Riemannian Walk for Incremental Learning: Understanding Forgetting and Intransigence

(ECCV18)

Puneet K. Dokania
(University of Oxford)

7th Aug, 2018

Jointly with

Arslan Chaudhry, Thalaiyasingam Ajanthan,
Philip H. S. Torr

Why Incremental Learning?

Why Incremental Learning?

- Standard Learning (Classification): Given $\mathcal{D} = \{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^n$ learn mapping $f_\theta : \mathbf{x} \rightarrow \mathbf{y}$

$$\min_\theta \mathbb{E}_{\mathbf{x},\mathbf{y}\sim\mathcal{D}} L(f_\theta(\mathbf{x}), \mathbf{y})$$

Why Incremental Learning?

- Standard Learning (Classification): Given $\mathcal{D} = \{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^n$ learn mapping $f_\theta : \mathbf{x} \rightarrow \mathbf{y}$

$$\min_\theta \mathbb{E}_{\mathbf{x},\mathbf{y}\sim\mathcal{D}} L(f_\theta(\mathbf{x}), \mathbf{y})$$

- Given new dataset (may contain samples for previous labels or new ones)

$$\min_\theta \mathbb{E}_{\mathbf{x},\mathbf{y}\sim\mathcal{D}\cup\bar{\mathcal{D}}} L(f_\theta(\mathbf{x}), \mathbf{y})$$

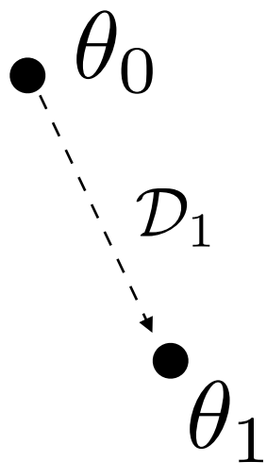
  - Not scalable
  - Privacy
  - Redundancy

Incremental Learning **Training** Objectives

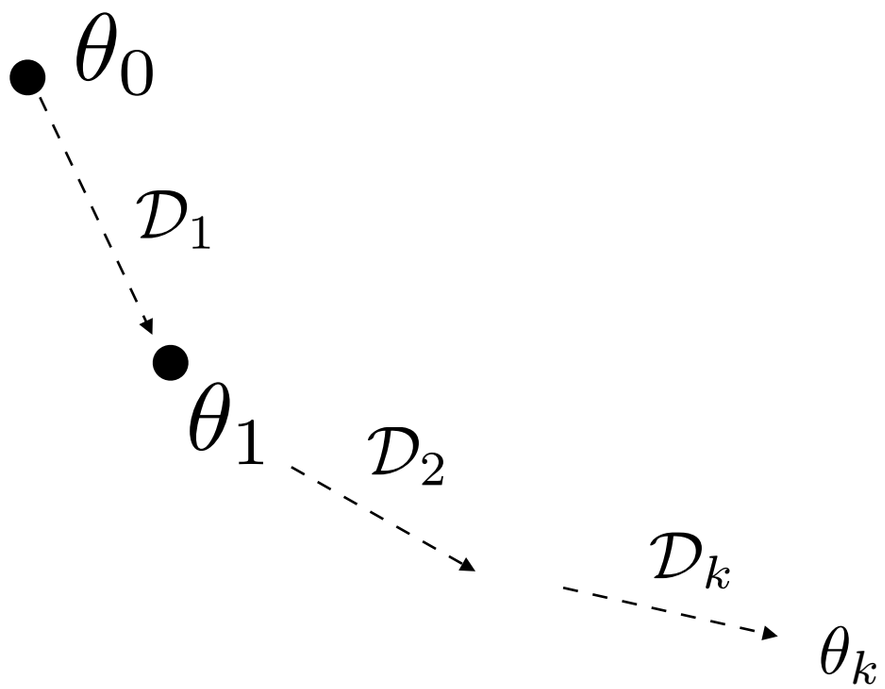
Incremental Learning **Training** Objectives

- θ_0

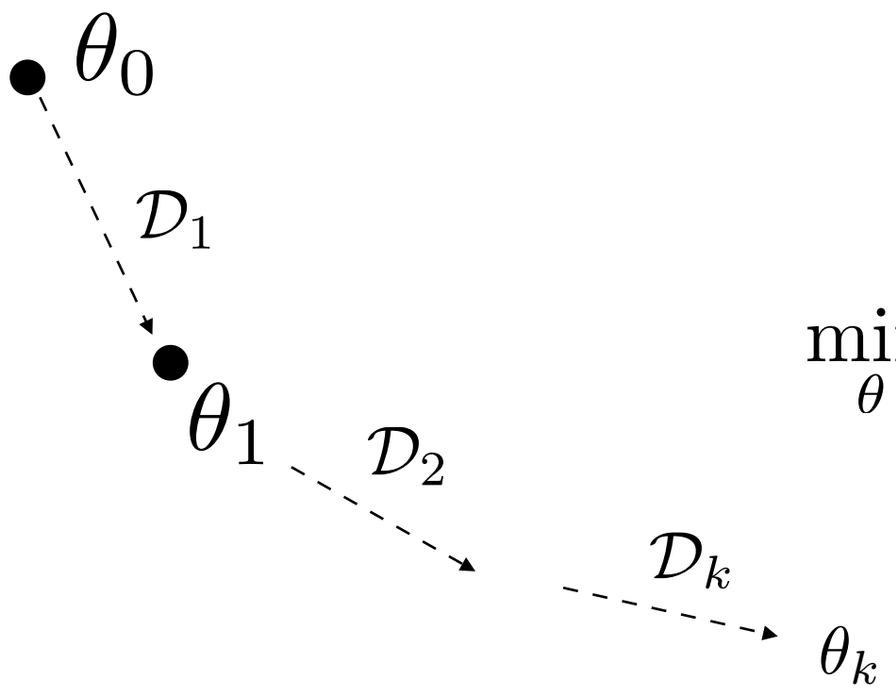
Incremental Learning **Training** Objectives



Incremental Learning **Training** Objectives



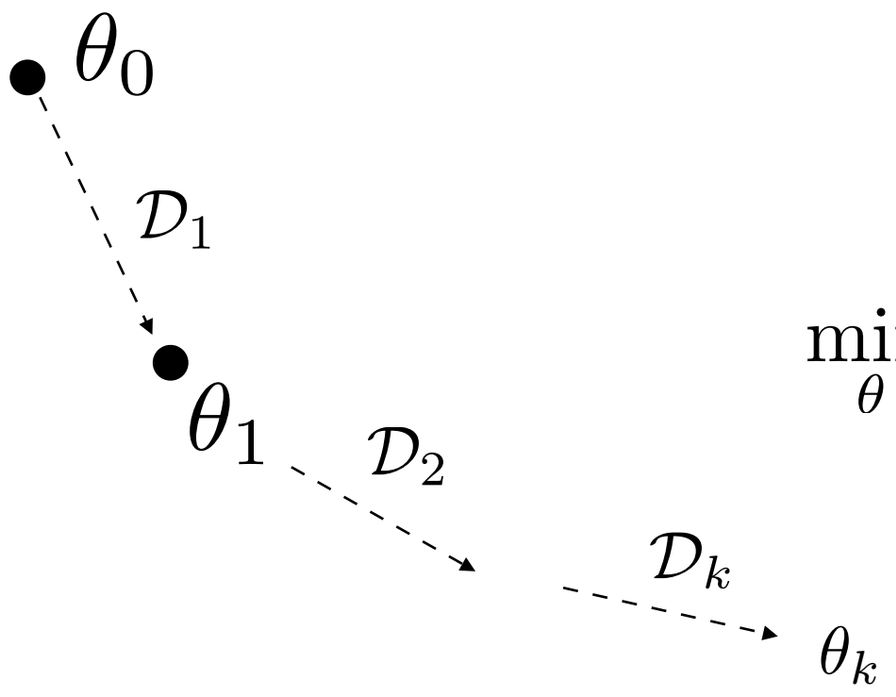
Incremental Learning **Training** Objectives



$$\min_{\theta} \mathbb{E}_{\cup_{i=1}^{k} \mathcal{D}_i} L(f_\theta(\mathbf{x}), \mathbf{y}) \equiv \min_{\theta} \mathbb{E}_{\mathcal{D}_k} L(f_\theta(\mathbf{x}), \mathbf{y}; \mathbb{K})$$

(Previous knowledge)

Incremental Learning **Training** Objectives

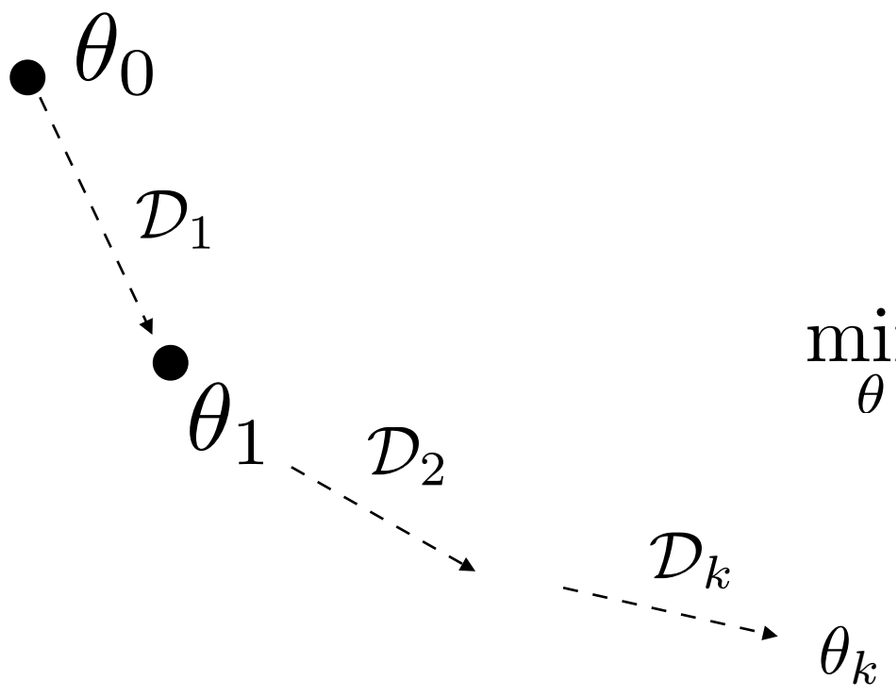


$$\min_{\theta} \mathbb{E}_{\cup_{i=1}^{k} \mathcal{D}_i} L(f_\theta(\mathbf{x}), \mathbf{y}) \equiv \min_{\theta} \mathbb{E}_{\mathcal{D}_k} L(f_\theta(\mathbf{x}), \mathbf{y}; \mathbb{K})$$

(Previous knowledge)

- Define Knowledge
  - Input-Output Behaviour (knowledge distillation type)
  - Parameters

Incremental Learning **Training** Objectives

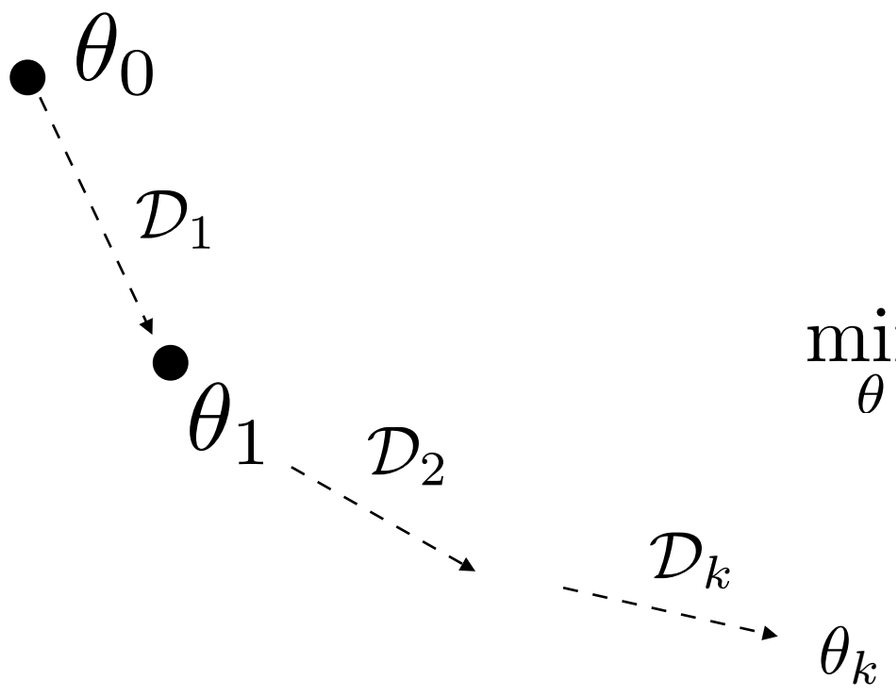


$$\min_{\theta} \mathbb{E}_{\cup_{i=1}^{k} \mathcal{D}_i} L(f_\theta(\mathbf{x}), \mathbf{y}) \equiv \min_{\theta} \mathbb{E}_{\mathcal{D}_k} L(f_\theta(\mathbf{x}), \mathbf{y}; \mathbb{K})$$

(Previous knowledge)

- Define Knowledge
  - Input-Output Behaviour (knowledge distillation type)
  - Parameters
- Preserve Knowledge? Avoid Forgetting

Incremental Learning **Training** Objectives



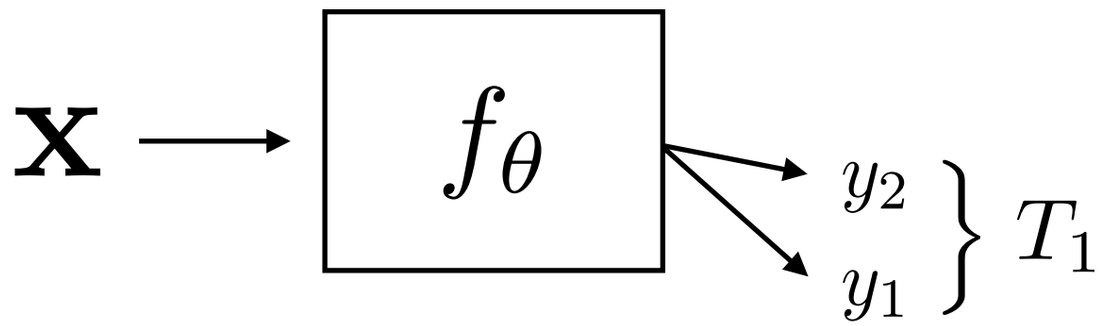
$$\min_{\theta} \mathbb{E}_{\cup_{i=1}^{k} \mathcal{D}_i} L(f_\theta(\mathbf{x}), \mathbf{y}) \equiv \min_{\theta} \mathbb{E}_{\mathcal{D}_k} L(f_\theta(\mathbf{x}), \mathbf{y}; \mathbb{K})$$

(Previous knowledge)

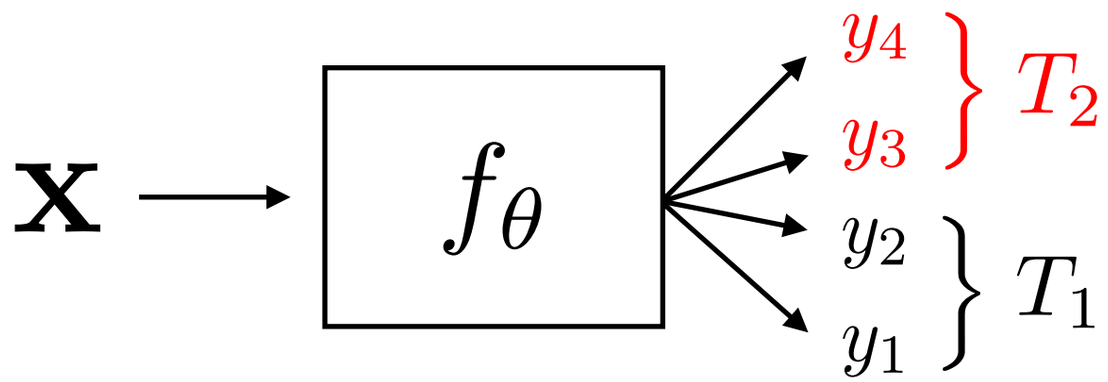
- Define Knowledge
  - Input-Output Behaviour (knowledge distillation type)
  - Parameters
- Preserve Knowledge? Avoid Forgetting
- Update Knowledge? Avoid Intransigence (inability to learn new tasks)

Incremental Learning **Evaluation** Set-up (Multi/Single-head)

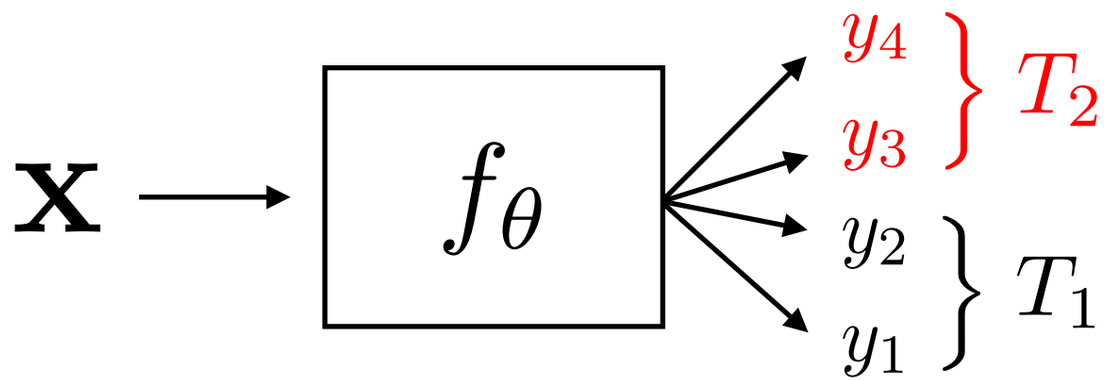
Incremental Learning **Evaluation** Set-up (Multi/Single-head)



Incremental Learning **Evaluation** Set-up (Multi/Single-head)

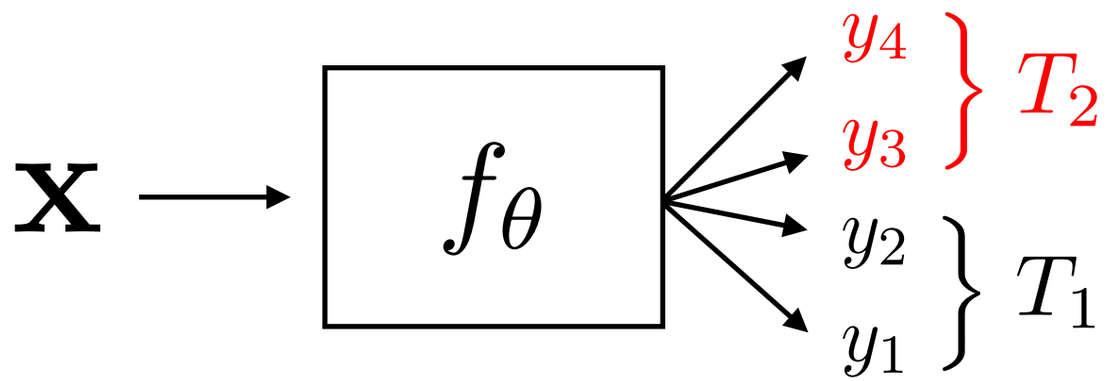


Incremental Learning **Evaluation** Set-up (Multi/Single-head)



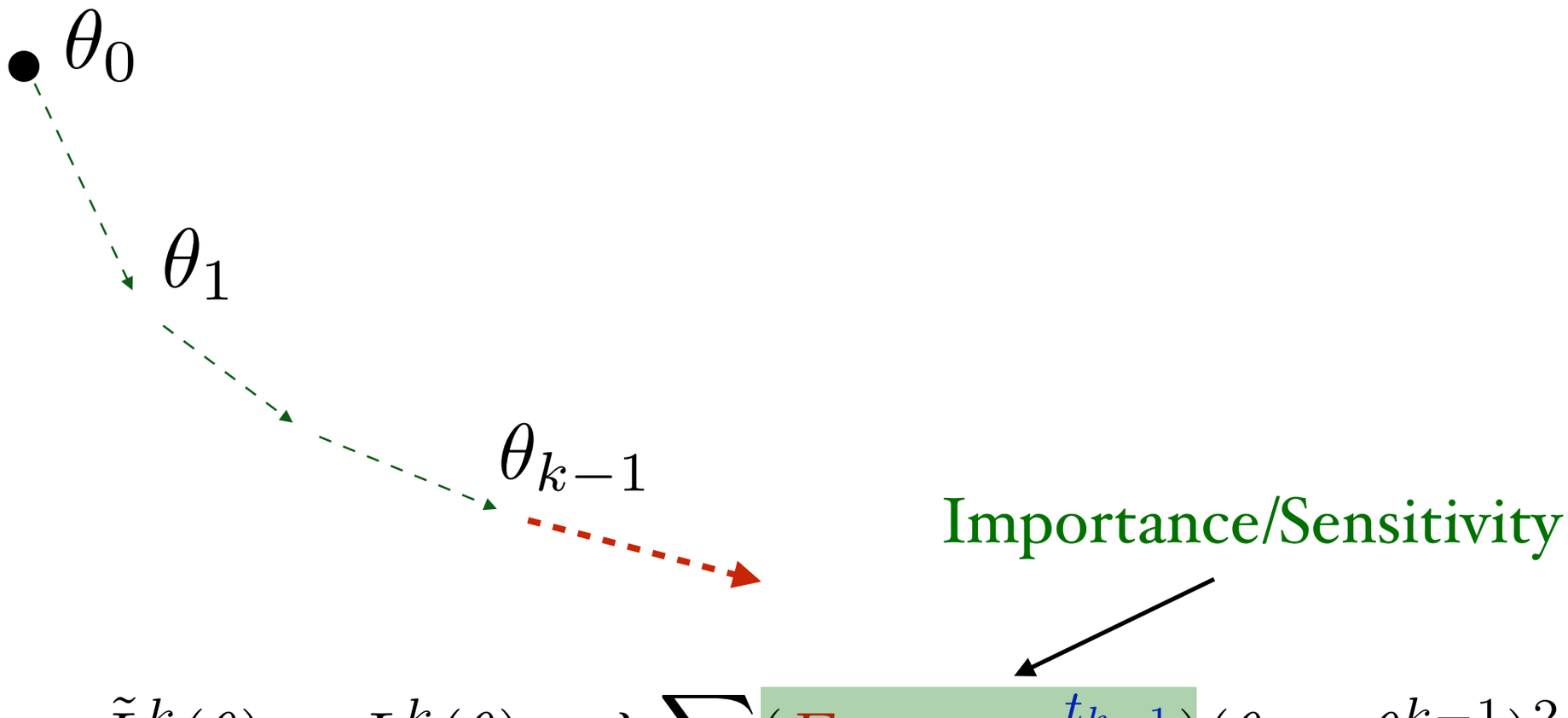
- **Multi-head**
  - task id is known (for *k-th* task, the test label set is $\mathcal{Y}_k$)
  - violates the objective behind incremental learning
  - if *k-th* task contains only one label, this evaluation implies knowing the ground-truth label itself at the test time

Incremental Learning **Evaluation** Set-up (Multi/Single-head)



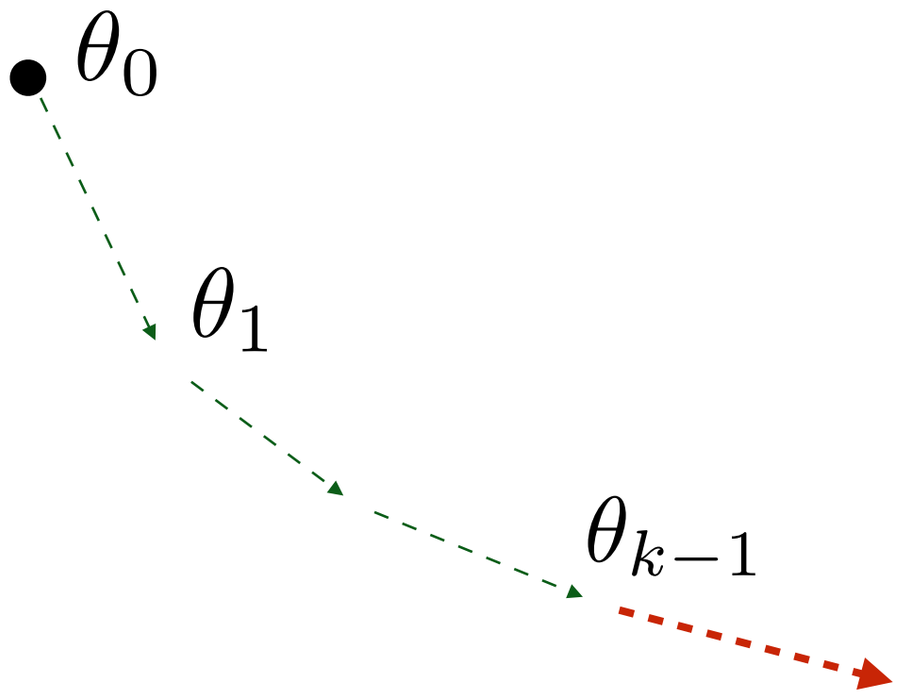
- **Multi-head**
  - task id is known (for *k-th* task, the test label set is $\mathcal{Y}_k$)
  - violates the objective behind incremental learning
  - if *k-th* task contains only one label, this evaluation implies knowing the ground-truth label itself at the test time
- **Single-head**
  - task id is unknown (test set contains all the labels seen $\cup_{i=1}^{k} \mathcal{Y}_i$)
  - much more challenging

RWalk Objective Function and Intuitions



$$\tilde{L}^k(\theta) = L^k(\theta) + \lambda \sum_i (F_{\theta_i^{k-1}} + s_{t_0}^{t_{k-1}})(\theta_i - \theta_i^{k-1})^2$$

RWalk Objective Function and Intuitions

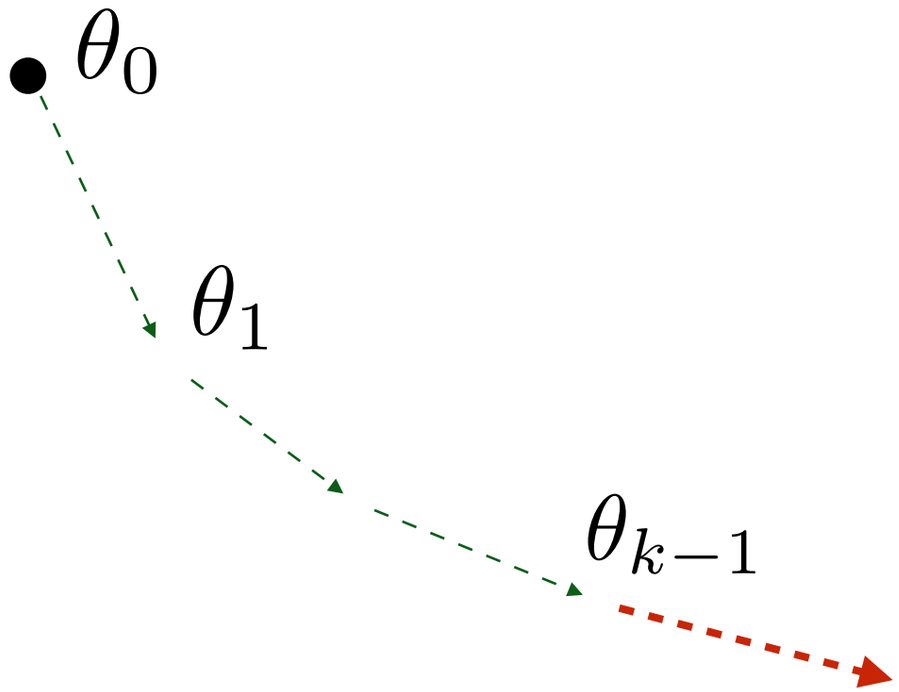


Importance/Sensitivity

$$\tilde{L}^k(\theta) = L^k(\theta) + \lambda \sum_i (F_{\theta_i^{k-1}} + s_{t_0}^{t_{k-1}})(\theta_i - \theta_i^{k-1})^2$$

Fisher importance

RWalk Objective Function and Intuitions



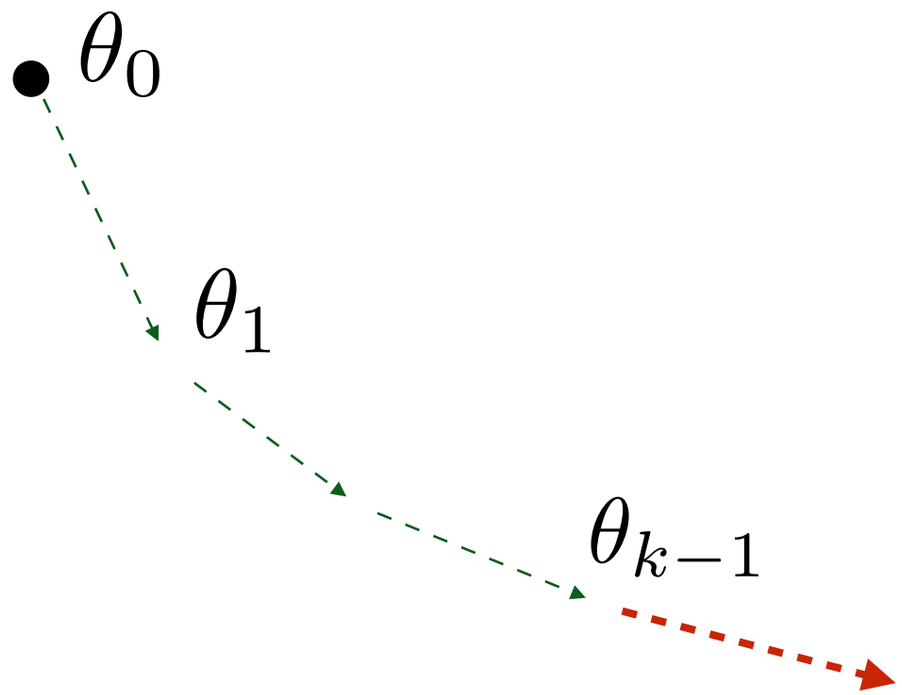
Importance/Sensitivity

$$\tilde{L}^k(\theta) = L^k(\theta) + \lambda \sum_i (F_{\theta_i^{k-1}} + s_{t_0}^{t_{k-1}})(\theta_i - \theta_i^{k-1})^2$$

Fisher importance

Optimization path based importance

RWalk Objective Function and Intuitions



Importance/Sensitivity

$$\tilde{L}^k(\theta) = L^k(\theta) + \lambda \sum_i (F_{\theta_i^{k-1}} + s_{t_0}^{t_{k-1}})(\theta_i - \theta_i^{k-1})^2$$

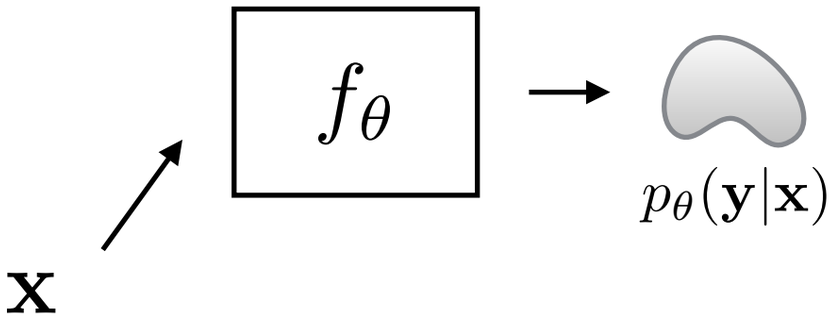
Fisher importance

Optimization path based importance

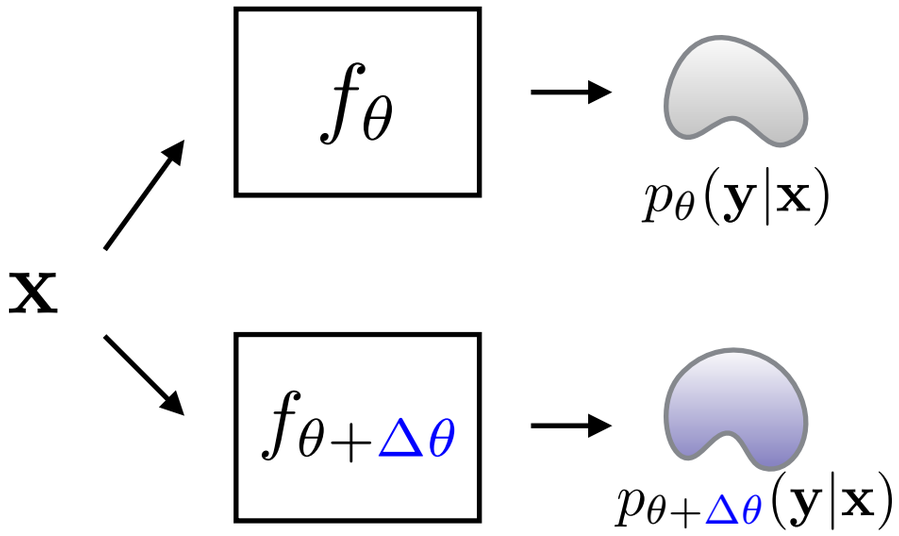
Let us now look into these parameter importances individually

Fisher based parameter importance (1/2)
(approximate KL and Fisher)

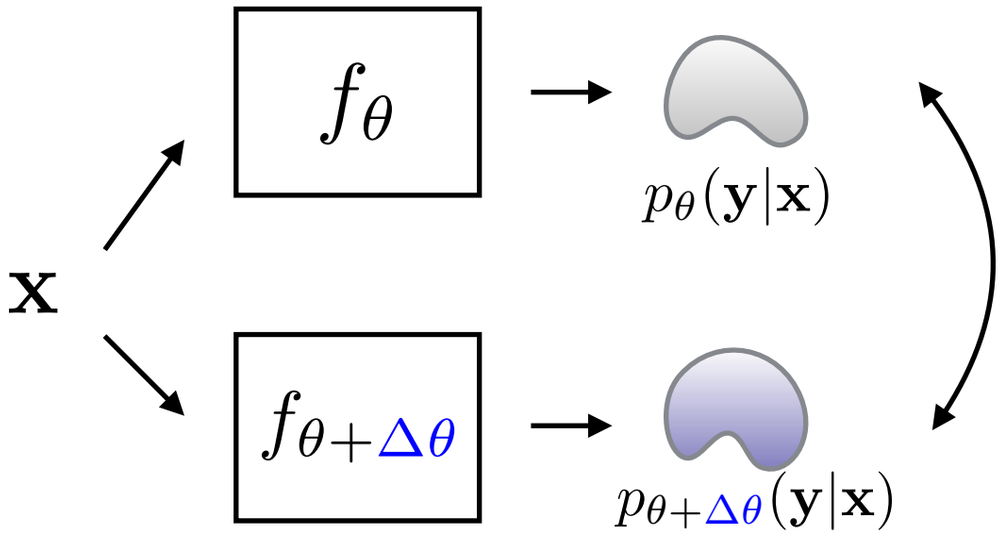
Fisher based parameter importance (1/2)
(approximate KL and Fisher)



Fisher based parameter importance (1/2)
(approximate KL and Fisher)



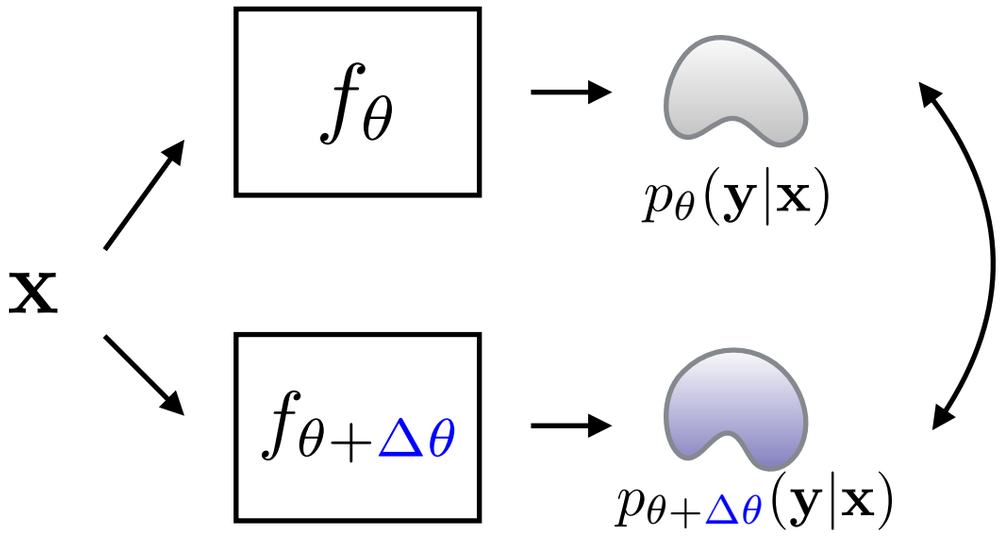
Fisher based parameter importance (1/2)
(approximate KL and Fisher)



Fisher Information Matrix

$$D_{KL}(p_\theta \| p_{\theta+\Delta\theta}) \approx \frac{1}{2}\Delta\theta^\top F_\theta \Delta\theta$$

Fisher based parameter importance (1/2)
(approximate KL and Fisher)

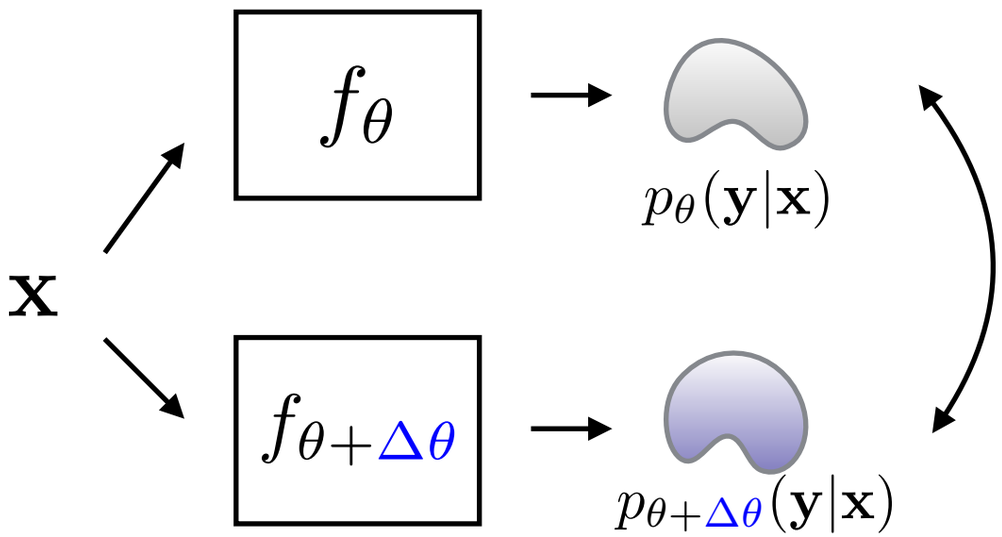


Fisher Information Matrix

$$D_{KL}(p_\theta \| p_{\theta+\Delta\theta}) \approx \frac{1}{2} \Delta\theta^\top F_\theta \Delta\theta$$

$$F_\theta = \mathbb{E}_{\mathbf{x}\sim\mathcal{D}, \mathbf{y}\sim p_\theta(\mathbf{y}|\mathbf{x})} \left[\left(\frac{\partial \log p_\theta(\mathbf{y}|\mathbf{x})}{\partial \theta} \right) \left(\frac{\partial \log p_\theta(\mathbf{y}|\mathbf{x})}{\partial \theta} \right)^\top \right].$$

Fisher based parameter importance (1/2)
(approximate KL and Fisher)



Fisher Information Matrix

$$D_{KL}(p_\theta \| p_{\theta+\Delta\theta}) \approx \frac{1}{2}\Delta\theta^\top F_\theta \Delta\theta$$

$$F_\theta = \mathbb{E}_{\mathbf{x}\sim\mathcal{D},\mathbf{y}\sim p_\theta(\mathbf{y}|\mathbf{x})}\left[\left(\frac{\partial \log p_\theta(\mathbf{y}|\mathbf{x})}{\partial\theta}\right)\left(\frac{\partial \log p_\theta(\mathbf{y}|\mathbf{x})}{\partial\theta}\right)^\top\right].$$

- Cross Entropy Loss = $-\log p_\theta(\mathbf{y}|\mathbf{x})$
- By construction PSD (outer-product)
- Computationally expensive
- Huge matrix

Fisher based parameter importance (2/2)

Fisher based parameter importance (2/2)

$$F_\theta = \mathbb{E}_{\substack{\mathbf{y}\sim p_\theta(\mathbf{y}|\mathbf{x}) \\ \mathbf{x}\sim\mathcal{D}}} \mathbf{g}\mathbf{g}^\top$$

multi backward passes

log-likelihood (CE) gradient

Fisher based parameter importance (2/2)

$$F_\theta = \mathbb{E}_{\substack{\mathbf{y}\sim p_\theta(\mathbf{y}|\mathbf{x}) \\ \mathbf{x}\sim\mathcal{D}}} \mathbf{g}\mathbf{g}^\top \approx \mathbb{E}_{(\mathbf{x},\mathbf{y})\sim\mathcal{D}} \; \mathbf{g}\mathbf{g}^\top$$

multi backward passes

log-likelihood (CE) gradient

empirical Fisher (one fwd-bwd)

Fisher based parameter importance (2/2)

$$F_\theta = \mathbb{E}_{\substack{\mathbf{y}\sim p_\theta(\mathbf{y}|\mathbf{x}) \\ \mathbf{x}\sim\mathcal{D}}} \mathbf{g}\mathbf{g}^\top \approx \mathbb{E}_{(\mathbf{x},\mathbf{y})\sim\mathcal{D}} \; \mathbf{g}\mathbf{g}^\top$$

multi backward passes

log-likelihood (CE) gradient

empirical Fisher (one fwd-bwd)

$$D_{KL}(p_\theta \| p_{\theta+\Delta\theta}) \approx \frac{1}{2}\Delta\theta^\top F_\theta \Delta\theta$$

2nd order approx

Fisher based parameter importance (2/2)

$$F_\theta = \mathbb{E}_{\substack{\mathbf{y}\sim p_\theta(\mathbf{y}|\mathbf{x}) \\ \mathbf{x}\sim\mathcal{D}}} \mathbf{g}\mathbf{g}^\top \approx \mathbb{E}_{(\mathbf{x},\mathbf{y})\sim\mathcal{D}} \; \mathbf{g}\mathbf{g}^\top$$

multi backward passes

log-likelihood (CE) gradient

empirical Fisher (one fwd-bwd)

$$D_{KL}(p_\theta \| p_{\theta+\Delta\theta}) \approx \frac{1}{2}\Delta\theta^\top F_\theta \Delta\theta \approx \frac{1}{2}\sum_i (\Delta\theta_i)^2 \underbrace{\mathbb{E}_\mathcal{D}\left(g_i^2\right)}_{F_{\theta_i}}$$

2nd order approx

- empirical
- independence

Fisher based parameter importance (2/2)

$$F_\theta = \mathbb{E}_{\substack{\mathbf{y}\sim p_\theta(\mathbf{y}|\mathbf{x}) \\ \mathbf{x}\sim\mathcal{D}}} \mathbf{g}\mathbf{g}^\top \approx \mathbb{E}_{(\mathbf{x},\mathbf{y})\sim\mathcal{D}} \; \mathbf{g}\mathbf{g}^\top$$

multi backward passes

log-likelihood (CE) gradient

empirical Fisher (one fwd-bwd)

$$D_{KL}(p_\theta \| p_{\theta+\Delta\theta}) \approx \frac{1}{2}\Delta\theta^\top F_\theta \Delta\theta \approx \frac{1}{2}\sum_i (\Delta\theta_i)^2 \underbrace{\mathbb{E}_\mathcal{D}\left(g_i^2\right)}_{F_{\theta_i}}$$

2nd order approx

- empirical
- independence

- over batches
- moving-average
- storage: one set of Fisher (diagonal)

Fisher based parameter importance (2/2)

$$F_\theta = \mathbb{E}_{\substack{\mathbf{y}\sim p_\theta(\mathbf{y}|\mathbf{x}) \\ \mathbf{x}\sim\mathcal{D}}} \mathbf{g}\mathbf{g}^\top \approx \mathbb{E}_{(\mathbf{x},\mathbf{y})\sim\mathcal{D}} \; \mathbf{g}\mathbf{g}^\top$$

multi backward passes

log-likelihood (CE) gradient

empirical Fisher (one fwd-bwd)

$$D_{KL}(p_\theta \| p_{\theta+\Delta\theta}) \approx \frac{1}{2}\Delta\theta^\top F_\theta \Delta\theta \approx \frac{1}{2}\sum_i (\Delta\theta_i)^2 \underbrace{\mathbb{E}_{\mathcal{D}}\left(g_i^2\right)}_{F_{\theta_i}}$$

2nd order approx

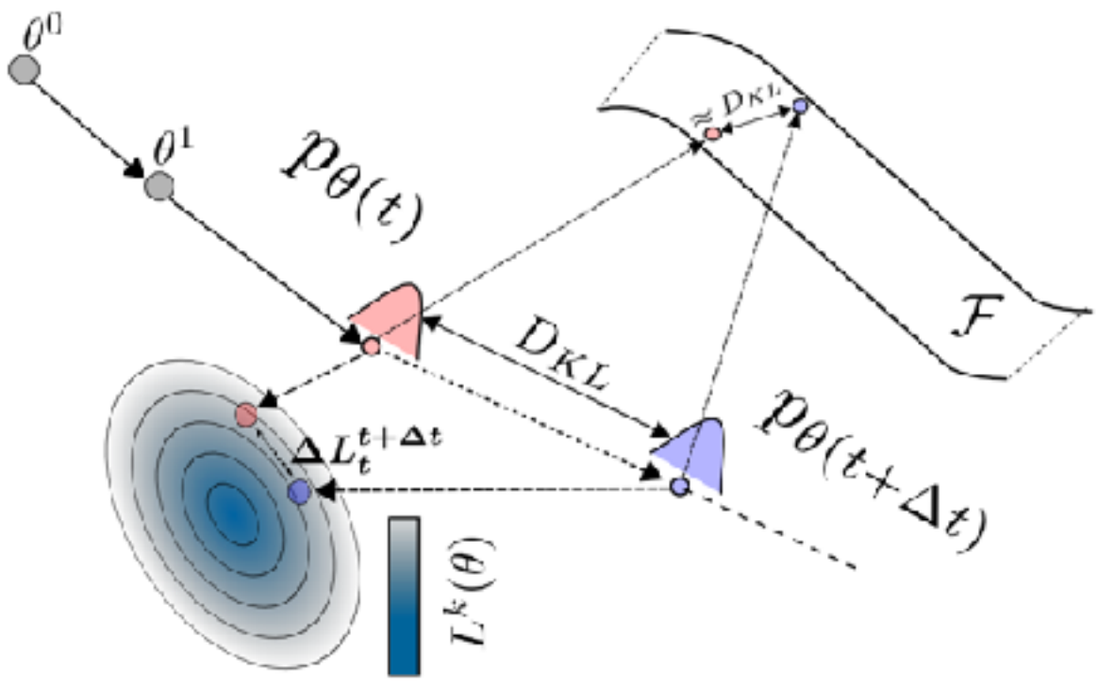
- empirical
- independence

- over batches
- moving-average
- storage: one set of Fisher (diagonal)

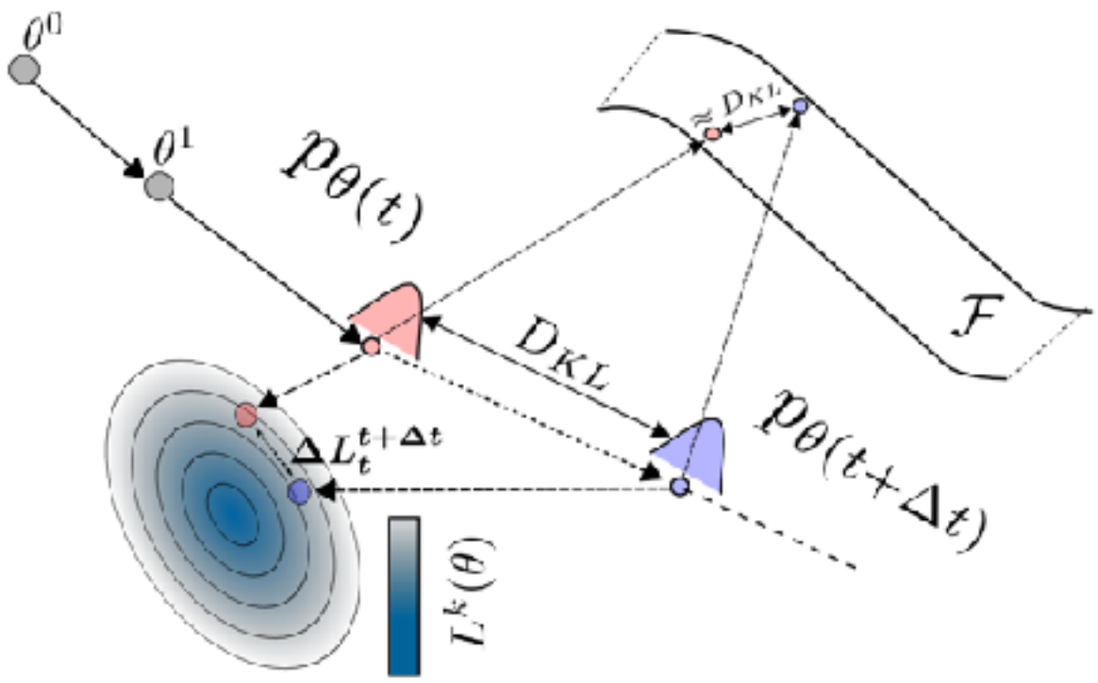
- Fisher approximates the **local curvature** of the KL-divergence surface
- **Higher Fisher**, higher curvature, **more sensitive**, move less in that direction

Optimization path based parameter importance

Optimization path based parameter importance

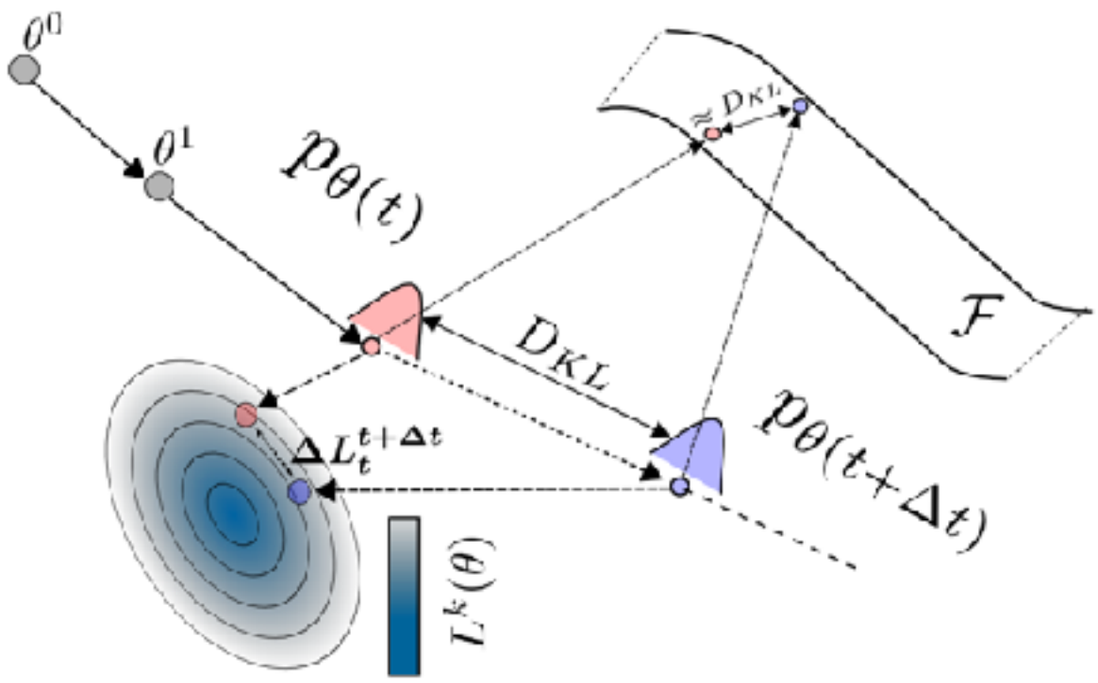


Optimization path based parameter importance



$$s_i = \frac{\Delta L_t^{t+\Delta t}(\theta_i)}{\Delta D_{KL}(\theta_i)}$$

Optimization path based parameter importance

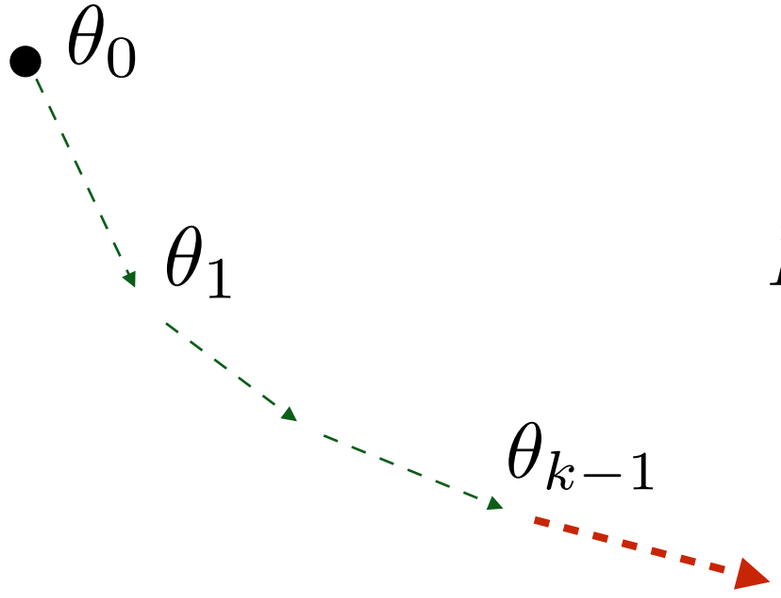


$$s_i = \frac{\Delta L_t^{t+\Delta t}(\theta_i)}{\Delta D_{KL}(\theta_i)}$$

Change in the loss per unit movement in the Riemannian manifold defined by the Fisher (approx. KL-div)

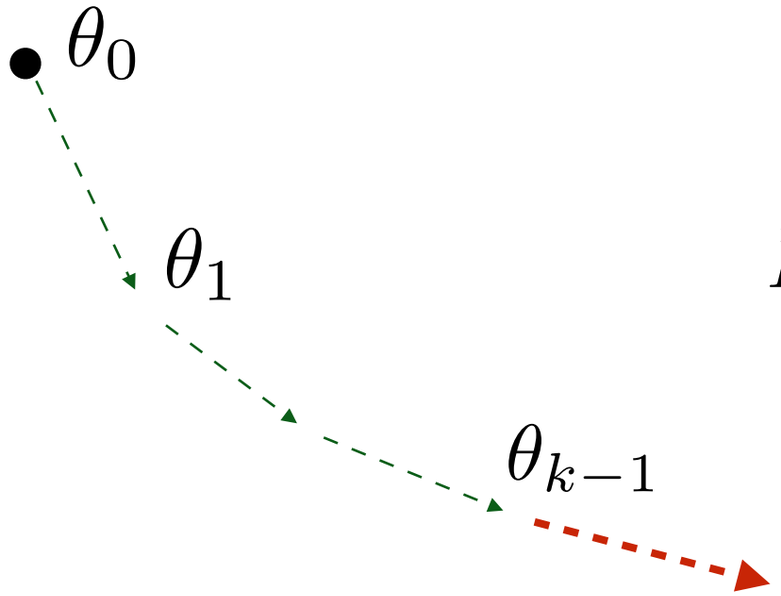
Final Objective Again (RWalk)
(combining Fisher and Optimization path)

Final Objective Again (RWalk)
(combining Fisher and Optimization path)



$$\tilde{L}^k(\theta) = L^k(\theta) + \lambda \sum_i (F_{\theta_i^{k-1}} + s_{t_0}^{t_{k-1}})(\theta_i - \theta_i^{k-1})^2$$

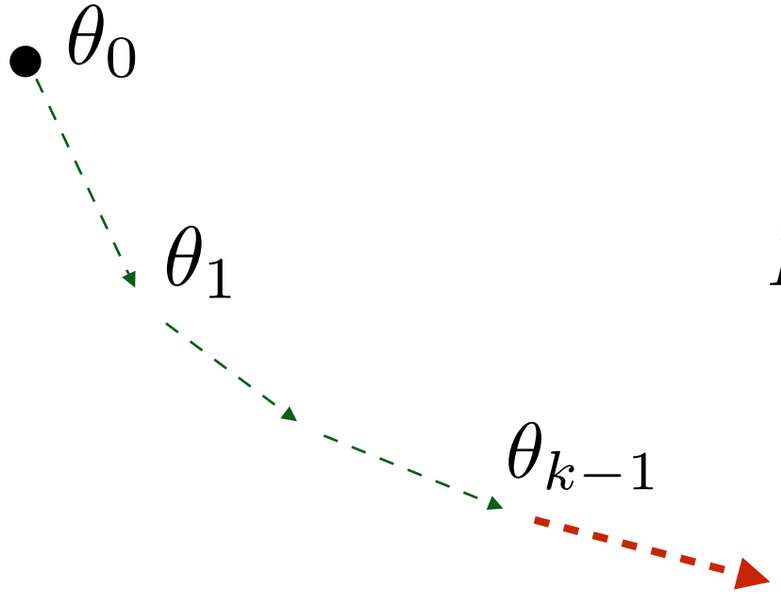
Final Objective Again (RWalk)
(combining Fisher and Optimization path)



$$\tilde{L}^k(\theta) = L^k(\theta) + \lambda \sum_i (F_{\theta_i^{k-1}} + s_{t_0}^{t_{k-1}})(\theta_i - \theta_i^{k-1})^2$$

- If no Fisher and Euclidean distance = Path Integral [Zenke et al, ICML17]

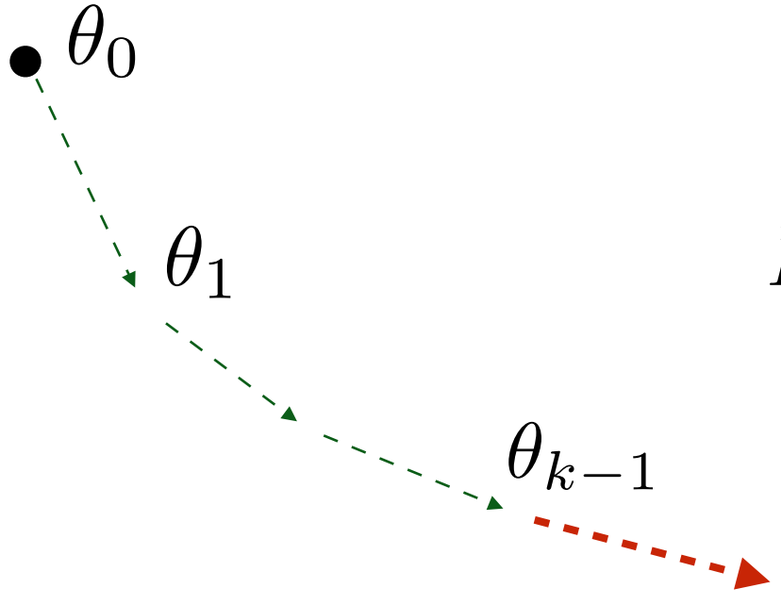
Final Objective Again (RWalk) (combining Fisher and Optimization path)



$$\tilde{L}^k(\theta) = L^k(\theta) + \lambda \sum_i (F_{\theta_i^{k-1}} + s_{t_0}^{t_{k-1}})(\theta_i - \theta_i^{k-1})^2$$

- If no Fisher and Euclidean distance = Path Integral [Zenke et al, ICML17]
- If no optimization path score
  - EWC [Kirkpatrick et al, PNAS16] with only one set of Fisher (similar idea got accepted in ICML18 [Schwarz et al, ICML18])

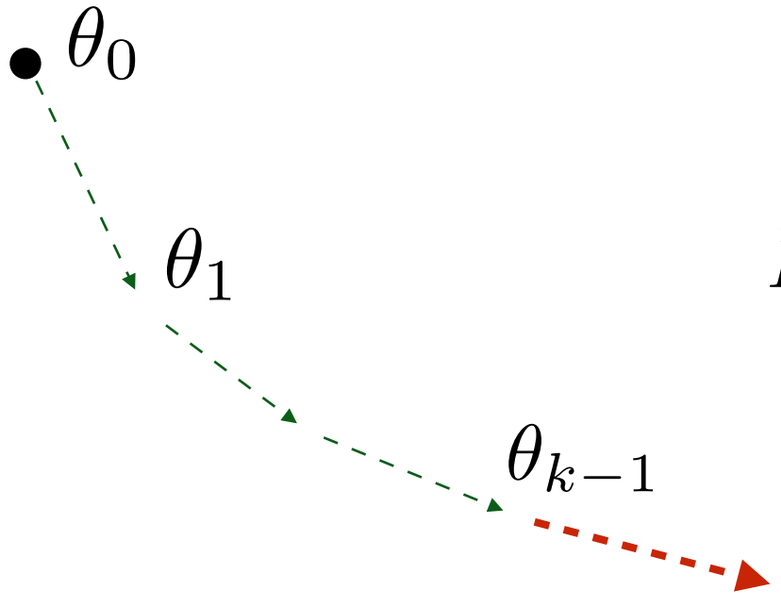
Final Objective Again (RWalk) (combining Fisher and Optimization path)



$$\tilde{L}^k(\theta) = L^k(\theta) + \lambda \sum_i (F_{\theta_i^{k-1}} + s_{t_0}^{t_{k-1}})(\theta_i - \theta_i^{k-1})^2$$

- If no Fisher and Euclidean distance = Path Integral [Zenke et al, ICML17]
- If no optimization path score
  - EWC [Kirkpatrick et al, PNAS16] with only one set of Fisher (similar idea got accepted in ICML18 [Schwarz et al, ICML18])
- Fisher updated using moving average and batch statistics
  - efficient as no need to pass the entire data
  - comes as by-product

Final Objective Again (RWalk) (combining Fisher and Optimization path)



$$\tilde{L}^k(\theta) = L^k(\theta) + \lambda \sum_i (F_{\theta_i^{k-1}} + s_{t_0}^{t_{k-1}})(\theta_i - \theta_i^{k-1})^2$$

- If no Fisher and Euclidean distance = Path Integral [Zenke et al, ICML17]
- If no optimization path score
  - EWC [Kirkpatrick et al, PNAS16] with only one set of Fisher (similar idea got accepted in ICML18 [Schwarz et al, ICML18])
- Fisher updated using moving average and batch statistics
  - efficient as no need to pass the entire data
  - comes as by-product
- Fisher and Scores individually normalized to [0,1]
  - makes lambda less sensitive to the num of tasks

Handling Forgetting and Intransigence

Handling Forgetting and Intransigence

$$\tilde{L}^k(\theta) = L^k(\theta) + \lambda \sum_i (F_{\theta_i^{k-1}} + s_{t_0}^{t_{k-1}})(\theta_i - \theta_i^{k-1})^2$$

- Encodes previous knowledge
- Handles forgetting

Handling Forgetting and Intransigence

$$\tilde{L}^k(\theta) = L^k(\theta) + \lambda \sum_i (F_{\theta_i^{k-1}} + s_{t_0}^{t_{k-1}})(\theta_i - \theta_i^{k-1})^2$$

- Encodes previous knowledge
- Handles forgetting

- Handling Intransigence (inability to learn)? Big problem for single-head evaluation
  - Recall single-head is evaluated over all the labels seen
  - Requires learning discrimination b/w previous and new labels

Handling Forgetting and Intransigence

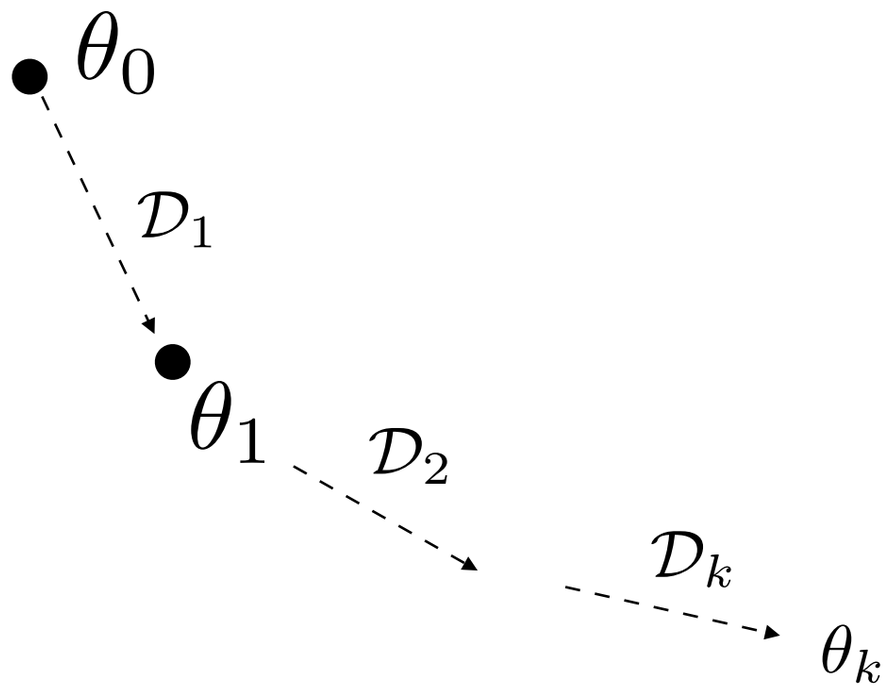
$$\tilde{L}^k(\theta) = L^k(\theta) + \lambda \sum_i (F_{\theta_i^{k-1}} + s_{t_0}^{t_{k-1}})(\theta_i - \theta_i^{k-1})^2$$

- Encodes previous knowledge
- Handles forgetting

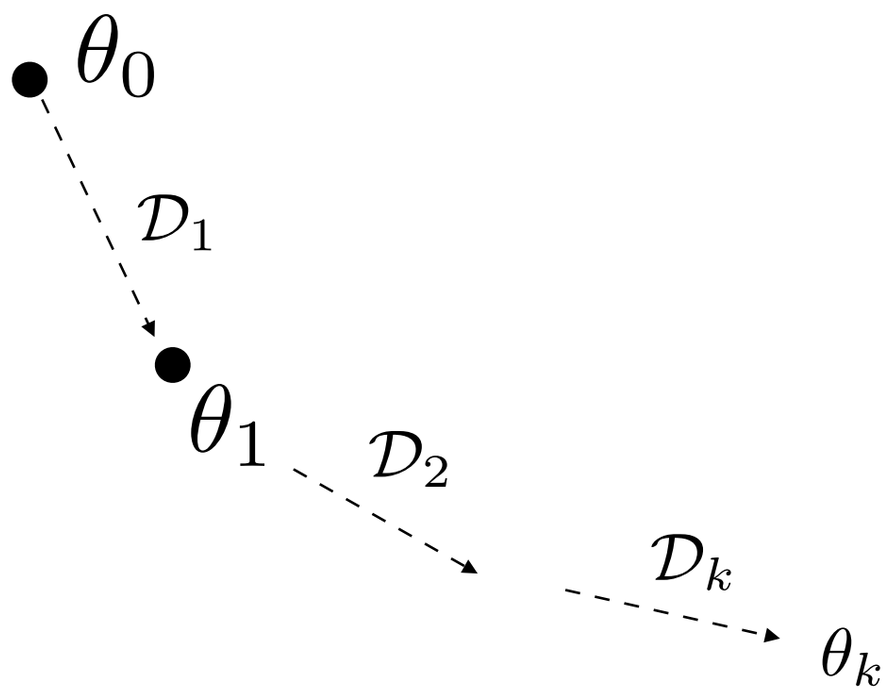
- Handling Intransigence (inability to learn)? Big problem for single-head evaluation
  - Recall single-head is evaluated over all the labels seen
  - Requires learning discrimination b/w previous and new labels
- Store few samples from previous tasks:
  - Uniform
  - Plane distance based: Better if closer to the boundary
  - Entropy based: Higher entropy, more difficult the sample
  - Mean of features (MoF)

Quantitative Evaluation of Incremental Learning

Quantitative Evaluation of Incremental Learning

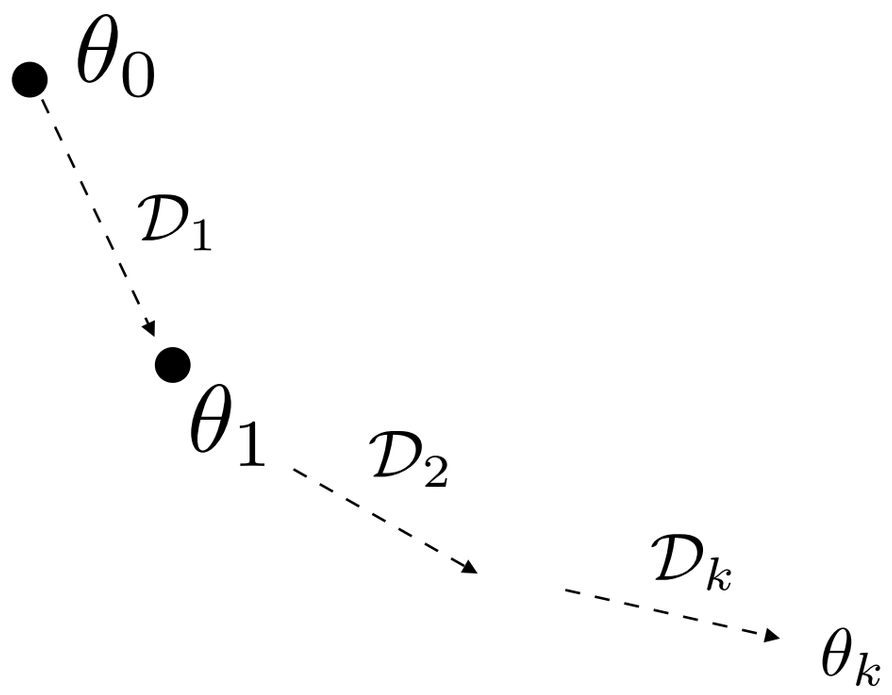


Quantitative Evaluation of Incremental Learning



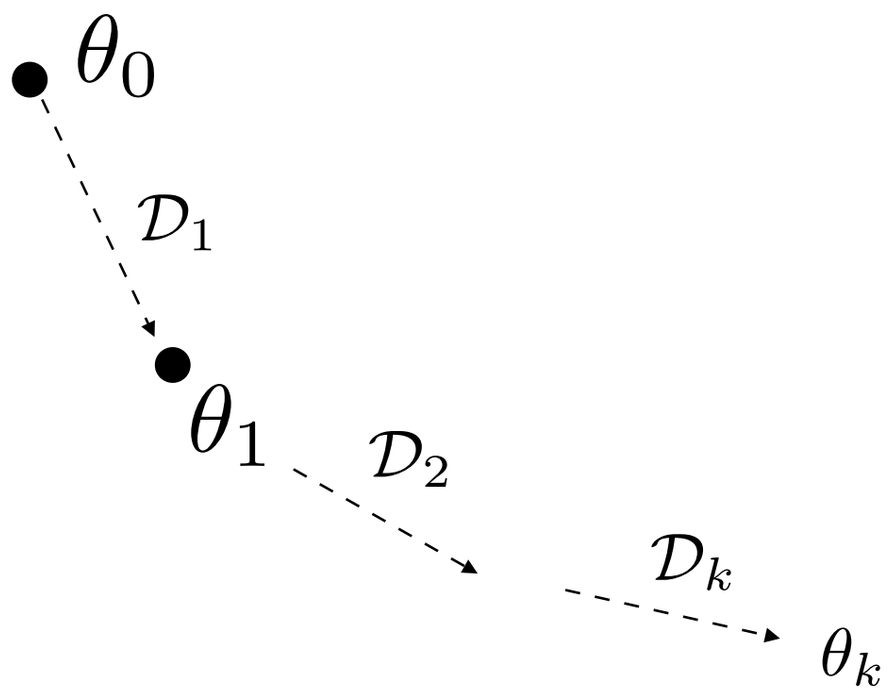
- Standard Approach: Compute accuracy at the end of all the tasks

Quantitative Evaluation of Incremental Learning



- Standard Approach: Compute accuracy at the end of all the tasks
- Doesn't capture the behaviour of the model throughout the learning path

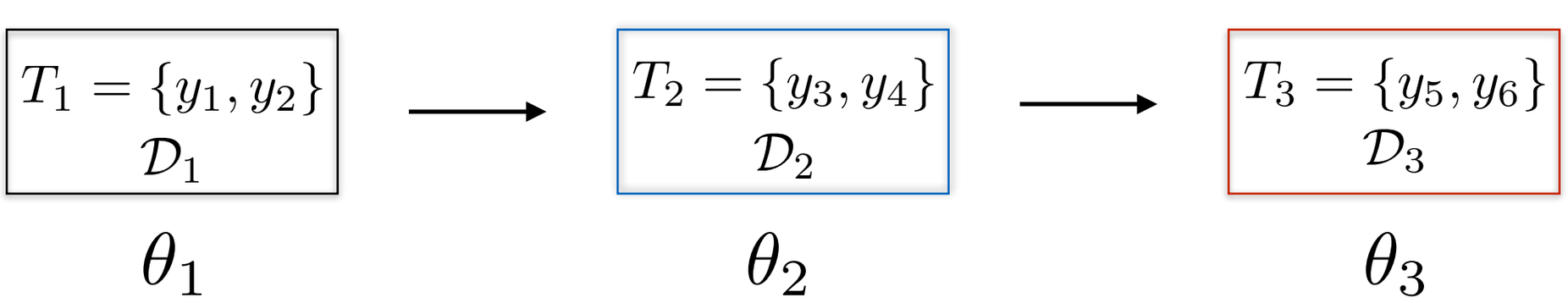
Quantitative Evaluation of Incremental Learning



- Standard Approach: Compute accuracy at the end of all the tasks
- Doesn't capture the behaviour of the model throughout the learning path
- Doesn't tell us anything about Forgetting and Instrasigence

Forgetting

Forgetting


$T_1 = \{y_1, y_2\}$
$\mathcal{D}_1$
$T_2 = \{y_3, y_4\}$
$\mathcal{D}_2$
$T_3 = \{y_5, y_6\}$
$\mathcal{D}_3$
θ_1
θ_2
θ_3

Forgetting

$T_1 = \{y_1, y_2\}$ $\mathcal{D}_1$ → $T_2 = \{y_3, y_4\}$ $\mathcal{D}_2$ → $T_3 = \{y_5, y_6\}$ $\mathcal{D}_3$

θ_1 θ_2 θ_3

$$f_1^3 =$$

Forgetting

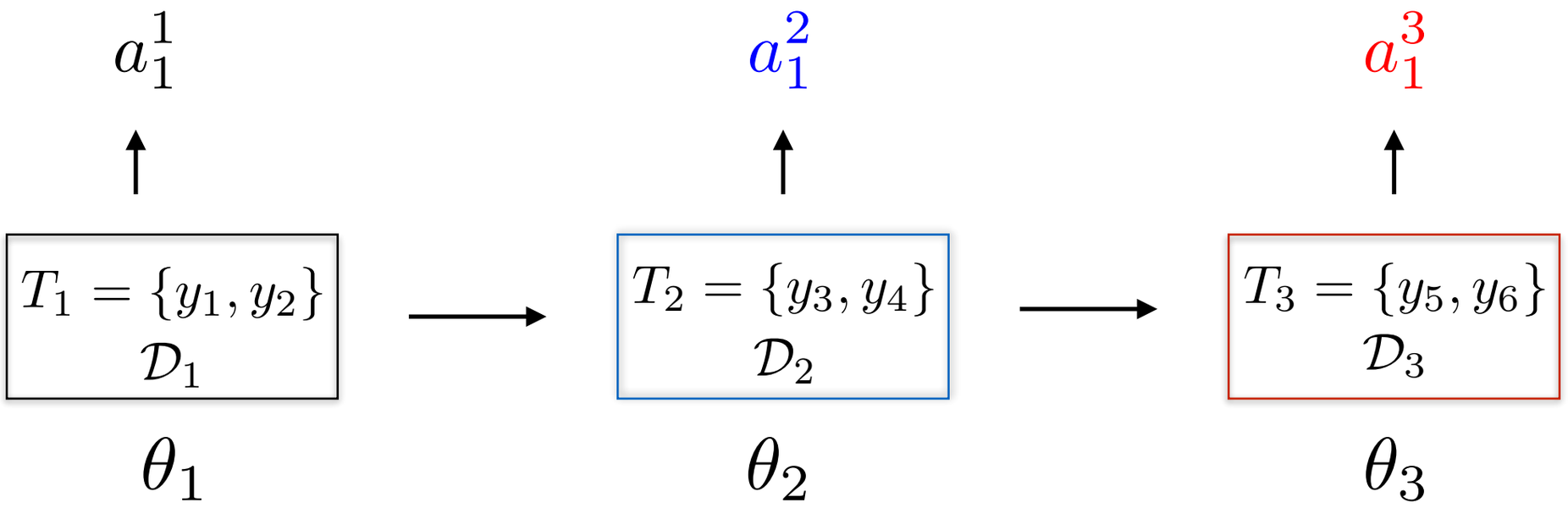
a_1^1 a_1^2 a_1^3

$T_1 = \{y_1, y_2\}$ $\mathcal{D}_1$ → $T_2 = \{y_3, y_4\}$ $\mathcal{D}_2$ → $T_3 = \{y_5, y_6\}$ $\mathcal{D}_3$

θ_1 θ_2 θ_3

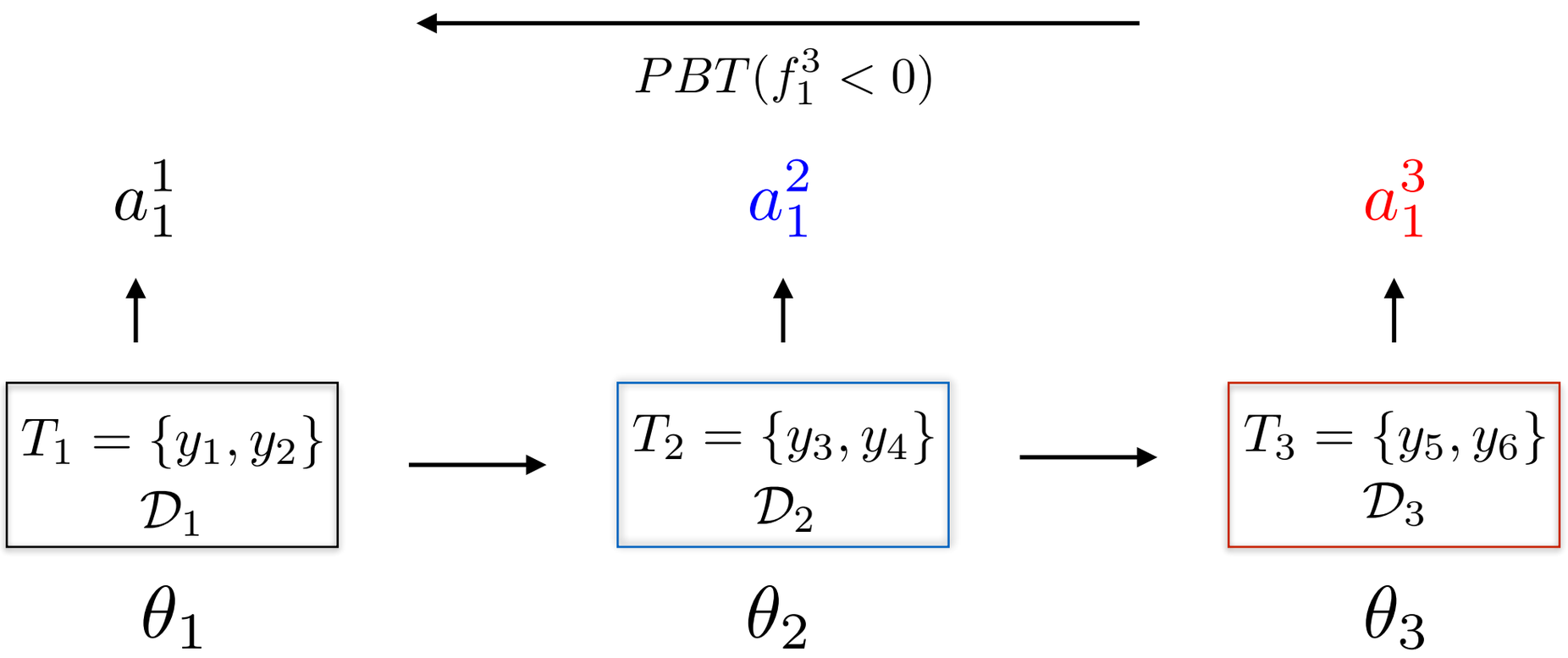
$f_1^3 =$

Forgetting



$$f_1^3 = \max(a_1^1, a_1^2) - a_1^3$$

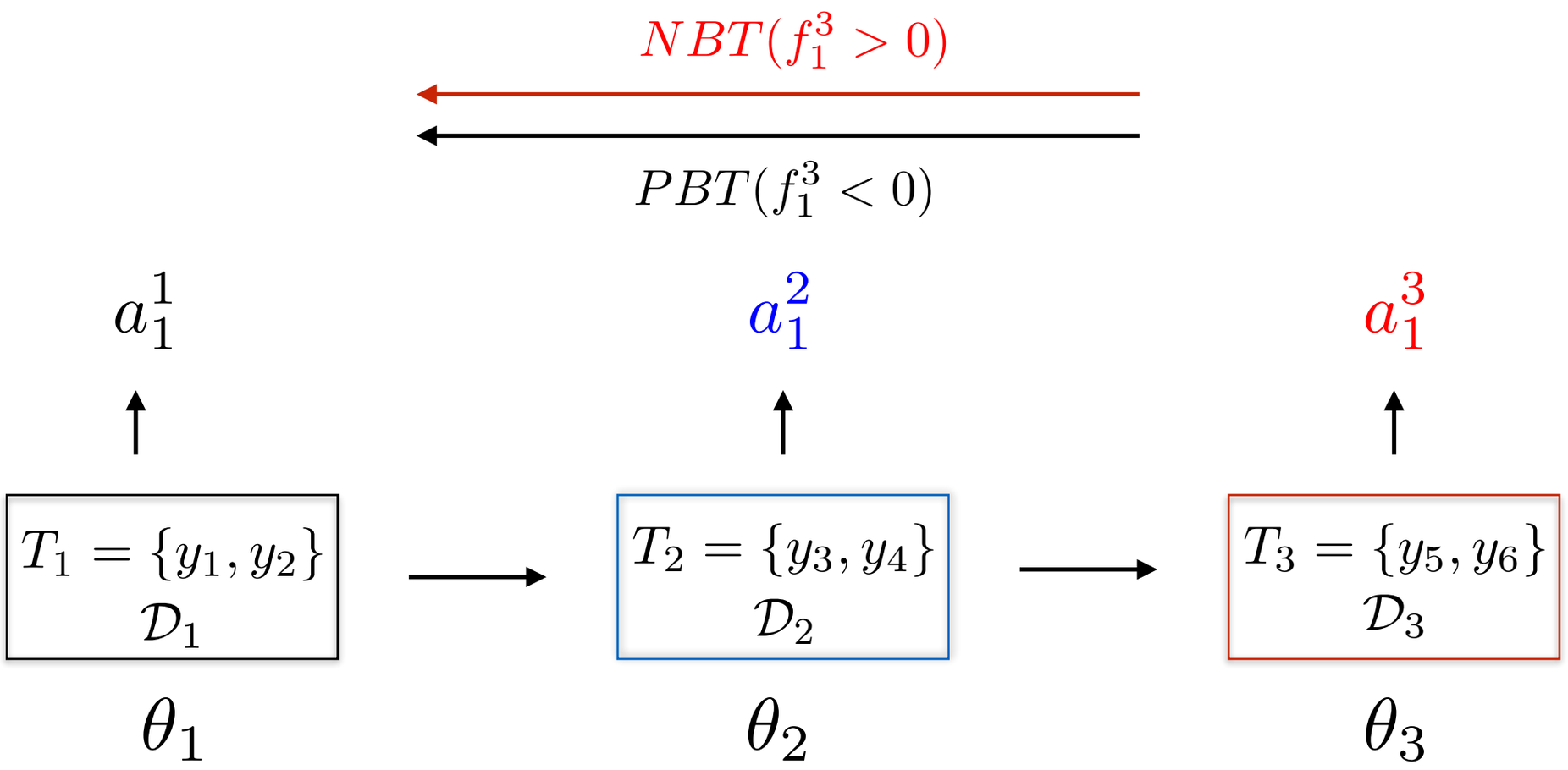
Forgetting



$$f_1^3 = \max(a_1^1, a_1^2) - a_1^3$$

- **PBT** (Positive Backward Transfer): Learning task k **helped improve** knowledge about previous task j (j<k)

Forgetting

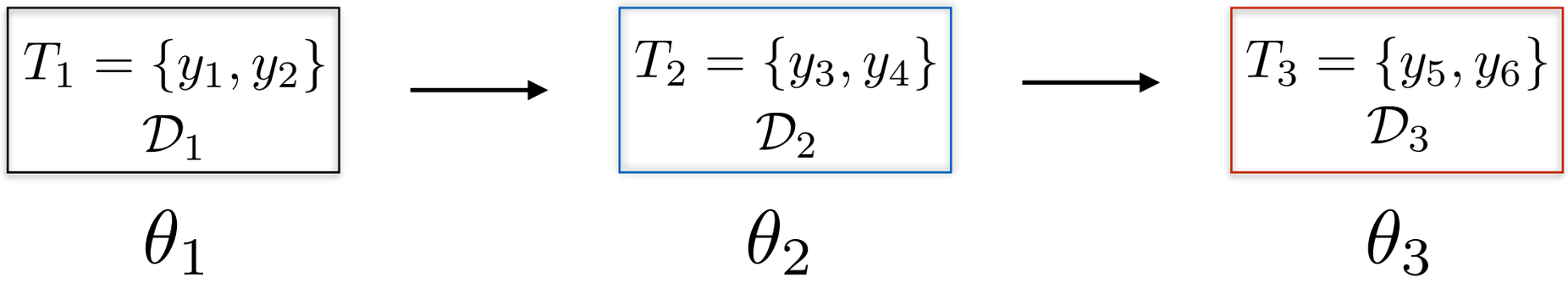


$$f_1^3 = \max(a_1^1, a_1^2) - a_1^3$$

- **PBT** (Positive Backward Transfer): Learning task k **helped improve** knowledge about previous task j (j<k)
- **NBT** (Negative Backward Transfer): Learning task k forced the model to **forget** some information about previous task j (j<k)

Intransigence (Inability to learn)

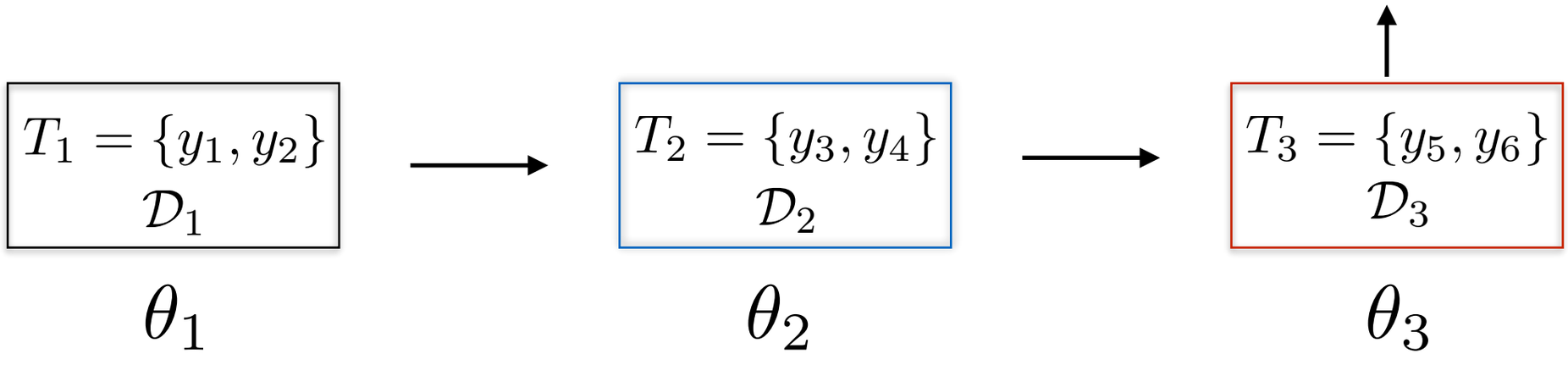
Intransigence (Inability to learn)



Intransigence (Inability to learn)

$$I_5^3 =$$

a_5^3



Intransigence (Inability to learn)

$T = \{y_1, \cdots, y_6\}$

$\mathcal{D} = \cup_{i=1}^{3} \mathcal{D}_i$

$\longrightarrow a_5^*$

θ^*

$I_5^3 =$

a_5^3

$T_1 = \{y_1, y_2\}$

$\mathcal{D}_1$

θ_1

$\longrightarrow$

$T_2 = \{y_3, y_4\}$

$\mathcal{D}_2$

θ_2

$\longrightarrow$

$T_3 = \{y_5, y_6\}$

$\mathcal{D}_3$

θ_3

Intransigence (Inability to learn)

$T = \{y_1, \cdots, y_6\}$

$\mathcal{D} = \cup_{i=1}^{3} \mathcal{D}_i$

$\longrightarrow a_5^*$

θ^*

$$I_5^3 = a_5^* - a_5^3$$

a_5^3

$T_1 = \{y_1, y_2\}$
$\mathcal{D}_1$

θ_1

$\longrightarrow$

$T_2 = \{y_3, y_4\}$
$\mathcal{D}_2$

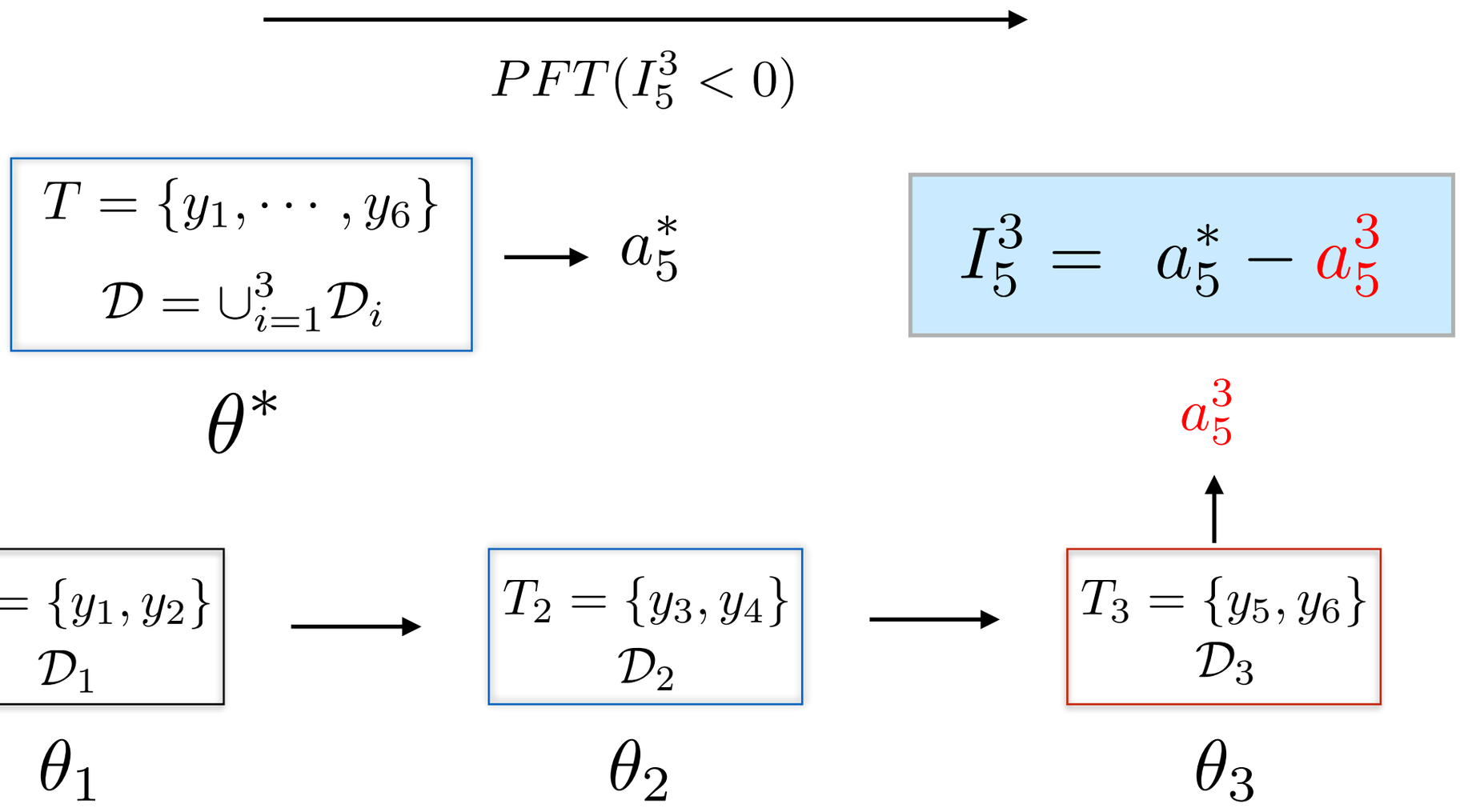
θ_2

$\longrightarrow$

$T_3 = \{y_5, y_6\}$
$\mathcal{D}_3$

θ_3

Intransigence (Inability to learn)



- **PFT** (Positive Forward Transfer): Incrementally learning task k **improves** model's knowledge about it.

Intransigence (Inability to learn)

$NFT(I_5^3 > 0)$

$PFT(I_5^3 < 0)$

$T = \{y_1, \cdots, y_6\}$

$\mathcal{D} = \cup_{i=1}^{3} \mathcal{D}_i$

$\longrightarrow a_5^*$

$$I_5^3 = a_5^* - a_5^3$$

θ^*

a_5^3

$T_1 = \{y_1, y_2\}$ $\mathcal{D}_1$ $\longrightarrow$ $T_2 = \{y_3, y_4\}$ $\mathcal{D}_2$ $\longrightarrow$ $T_3 = \{y_5, y_6\}$ $\mathcal{D}_3$

θ_1 θ_2 θ_3

- **PFT** (Positive Forward Transfer): Incrementally learning task k **improves** model's knowledge about it.
- **NFT** (Negative Forward Transfer): Incrementally learning task k had **negative influence** on it.

Experimental Settings

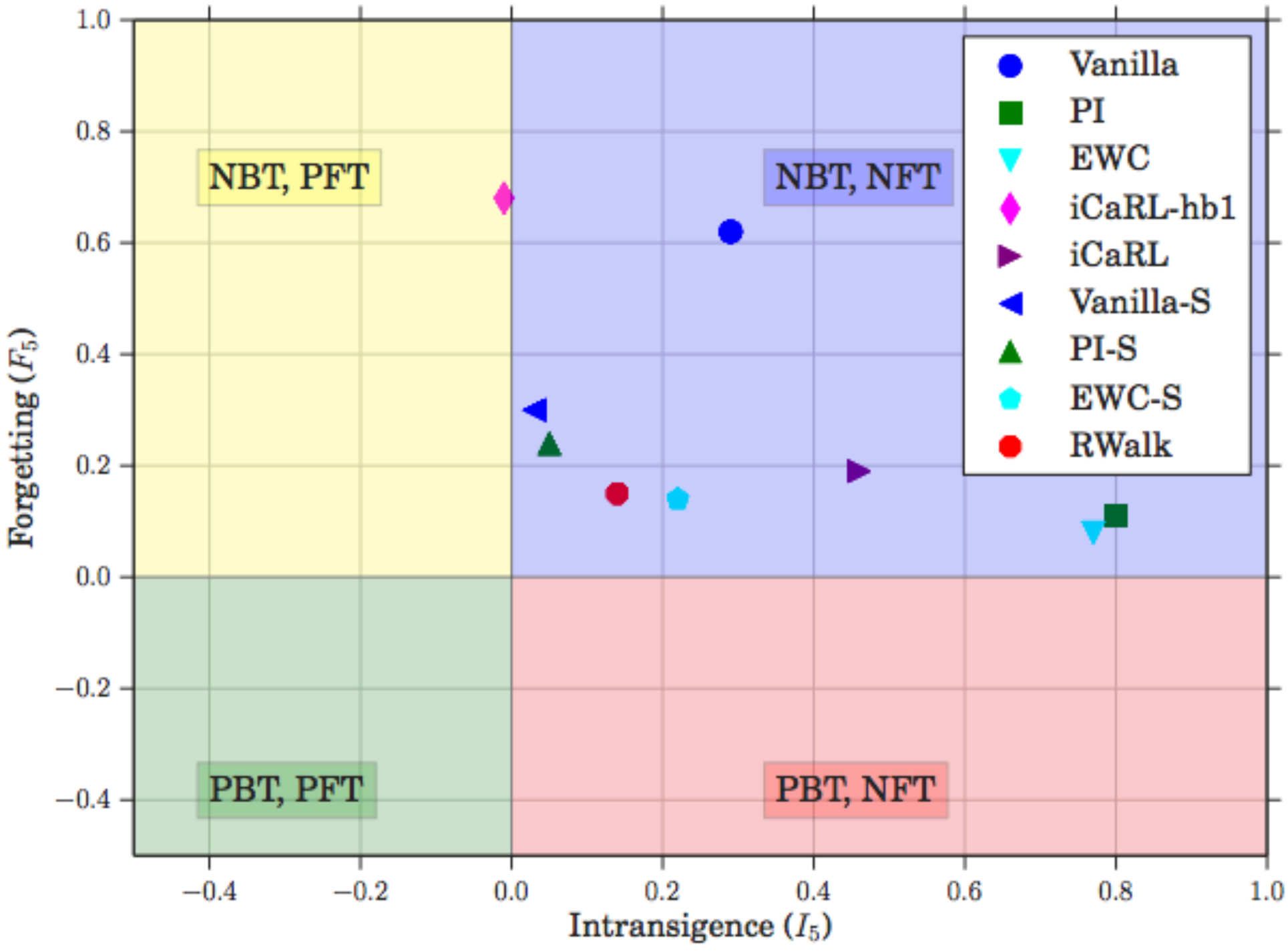
- Incremental MNIST
  - Split into five disjoint subsets
  - Network: MLP
- Incremental CIFAR-100
  - Split into ten disjoint subset
  - Network: CNN
- Optimizer: Adam, batch size: 64

Results

Methods	MNIST				CIFAR			
	Multi-head Evaluation							
	λ	$A_5(\%)$	F_5	I_5	λ	$A_{10}(\%)$	F_{10}	I_{10}
Vanilla	0	90.3	0.12	6.6×10^{-4}	0	44.4	0.36	0.02
EWC [7]	75000	**99.3**	0.001	0.01	3×10^6	72.8	0.001	0.07
PI [25]	0.1	**99.3**	0.002	0.01	10	73.2	0	0.06
RWalk (Ours)	1000	**99.3**	0.003	0.01	1000	**74.2**	0.004	0.04
	Single-head Evaluation							
Vanilla	0	38.0	0.62	0.29	0	10.2	0.36	-0.06
EWC [7]	75000	55.8	0.08	0.77	3×10^6	23.1	0.03	0.17
PI [25]	0.1	57.6	0.11	0.8	10	22.8	0.04	0.2
iCaRL-hb1 [21]	-	36.6	0.68	-0.01	-	7.4	0.40	0.06
iCaRL [21]	-	55.8	0.19	0.46	-	9.5	0.11	0.35
Vanilla-S	0	73.7	0.30	0.03	0	12.9	0.64	-0.3
EWC-S	75000	79.7	0.14	0.22	15×10^5	33.6	0.27	-0.05
PI-S	0.1	78.7	0.24	0.05	10	33.6	0.27	-0.03
RWalk (Ours)	1000	**82.5**	0.15	0.14	500	**34.0**	0.28	-0.06

For MNIST and CIFAR, 10 (0.2%) and 25(5%) samples are used from the previous tasks using mean of features (MoF) sampling. Baselines where samples are used are appended with '-S'.

Forgetting vs Intransigence Trade-off (MNIST)



(a) MNIST

Few open problems and insights (based on our recent analysis work)

Few open problems and insights (based on our recent analysis work)

- ResNets?
  - Need more data

Few open problems and insights (based on our recent analysis work)

- ResNets?
  - Need more data
- Use generative models
  - Encoder-Decoder type to learn data-distribution?

Few open problems and insights (based on our recent analysis work)

- ResNets?
  - Need more data
- Use generative models
  - Encoder-Decoder type to learn data-distribution?
- Saddle Point?

Few open problems and insights (based on our recent analysis work)

- ResNets?
  - Need more data
- Use generative models
  - Encoder-Decoder type to learn data-distribution?
- Saddle Point?
- Order of tasks, subset of labels?

Few open problems and insights (based on our recent analysis work)

- ResNets?
  - Need more data
- Use generative models
  - Encoder-Decoder type to learn data-distribution?
- Saddle Point?
- Order of tasks, subset of labels?
- Judge when to stop training on more tasks? Saturate?