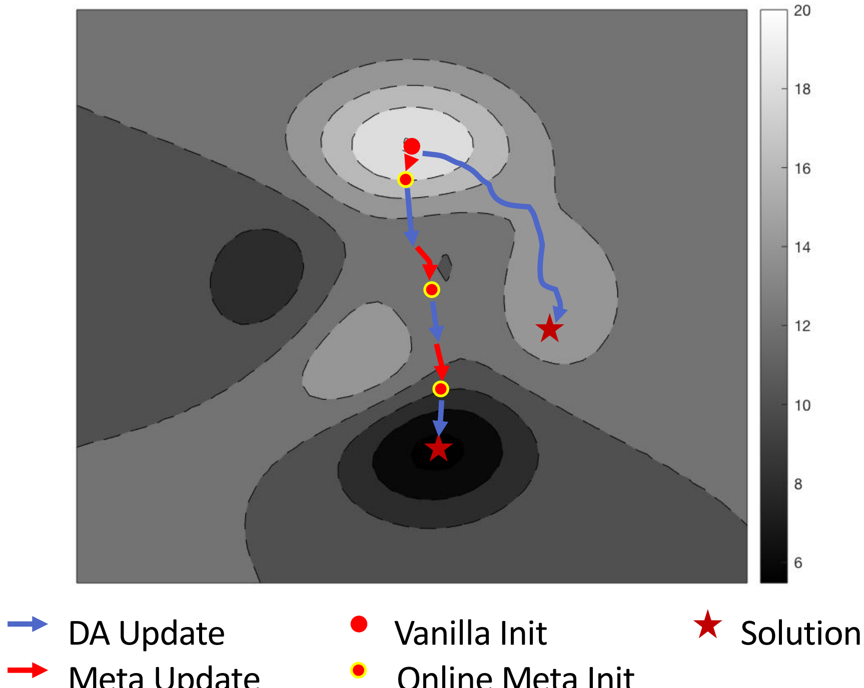
Split source domains into train+val

Unsupervised DA from train => val

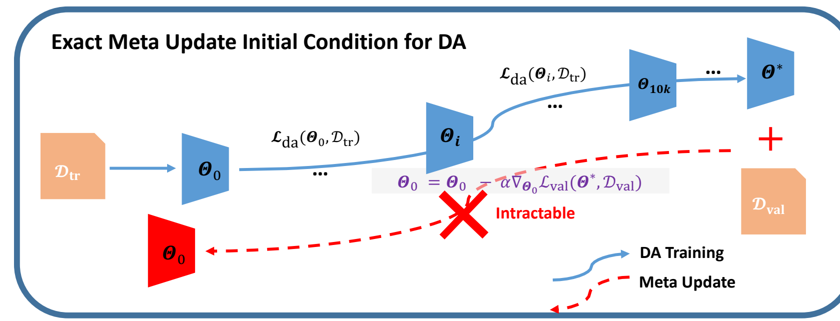
Evaluate with labels on val

Meta-DA: Optimization

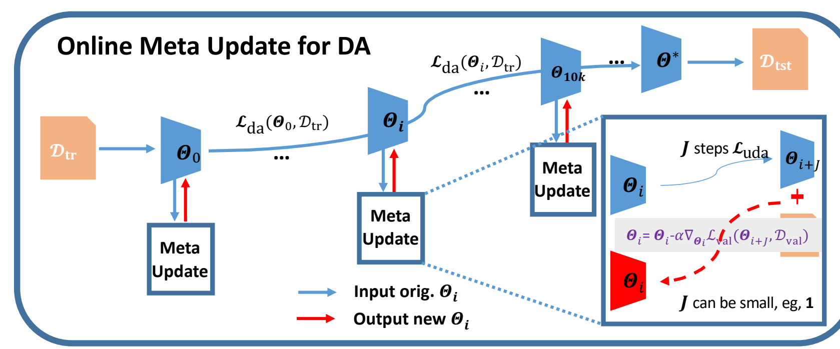
$$\Theta_0 = \underset{\Theta_0}{\operatorname{argmin}} \sum \mathcal{L}_{\text{sup}}(\mathcal{L}_{\text{uda}}(\mathcal{D}_{\text{S}}, \overline{\mathcal{D}}_T; \Theta_0), \mathcal{D}_T)$$



Vanilla optimization is intractable



Tractable: Alternate DA & Meta-Updates



Meta-DA: Result

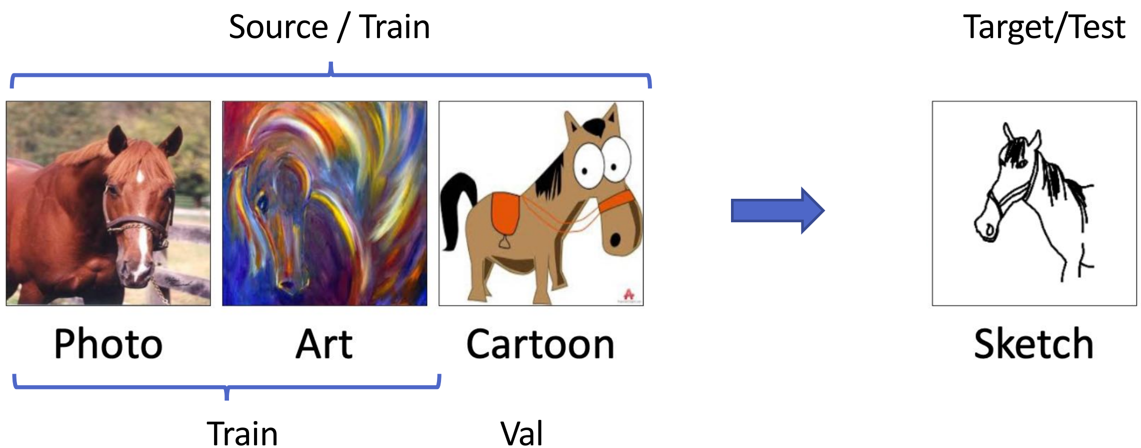
	CNN	Dataset	Base Method	MetaDA Benefit
Multi-Source	ResNet-18	PACS	MCD	+2.5%
	ResNet-18	PACS	DANN	+2.0%
	ResNet-18	PACS	JiGen	+3.4%
	ResNet-50	Office-Home	DANN	+0.7%
		Digit-Five	M^3SDA	+1.2%
Semi-Supervised	ResNet-34	DomainNet	MCD	+0.3%
	ResNet-34	Office-Home	MME	+0.7%
	ResNet-34	DomainNet	MME	+1.2%

Outline – Perspectives & Outlook

- Multi-Domain Learning & Tensor Methods
- Meta-Learning for DA and DG
  - Meta-Learning Mini-Intro
  - Meta-Learning for DA and DG
- Emerging Problem Areas and Applications

Meta-Domain Generalisation: Concept

- Recall DG problem setting



- How can we define a meta-objective?
  - In multi-source, we can optimize on validation domain performance.

Meta-DG: Cross-Domain Objective

Source

Target

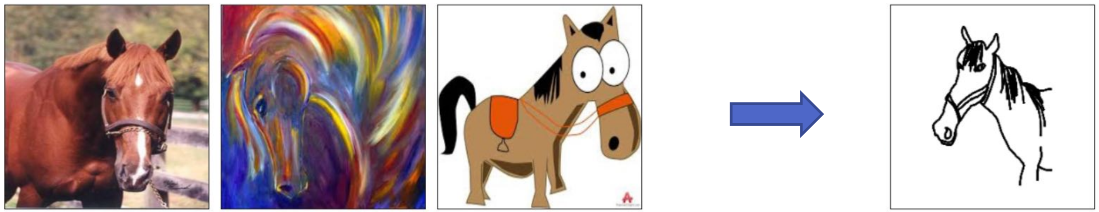


Photo Art Cartoon Sketch

Train

Val

Denote: Loss of model when trained on D_{tr}^{src} and evaluated on D_{val}^{src}

Meta-Train: $\omega^* = \arg\min_{\omega} \sum_{(D_{val}^{src}, D_{tr}^{src}) \sim D} \mathcal{L}^+(D_{val}^{src} | D_{tr}^{src}; \omega)$

Meta-Train: Bilevel Optimization

$$\omega^* = \arg\min_{\omega} \mathcal{L}^+(D_{val}^{src}; \theta^*, \omega)$$

$$\text{s.t. } \theta^* = \arg\min_{\theta} \mathcal{L}(D_{tr}^{src}; \theta, \omega)$$

Meta-Representation

Meta-Test: $\theta^* = \arg\min_{\theta} \mathcal{L}(D^{src}; \theta, \omega^*)$

Test: $y = f_{\theta^*}(x)$

Meta-DG: MetaReg (NeurIPS-18) & Feature Critic (ICML-19)

- What hyper-parameter to (meta)-learn?
- MetaReg (NeurIPS-18):
  - Assume regularizer: $\Omega_{\omega}(\theta) = \sum_k \omega_k |\theta_k|$
- Feature-Critic (ICML-19):
  - Introduce a feature quality "critic" as additional loss:
    - Assume classifier is composed as: $y = f_\theta(g_\theta(x))$

$$\Omega_{\omega}(\theta, x) = h_{\omega}(g_\theta(x))$$

"Critic" Neural Network
EG: Do the features look separable?

- Learning: Optimize base objective:

$$\mathcal{L}^{\sup}(D^{src}) + \Omega_{\omega}(\theta, x)$$

[Balaji, MetaReg: Towards Domain Generalization using Meta-Regularization, NIPS-18]
[Li, Feature Critic Networks for Heterogeneous Domain Generalization, ICML-19]

Meta-DG: Meta-Optimization

- How to meta-optimize?
- Concept:

Meta-Train: Bilevel Optimization

$$\omega^* = \arg\min_{\omega} \mathcal{L}^+(D_{val}^{src}; \theta^*, \omega)$$

$$\text{s.t. } \theta^* = \arg\min_{\theta} \mathcal{L}(D_{tr}^{src}; \theta) + \Omega_{\omega}(D_{tr}^{src}; \theta)$$

$$\Omega_\omega(\theta) = \sum_k \omega_k |\theta_k| \qquad \Omega_\omega(\theta, x) = h_\omega(g_\theta(x))$$

- In practice, alternate:

Conventional model update

$$\theta^- \leftarrow \theta - \alpha \nabla_\theta \mathcal{L}(D_{tr}^{src}, \theta)$$

Update model using ω

$$\theta^+ \leftarrow \theta - \alpha \nabla_\theta \big(\mathcal{L}(D_{tr}^{src}, \theta) + \Omega_\omega(D_{tr}^{src}, \theta)\big)$$

$$\omega \leftarrow \omega - \eta \nabla_\omega \tanh\big(\mathcal{L}^+(D_{val}^{src}; \theta^+) - \mathcal{L}^+(D_{val}^{src}; \theta^-)\big)$$

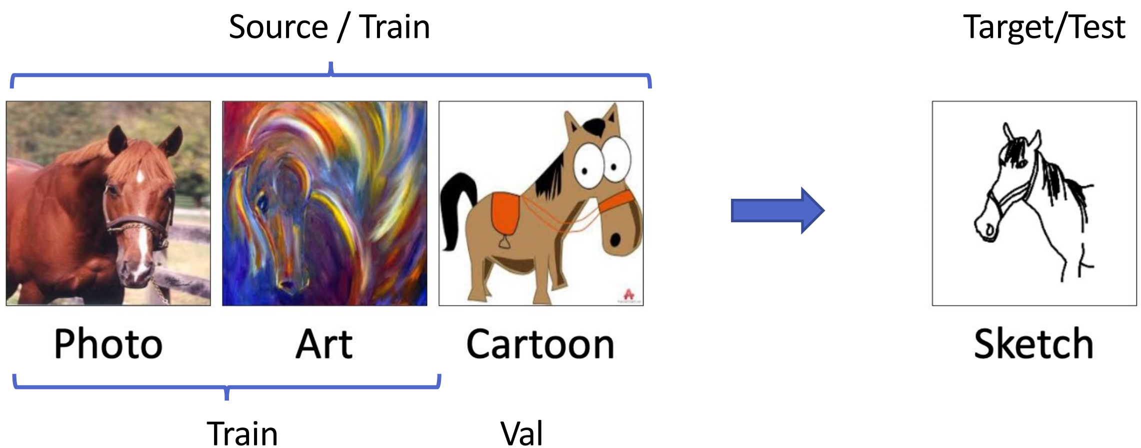
Optimize ω so that cross-domain performance Is better than without it.

- => Generally expensive & unstable

[Balaji, MetaReg: Towards Domain Generalization using Meta-Regularization, NIPS-18]
[Li, Feature Critic Networks for Heterogeneous Domain Generalization, ICML-19]

Meta-Domain Generalisation: Summary

- DG problem setting



- Optimize regularizer/lss for validation domain performance.

$$\Omega_\omega(\theta) = \sum_k \omega_k |\theta_k| \qquad \Omega_\omega(\theta, x) = h_\omega(g_\theta(x))$$

[Balaji, MetaReg: Towards Domain Generalization using Meta-Regularization, NIPS-18]
[Li, Feature Critic Networks for Heterogeneous Domain Generalization, ICML-19]

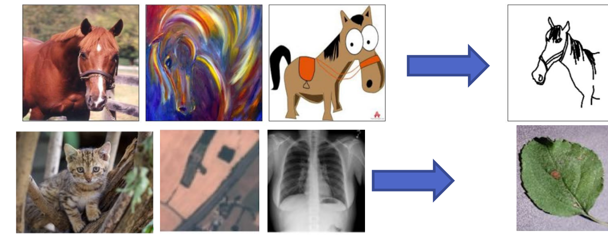
Meta-Learning: Outlook/Open Questions

- Meta-DA/DG can complement conventional DA/DG research.
- Very early days for meta-learning in DA/DG.
  - Which components are practically important to learn?
  - Which base DA/DG algorithms perform well when meta-optimized?
  - Defining meta-objectives and data-flow for DA/DG/etc.
  - Draw on basic research progress in meta-learning.

Outline – Perspectives & Outlook

- Multi-Domain Learning & Tensor Methods
- Meta-Learning for DA and DG
- Emerging Problem Areas and Applications

DG: Heterogeneous Case



Homogeneous DG:

- Shared source & target Label space

Heterogeneous DG:

➢ Disjoint label space in source + target → Feature generalisation.

➢ CF: "*ImageNet trained CNN as feature extractor*"

Source domains: ImageNet CNN | Hetero DG trained CNN

Fix the Feature Extractor

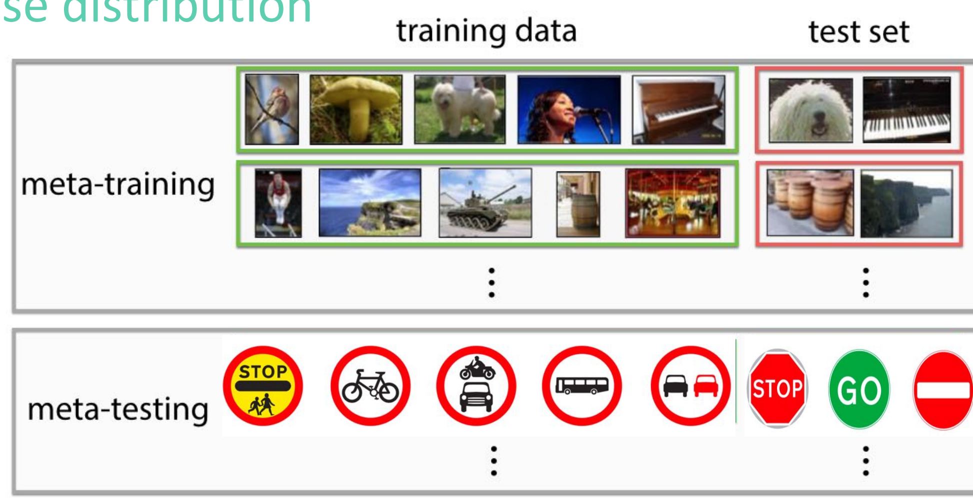
Train split of **target domains**:
- Extract features
- Train a SVM/KNN classifier

Test split of **target domains**: Evaluate performance

[Li, Feature Critic Networks for Heterogeneous Domain Generalization, ICML-19; Li, Episodic Training for DG, ICCV-19]

Cross-Domain Few-Shot Learning

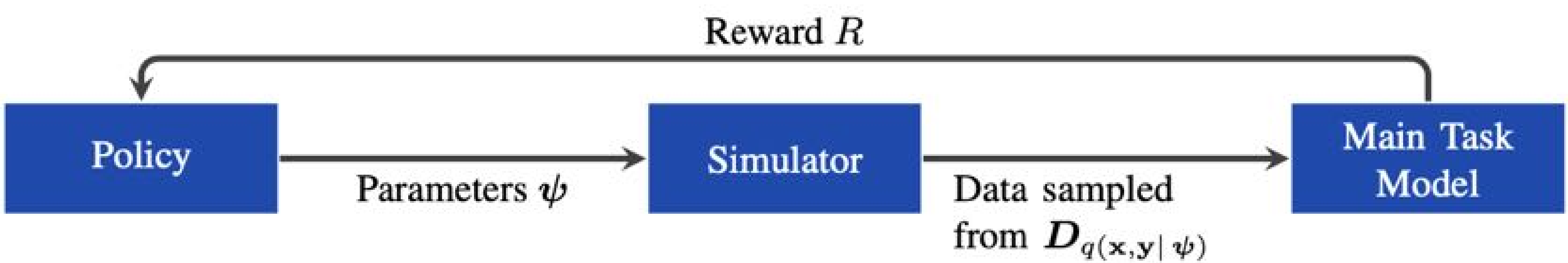
- Traditional FSL:
  - Transfer knowledge from meta-train to meta-test?
- CD-FSL:
  - How to transfer across domain-shift? (See: CD-FSL, Meta-Dataset)
- "Learned Feature-Wise Transforms" (ICLR-20)
  - Observation: Stochastic layers can improve cross-domain generalization.
- => Apply meta-learning to train noise distribution



[Tseng, Cross-Domain Few-Shot Classification via Learned Feature-Wise Transformation, ICLR-20]
[Triantafillou, Meta-Dataset: A Dataset of Datasets for Learning to Learn from Few Examples, ICLR-20]

Sim-2-Real Meta-Learning

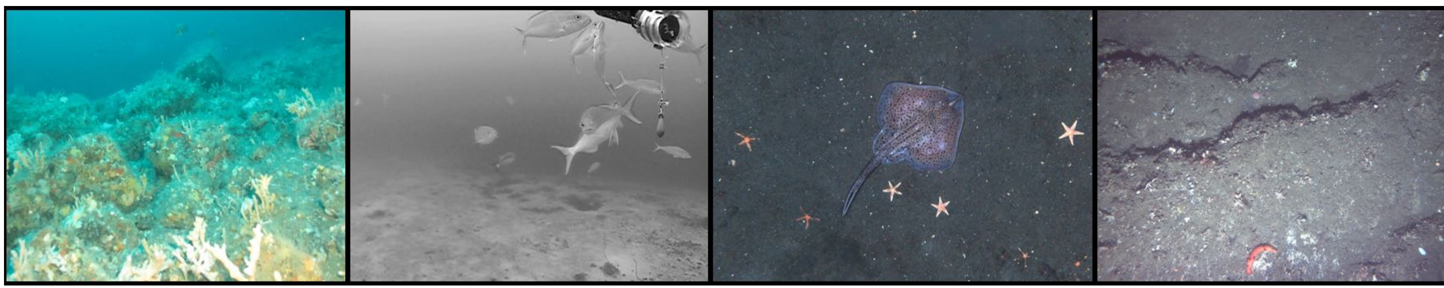
- Goal: Sim2Real for object detection & segmentation
- Meta-Representation: Simulator/Dataset.
  - Idea: Train simulator, so that it makes data, which when used to train a model, leads to high accuracy on real-world validation set.
  - Note: (Requires RL/ES. Simulator non-differentiable).



[ICLR'19, Learning to Simulate ; ICCV'19, Meta-Sim: Learning to Generate Synthetic Datasets]

Real-World Consumers of DA/DG methods

- Underwater imaging
  - Check out, e.g., CVPR Workshop + Challenge: “Automated Analysis of Marine Video for Environmental Monitoring”



Real-World Consumers of DA/DG methods

- Geographic Diversity.

Ground truth: Soap — **Nepal, 288 $/month**

Azure: food, cheese, bread, cake, sandwich
Clarifai: food, wood, cooking, delicious, healthy
Google: food, dish, cuisine, comfort food, spam
Amazon: food, confectionary, sweets, burger
Watson: food, food product, turmeric, seasoning
Tencent: food, dish, matter, fast food, nutriment

Ground truth: Soap — **UK, 1890 $/month**

Azure: toilet, design, art, sink
Clarifai: people, faucet, healthcare, lavatory, wash closet
Google: product, liquid, water, fluid, bathroom accessory
Amazon: sink, indoors, bottle, sink faucet
Watson: gas tank, storage tank, toiletry, dispenser, soap dispenser
Tencent: lotion, toiletry, soap dispenser, dispenser, after shave

Ground truth: Spices — **Phillipines, 262 $/month**

Azure: bottle, beer, counter, drink, open
Clarifai: container, food, bottle, drink, stock
Google: product, yellow, drink, bottle, plastic bottle
Amazon: beverage, beer, alcohol, drink, bottle
Watson: food, larder food supply, pantry, condiment, food seasoning
Tencent: condiment, sauce, flavorer, catsup, hot sauce

Ground truth: Spices — **USA, 4559 $/month**

Azure: bottle, wall, counter, food
Clarifai: container, food, can, medicine, stock
Google: seasoning, seasoned salt, ingredient, spice, spice rack
Amazon: shelf, tin, pantry, furniture, aluminium
Watson: tin, food, pantry, paint, can
Tencent: spice rack, chili sauce, condiment, canned food, rack

[de Vries, Does Object Recognition Work for Everyone?, ICCV WS-2019]

Visual Question Answering

Can you park here? | no no no | no no yes

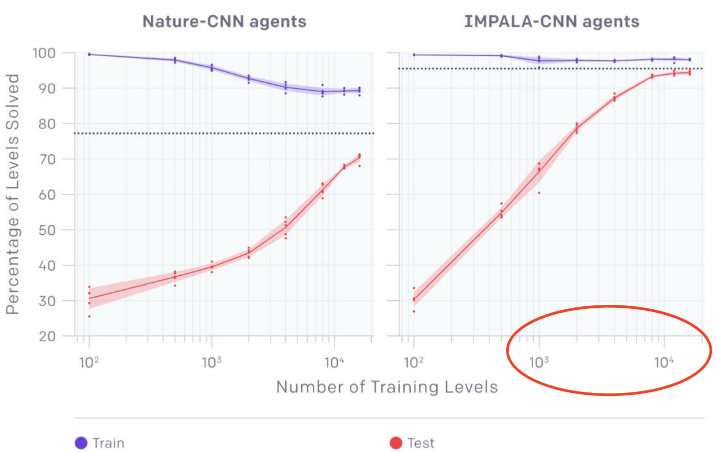
- VQA – 75% accuracy?
  - What about domain-shift?
- When our question asking-RL agent makes up new questions....

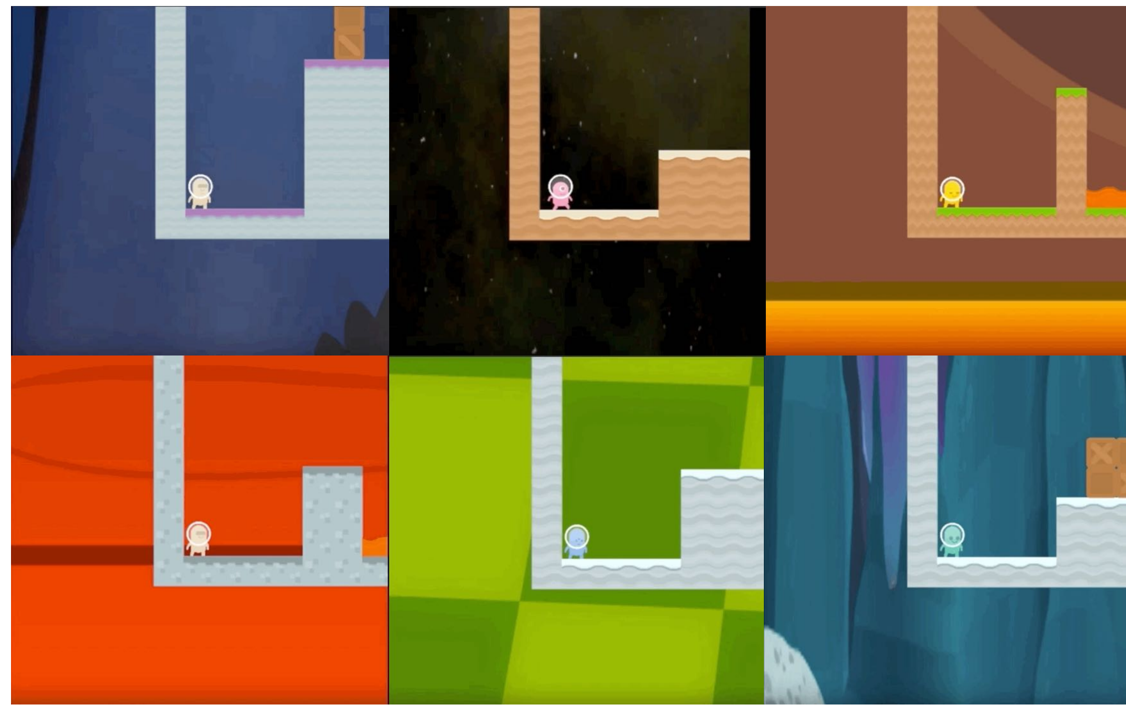
Q: Is the landing gear down?
A: Yes.

[Liu, Inverse Visual Question Answering: A New Benchmark and VQA Diagnosis Tool, T-PAMI-18]

Domain-Shift in RL

- RL agents often bottlenecked by visual domain-shift





[Cobbe, Quantifying Generalization in Reinforcement Learning, ICML-19]

Applications & Open Questions

- Heterogeneous DG
- Cross-Domain Few-Shot Learning
- Sim-2-Real
- Real-World Consumers: Underwater & Geographic Diversity
- VQA
- Domain-Shift in RL

Thanks for Listening!

References

Multi-Domain Learning

Rebuffi, Learning multiple visual domains with residual adapters, NIPS-17

Rebuffi, Efficient parametrization of multi-domain deep neural networks, CVPR-18

Yang & Hospedales, A Unified Perspective on Multi-Domain and Multi-Task Learning, ICLR-17

Yang & Hospedales, Deep Multi-task Representation Learning: A Tensor Factorisation Approach, ICLR-17

Bulat, Incremental multi-domain learning with network latent tensor factorization, AAAI-20

Bulat, Factorized Higher-Order CNNs With an Application to Spatio-Temporal Emotion Estimation, CVPR-20

Deecke, Latent Domain Learning with Dynamic Residual Adapters, arXiv:2006.00996

Bragman, Stochastic filter groups for multi-task cnns: Learning specialist and generalist convolution kernels, ICCV-19

Kanakis, Reparameterizing Convolutions for Incremental Multi-Task Learning without Task Interference, ECCV-20

Meta Learning

Finn, Model-Agnostic Meta-learning for Fast Adaptation of Deep Networks, ICML-17

Hospedales et al, Meta-Learning in Neural Networks: A Survey, arXiv:2004.05439, 2020

Li, Meta Learning for Domain Adaptation, ECCV-20

Balaji, MetaReg: Towards Domain Generalization using Meta-Regularization, NIPS-18

Li, Feature Critic Networks for Heterogeneous Domain Generalization, ICML-19

Applications

Li, Feature Critic Networks for Heterogeneous Domain Generalization, ICML-19;

Li, Episodic Training for DG, ICCV-19

Tseng, Cross-Domain Few-Shot Classification via Learned Feature-Wise Transformation, ICLR-20

Triantafillou, Meta-Dataset: A Dataset of Datasets for Learning to Learn from Few Examples, ICLR-20

de Vries, Does Object Recognition Work for Everyone?, ICCV WS-2019

Liu, Inverse Visual Question Answering: A New Benchmark and VQA Diagnosis Tool, T-PAMI-18

Cobbe, Quantifying Generalization in Reinforcement Learning, ICML-19