NeurIPS 2020

주제 : Meta-learning

Reviewed by Kyeong Ryeol Go

Contents

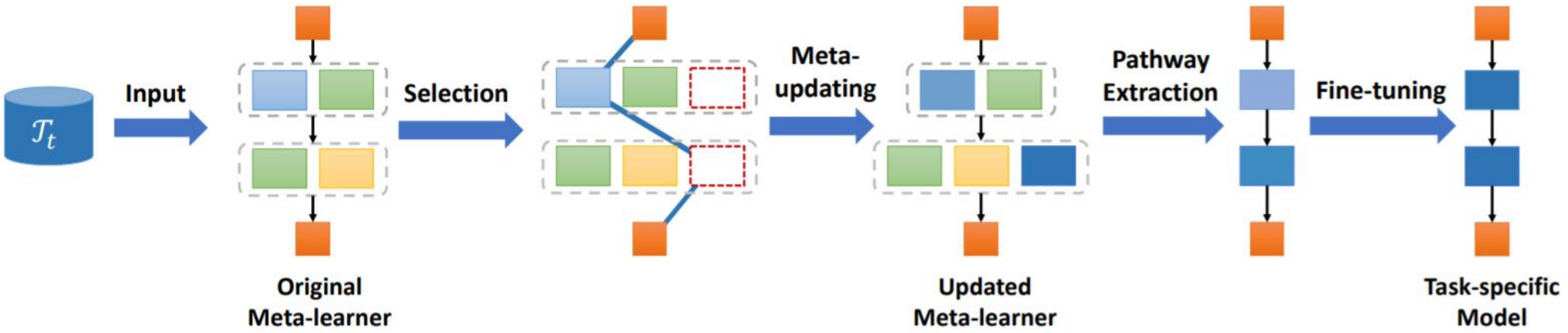
- Task adaptive customization
  - Online structured meta-learning
  - Structured prediction for conditional meta-learning
  - MATE : Plugging in model awareness to task embedding for meta-learning
- Generalizing framework
  - Bayesian meta-learning for the few-shot setting via deep kernels
  - Continuous meta-learning without tasks

Online structured meta-learning

- Well-organized meta-learner
  - Benefit fast adaptation with task-specific prior
  - Accumulate and organize the newly learned experience
  - Automatically adapt and expand for unseen knowledge

- Previous works
  - Modulation relies on a well-trained task representation network
    - Yao, Huaxiu, et al. "Hierarchically structured meta-learning." *arXiv preprint arXiv:1905.05301* (2019).
  - Construct a totally new meta-learner for dissimilar task
    - Jerfel, Ghassen, et al. "Reconciling meta-learning and continual learning with online mixtures of tasks." *Advances in Neural Information Processing Systems*. 2019.

Online structured meta-learning



* $w_{0,t}$ | w_t | $g_{l,t}$ | o_{b_l} | K : initial | adapted | layer output | block importance | # of tasks sharing blocks

- Selection
  - $g_{l,t} = \sum_{b_l=1}^{B_l+1} \frac{\exp(o_{b_l})}{\sum_{b_l'}^{B_l+1} \exp(o_{b_l'})} w_t g_{l-1,t}$
  - $w_t = w_{0,t} - \alpha \nabla_{w_{0,t}} L(w_{0,t}; D_t^s)$
  - $w_{0,t} = w_{0,t} - \beta_1 \nabla_{w_{0,t}} L(w_t, o; D_t^q)$
  - $o = o - \beta_2 \nabla_o L(w_t, o; D_t^q)$
  - $\boldsymbol{b_l^* = \arg\max_{b_l \in [1, B_l]} o_{b_l}}$
- Meta update
  - $\boldsymbol{w_{0b_l^*,t} = w_{0b_l^*,t} - \beta_3 \sum_{k=1}^{K} \nabla_{w_{b_l^*,k}} L(w_k; D_k^q)}$
  - $w_k = w_{0,k} - \beta_4 \nabla_{w_{0,k}} L(w_{0,k}; D_k^s)$
- Fine tuning
  - $\boldsymbol{w_{b_l^*,t} = w_{0b_l^*,t} - \beta_5 \nabla_{w_{0b_l^*,t}} L(w_{0b_l^*,t}; D_t^s \cup D_t^q)}$

Online structured meta-learning

Constructed knowledge block

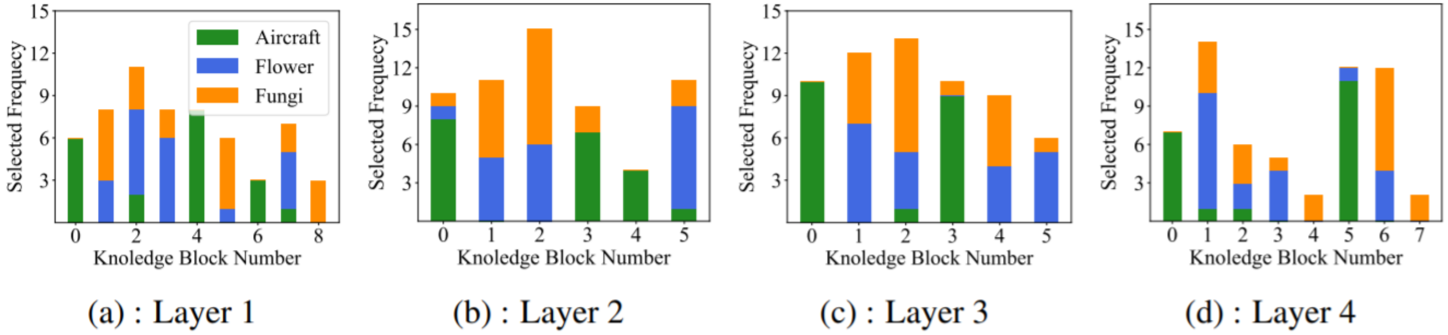


Figure 5: Selected ratio of knowledge blocks for each sub-dataset in Meta-dataset. Figure (a)-(d) illustrate the knowledge blocks in layer 1-4.

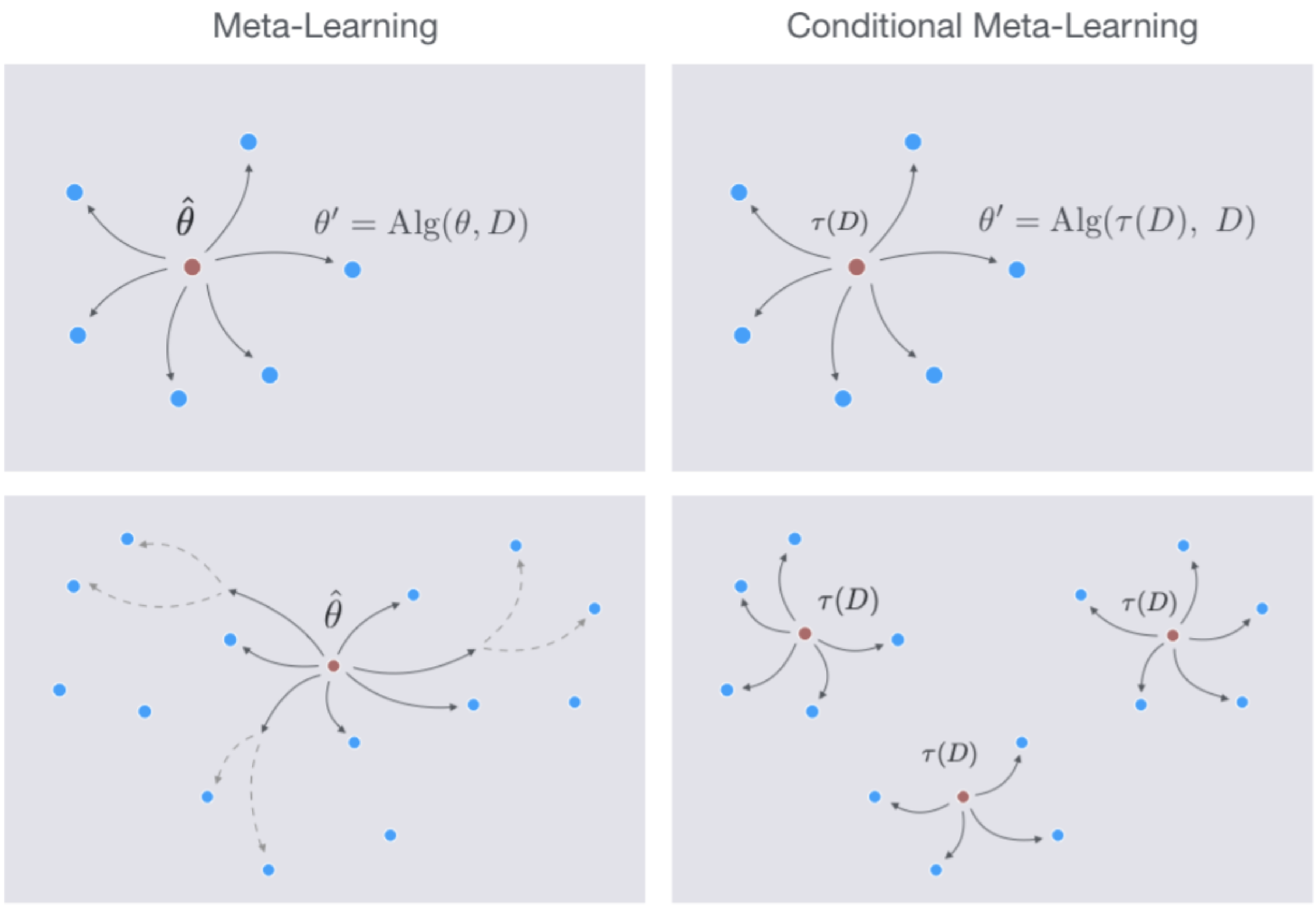
Increased model capacity

Models	Blur Acc.	Night Acc.	Pencil Acc.	Overall Acc.	AR
NT	$49.80 \pm 3.91\%$	$47.70 \pm 2.91\%$	$47.55 \pm 5.18\%$	$53.32 \pm 2.30\%$	-
NT-Large	$43.05 \pm 3.99\%$	$41.30 \pm 2.66\%$	$43.25 \pm 4.24\%$	$42.53 \pm 2.15\%$	3.92
FT	$51.50 \pm 4.90\%$	$49.00 \pm 3.82\%$	$50.90 \pm 5.30\%$	$50.47 \pm 2.73\%$	-
FT-Large	$54.60 \pm 2.99\%$	$50.35 \pm 2.64\%$	$52.45 \pm 3.92\%$	$52.46 \pm 1.91\%$	2.82
FTML	$58.90 \pm 3.52\%$	$56.40 \pm 3.53\%$	$54.60 \pm 5.46\%$	$56.63 \pm 2.50\%$	-
FTML-Large	$57.55 \pm 3.76\%$	$56.70 \pm 3.92\%$	$56.30 \pm 4.05\%$	$56.85 \pm 2.26\%$	2.13
OSML	$\mathbf{64.10 \pm 3.12\%}$	$\mathbf{65.25 \pm 3.24\%}$	$\mathbf{60.35 \pm 3.65\%}$	$\mathbf{63.23 \pm 2.00\%}$	**1.13**

Structured prediction for conditional meta-learning

- Motivation
  - Relying on the shared parameter is challenging for complex task distribution
  - Previous works lack theoretical guarantees on generalization performance
    - Rusu, Andrei A., et al. "Meta-learning with latent embedding optimization." *arXiv preprint arXiv:1807.05960* (2018).

- Conditional meta learning
  - $\min_{\theta} \frac{1}{N} \sum_{i=1}^{N} L\left(Alg(\theta, D_i^s), D_i^q\right) \rightarrow \min_{\tau: D \rightarrow \Theta} \frac{1}{N} \sum_{i=1}^{N} L\left(Alg\left(\tau(D_i^{tr}), D_i^{tr}\right), D_i^{val}\right)$
  - Superior when the larger the number of clusters
    - Denevi, Giulia, Massimiliano Pontil, and Carlo Ciliberto. "The advantage of conditional meta-learning for biased regularization and fine-tuning." *arXiv preprint arXiv:2008.10857* (2020).



Meta-Learning
Conditional Meta-Learning
$\hat{\theta}$
$\theta' = \mathrm{Alg}(\theta, D)$
$\tau(D)$
$\theta' = \mathrm{Alg}(\tau(D),\ D)$
$\hat{\theta}$
$\tau(D)$
$\tau(D)$
$\tau(D)$

Structured prediction for conditional meta-learning

- Structured prediction → direct sup. Learn. Not allowed
  - $X \mid Y \mid Z$: input | label | output spaces where Z is not linear (structured) (ex. strings, graphs, points on a manifold, probability distributions …)
  - $\min_{f:X \to Z} \int g(f(x), y \mid x)\ d\rho(x, y)\ \ where\ \ g: X \times Y \times Z \to R$
  - Estimator : $\hat{f}(x) = \arg\min_{z} \sum_{i=1}^{n} \alpha_i(x) g(z, y_i | x_i)$
    - C. Ciliberto, F. Bach, and A. Rudi. Localized structured prediction. In Advances in Neural Information Processing Systems, 2019.

- Task-adaptive structured meta-learning (TASML)
  - $\hat{\tau}(D) = \arg\min_{\theta \in \Theta} \sum_{i=1}^{N} \alpha_i(D) L\left(Alg\left(\theta, D_i^{tr}\right), D_i^{val}\right)$
  - $\alpha(D) = (K + \lambda I)^{-1} v(D)\ \ where\ \ K_{ij} = k\left(D_i^{tr}, D_j^{tr}\right)\ \ and\ \ v_i(D) = k\left(D_i^{tr}, D\right)$
  - $k(D, D') = \exp\left(-\frac{\left\|\bar{\phi}(D) - \bar{\phi}(D')\right\|^2}{\sigma^2}\right),\ \ \bar{\phi}(D) = \frac{1}{|D|} \sum_{(x,y) \in D} \phi(x)$
    (measure the relevance of known training tasks to target tasks)

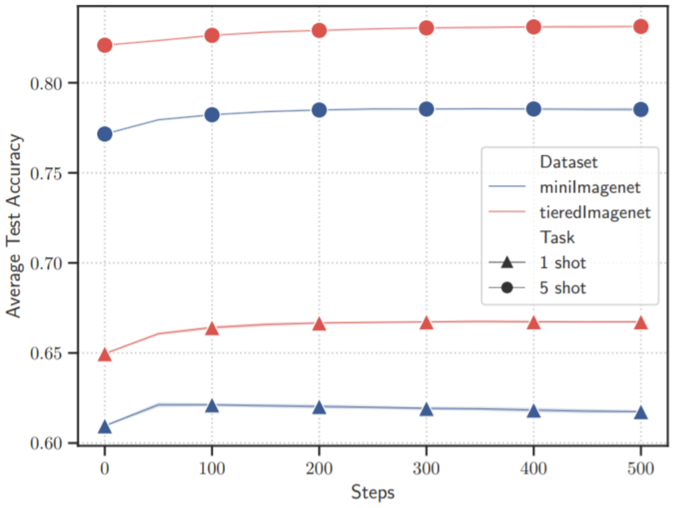
Structured prediction for conditional meta-learning

	ACCURACY (%)			
	*mini*IMAGENET		*tiered*IMAGENET	
CONDITIONAL METHODS	1-SHOT	5-SHOT	1-SHOT	5-SHOT
(JERFEL ET AL.) [23]	51.46 ± 1.68	65.00 ± 0.96	-	-
HSML [56]	50.38 ± 1.85	-	-	-
MMAML [54]	46.1 ± 1.63	59.8 ± 1.82	-	-
CAML [24]	59.23 ± 0.99	72.35 ± 0.71	-	-
LEO [44]	61.76 ± 0.08	77.59 ± 0.12	66.33 ± 0.05	81.44 ± 0.09
LEO (LOCAL) [44]	60.37 ± 0.74	75.36 ± 0.44	65.11 ± 0.72	79.70 ± 0.59
TASML (OURS)	$\mathbf{62.04 \pm 0.52}$	$\mathbf{78.22 \pm 0.47}$	$\mathbf{66.42 \pm 0.37}$	$\mathbf{82.62 \pm 0.31}$

	1-SHOT	5-SHOT
MAML [17]	58.9 ± 1.9	71.5 ± 1.0
R2D2 [9]	65.3 ± 0.2	79.4 ± 0.1
PROTONET(RESNET12 FEAT.) [47]	72.2 ± 0.7	83.5 ± 0.5
METAOPTNET [29]	72.0 ± 0.7	84.2 ± 0.5
TASML	$\mathbf{74.6 \pm 0.7}$	$\mathbf{85.1 \pm 0.4}$

Improvement from structured prediction

	ACCURACY (%)			
	*mini*IMAGENET		*tiered*IMAGENET	
	1-SHOT	5-SHOT	1-SHOT	5-SHOT
MAML (LEO FEAT.)	54.12 ± 1.84	67.58 ± 0.92	51.28 ± 1.81	69.80 ± 0.84
SP+MAML (LEO FEAT.)	$\mathbf{58.46 \pm 1.56}$	$\mathbf{74.51 \pm 0.75}$	$\mathbf{60.89 \pm 1.64}$	$\mathbf{78.42 \pm 0.73}$
LEO (LOCAL)	60.37 ± 0.74	75.36 ± 0.44	65.11 ± 0.72	79.70 ± 0.59
SP+LEO (LOCAL)	$\mathbf{61.46 \pm 0.69}$	$\mathbf{76.54 \pm 0.59}$	$\mathbf{66.07 \pm 0.66}$	$\mathbf{80.68 \pm 0.41}$
LS META-LEARN	60.19 ± 0.65	76.76 ± 0.43	64.32 ± 0.65	81.43 ± 0.55
TASML (SP + LS META-LEARN)	$\mathbf{62.04 \pm 0.52}$	$\mathbf{78.22 \pm 0.47}$	$\mathbf{66.42 \pm 0.37}$	$\mathbf{82.62 \pm 0.31}$



MATE : plugging in model awareness to task embedding for meta-learning

- Inductive bias decides the hypothesis space $\mathcal{F}$
  1) Capacity
  2) Performance of optimal hypothesis f^*
  3) Difficulty of identifying optimal hypothesis f^*

- Meta-learning – Find suitable inductive bias using prior knowledge

  Ex. parameters for different tasks lie within neighborhood of meta parameter

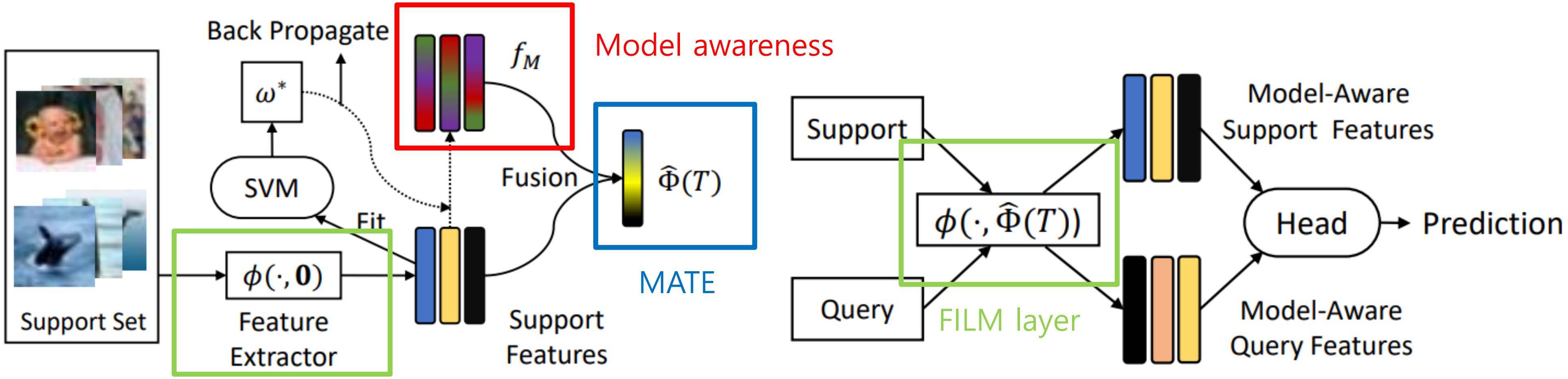
- Model is itself part of the inductive bias
  - Establish a relationship b/t tasks and their corresponding models
  - Exploit not only the data distribution, but also the feature obtained by model

MATE : plugging in model awareness to task embedding for meta-learning

* T : task, P_D : data dist., M : model

- Model-awareness via the incorporation of model into embedding
  - $T = (P_D) \rightarrow T = (P_D, M)$
  - $\Phi(T) \coloneqq \int f_M(x) \odot \phi(x)\, dP_D(x) \approx \frac{1}{K}\sum_{k=1}^{K} f_M(x_k) \odot \phi(x_k)$
  - $f_M(x) = c \cdot \left| \frac{\partial \|w^*\|_2^2}{\partial \phi(x)} \right|$

<u>Select the features that incur the big change in margin (essential features ≈ attention)</u>



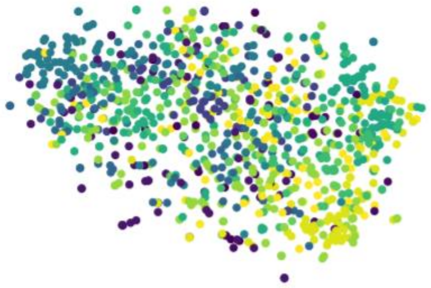
(a) SVM-based model-aware task embedding. (b) Classification with conditioned backbone.

MATE : plugging in model awareness to task embedding for meta-learning

Model	Backbone	CIFAR-FS [6] 5-way 1-shot	CIFAR-FS [6] 5-way 5-shot
MAML$^\diamond$ [10]	32-32-32-32	$58.9 \pm 1.9\%$	$71.5 \pm 1.0\%$
Relation Networks$^\diamond$ [45]	64-96-128-256	$55.0 \pm 1.0\%$	$69.3 \pm 0.8\%$
ProtoNets$^\diamond$ [44]	64-64-64-64	$55.5 \pm 0.7\%$	$72.0 \pm 0.6\%$
ProtoNets [44]	ResNet-12	$71.35 \pm 0.73\%$	$84.07 \pm 0.51\%$
MATE + ProtoNets	ResNet-12	$71.49 \pm 0.70\%$	$84.71 \pm 0.50\%$
R2D2$^\diamond$ [6]	96-192-384-512	$65.3 \pm 0.2\%$	$79.4 \pm 0.1\%$
R2D2 [6]	ResNet-12	$72.51 \pm 0.72\%$	$84.60 \pm 0.50\%$
MATE + R2D2	ResNet-12	$72.59 \pm 0.0.70\%$	$85.04 \pm 0.50\%$
MetaOptNet$^\diamond$ [21]	ResNet-12	$72.0 \pm 0.7\%$	$84.2 \pm 0.5\%$
MATE + MetaOptNet	ResNet-12	$72.3 \pm 0.7\%$	$85.2 \pm 0.4\%$

Model	Backbone	miniImageNet [53] 5-way 1-shot	miniImageNet [53] 5-way 5-shot
Matching Networks$^\diamond$ [53]	64-64-64-64	$43.56 \pm 0.84\%$	$55.31 \pm 0.73\%$
MAML$^\diamond$ [10]	32-32-32-32	$48.70 \pm 1.84\%$	$63.11 \pm 0.92\%$
ProtoNets$^\diamond$ [44]	64-64-64-64	$49.42 \pm 0.78\%$	$68.20 \pm 0.66\%$
Relation Networks$^\diamond$ [45]	64-96-128-256	$50.44 \pm 0.82\%$	$65.32 \pm 0.70\%$
R2D2$^\diamond$ [6]	96-192-384-512	$51.20 \pm 0.60\%$	$68.8 \pm 0.10\%$
LEO$^\diamond$ [40]	WRN-28-10	$61.76 \pm 0.08\%$	$77.59 \pm 0.12\%$
SNAIL$^\diamond$ [27]	ResNet-12	$55.71 \pm 0.99\%$	$68.88 \pm 0.92\%$
AdaResNet$^\diamond$ [30]	ResNet-12	$56.88 \pm 0.62\%$	$71.94 \pm 0.57\%$
TADAM$^\diamond$ [35]	ResNet-12	$58.50 \pm 0.30\%$	$76.70 \pm 0.30\%$
MetaOptNet$^\diamond$ [21]	ResNet-12	$62.64 \pm 0.61\%$	$78.63 \pm 0.46\%$
MetaOptNet† [21]	ResNet-12	$61.64 \pm 0.60\%$	$77.88 \pm 0.46\%$
MATE + MetaOptNet	ResNet-12	$62.08 \pm 0.64\%$	$78.64 \pm 0.46\%$

Cat	FiLM	KME	SVM	Load Backbone	Fix Backbone	FiLM Regularization	1-shot	5-shot
✓		✓					71.41%	84.40%
	✓	✓					71.55%	84.37%
	✓		✓				72.15%	84.72%
	✓		✓	✓			72.01%	85.13%
	✓		✓	✓	✓		**72.57%**	84.76%
	✓		✓	✓		✓	72.32%	**85.20%**



(b) KME + FiLM

(c) SVM + FiLM (MATE)

Bayesian meta-learning for the few-shot setting via deep kernels

- Two levels of inference
  - Inner loop : ρ_t (task-specific parameters)
  - Outer loop : θ (shared parameters)

  <u>(Learning is destabilized and higher-order derivatives is estimated)</u>

- Marginalize ρ_t with a Bayesian integral and just estimate θ
  - Simple and efficient (straightforward to implement and train)
  - Flexible (used in both classification and regression)
  - Robust (provide a measure of uncertainty)

Bayesian meta-learning for the few-shot setting via deep kernels

- Deep Kernel Transfer (DKT)
  - $\theta = \{\hat{\theta}, \hat{\phi}\}$ $where\ \ k\left(x, x' \middle| \hat{\theta}, \hat{\phi}\right) = k'(F_{\hat{\phi}}(x), F_{\hat{\phi}}(x') | \hat{\theta})$
  - $p\left(D^y \middle| D^x, \hat{\theta}, \hat{\phi}\right) = \prod_t p(T_t^y | T_t^x, \hat{\theta}, \hat{\phi}) = \prod_t \int \prod_k p\left(y_k \middle| x_k, \hat{\theta}, \hat{\phi}, \rho_t\right)\ d\rho_t$

Training

$$\log p\left(D^y \middle| D^x, \hat{\theta}, \hat{\phi}\right) = \sum_t \left[-\frac{1}{2} y_t^T K_t\left(\hat{\theta}, \hat{\phi}\right)^{-1} y_t - \frac{1}{2} \log\left| K_t\left(\hat{\theta}, \hat{\phi}\right) \right| + c \right]$$

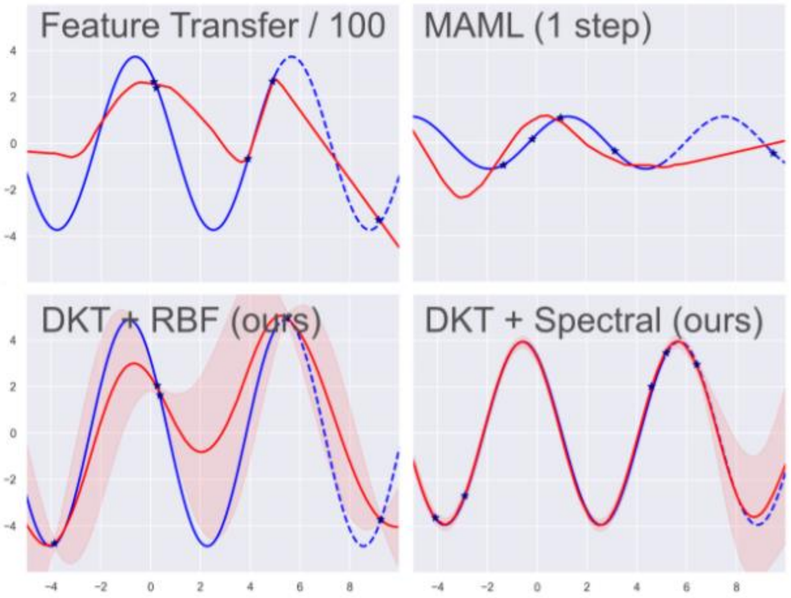
$$(\log p(y | x, \hat{\theta}, \hat{\phi}) = \sum_{c=1}^{C} \log p(y = c | x, \hat{\theta}, \hat{\phi})\ \ in\ \ case\ of\ classification)$$

Inference

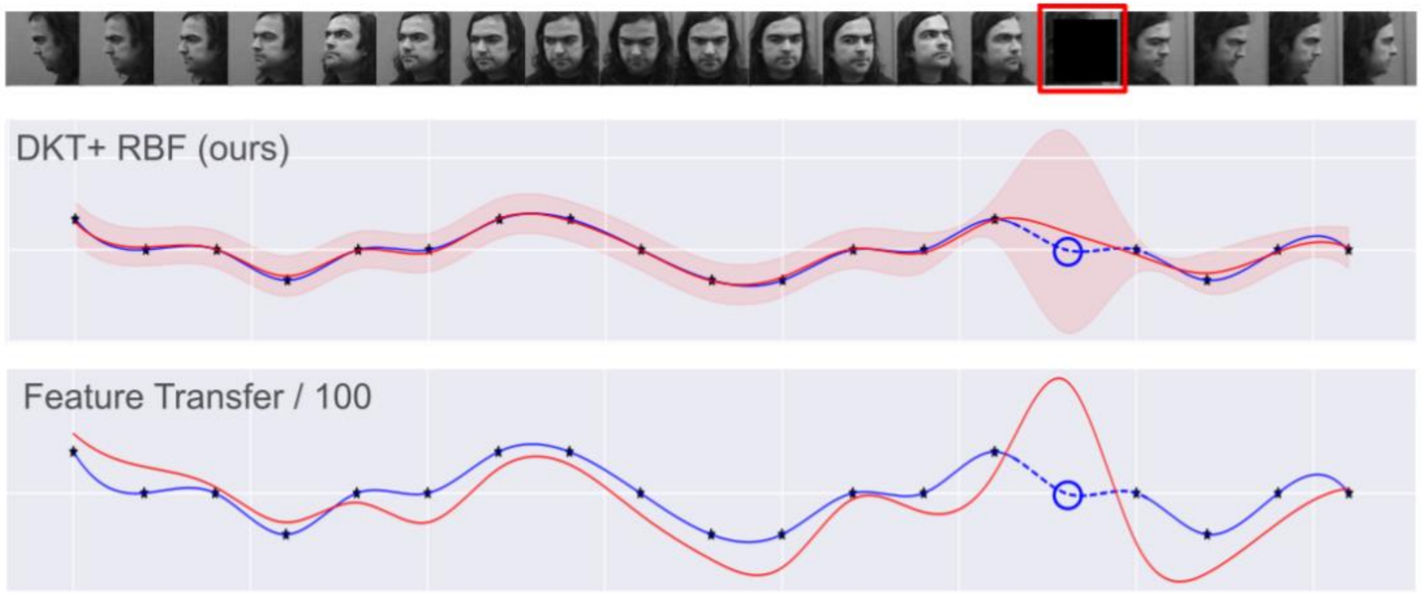
$$E[f_*] = k_*^T (K + \sigma^2 I)^{-1} y, \qquad cov(f_*) = k_{**} - k_*^T (K + \sigma^2 I)^{-1} k_*$$

$$(sigmoid\ activation\ for\ the\ predictive\ mean\ in\ case\ of\ classification)$$

Bayesian meta-learning for the few-shot setting via deep kernels



(a) Qualitative comparison



(b) Uncertainty estimation

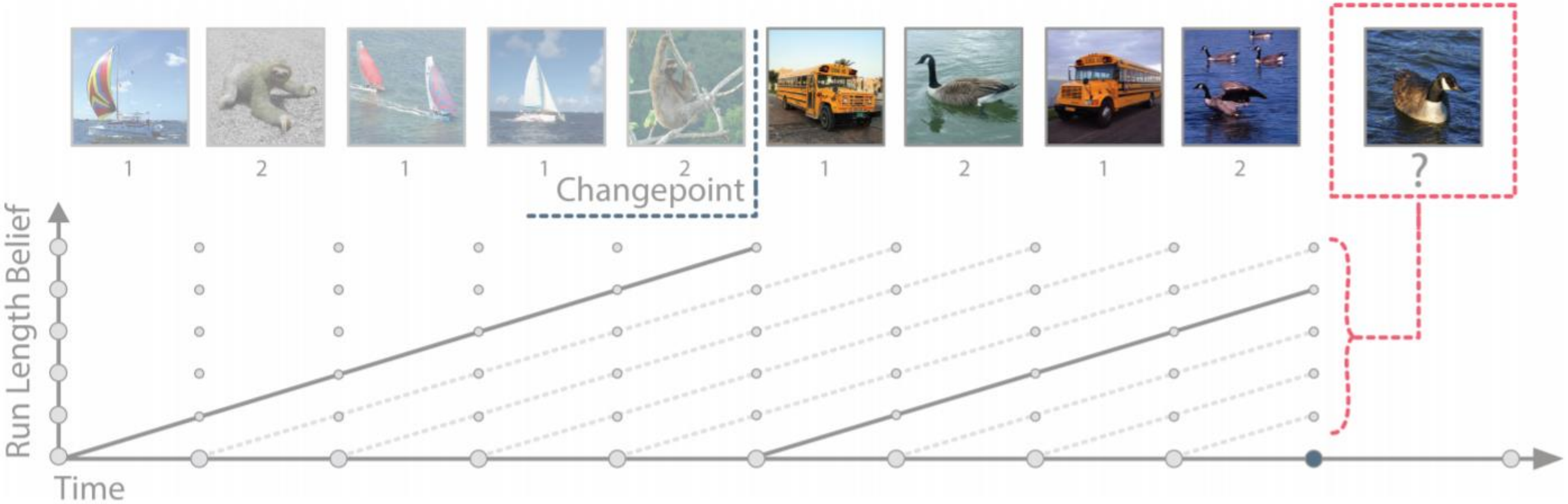
ALPaCA (Harrison et al., 2018)	0.14 ± 0.09	5.92 ± 0.11
Feature Transfer/1	2.94 ± 0.16	6.13 ± 0.76
Feature Transfer/100	2.67 ± 0.15	6.94 ± 0.97
MAML (1 step)	2.76 ± 0.06	8.45 ± 0.25
DKBaseline + RBF	2.85 ± 1.14	3.65 ± 1.63
DKBaseline + Spectral	2.08 ± 2.31	4.11 ± 1.92
DKT + RBF (ours)	1.38 ± 0.03	2.61 ± 0.16
DKT + Spectral (ours)	$\mathbf{0.08 \pm 0.06}$	$\mathbf{0.10 \pm 0.06}$

Feature Transfer/1	0.25 ± 0.04	0.20 ± 0.01
Feature Transfer/100	0.22 ± 0.03	0.18 ± 0.01
MAML (1 step)	0.21 ± 0.01	0.18 ± 0.02
DKT + RBF (ours)	0.12 ± 0.04	0.14 ± 0.03
DKT + Spectral (ours)	$\mathbf{0.10 \pm 0.01}$	$\mathbf{0.11 \pm 0.02}$

Continuous meta-learning without tasks

- Motivation
  - Many real world setting do not have known boundaries b/t tasks
  - A good learning system must detect the change in task

- Goal : enable use of meta-learning with unsegmented tasks



Continuous meta-learning without tasks

- Bayesian online change point detection (BOCPD)
  - Recursively estimate run length of current tasks r_t
  - $p(y_t|y_{1:t-1}) = \sum_{r_t} p(y_t|y_{1:t-1}, r_t)p(r_t|y_{1:t-1})$
  - $p(r_t|y_{1:t-1}) \propto p(r_t, y_{1:t-1})$
    $= \sum_{r_{t-1}} p(r_t|r_{t-1})p(y_{t-1}|y_{1:t-2}, r_{t-1})p(r_{t-1}, y_{1:t-2})$
  - $p(r_t|r_{t-1}) = \begin{cases} H(r_{t-1}+1) & if\ \ r_t = 0 \\ 1 - H(r_{t-1}+1) & if\ \ r_t = r_{t-1}+1 \\ 0 & otherwise \end{cases}$
  - $p(y_t|y_{1:t-1}, r_t) = \int p(y_t|\eta)p(\eta|y_{1:t-1}, r_t)\, d\eta$

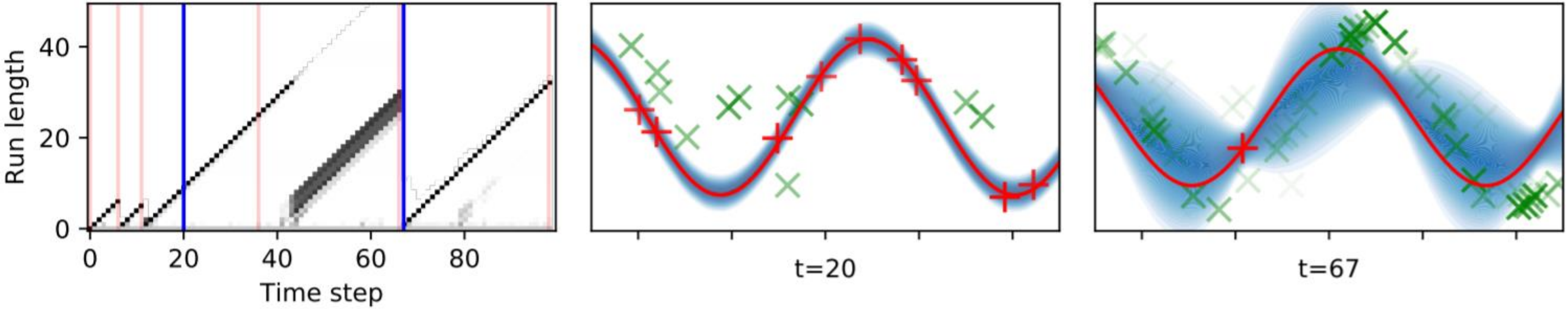
→ substitute into meta-learning algorithm

Adams, Ryan Prescott, and David JC MacKay. "Bayesian online changepoint detection." *arXiv preprint arXiv:0710.3742* (2007).

Continuous meta-learning without tasks

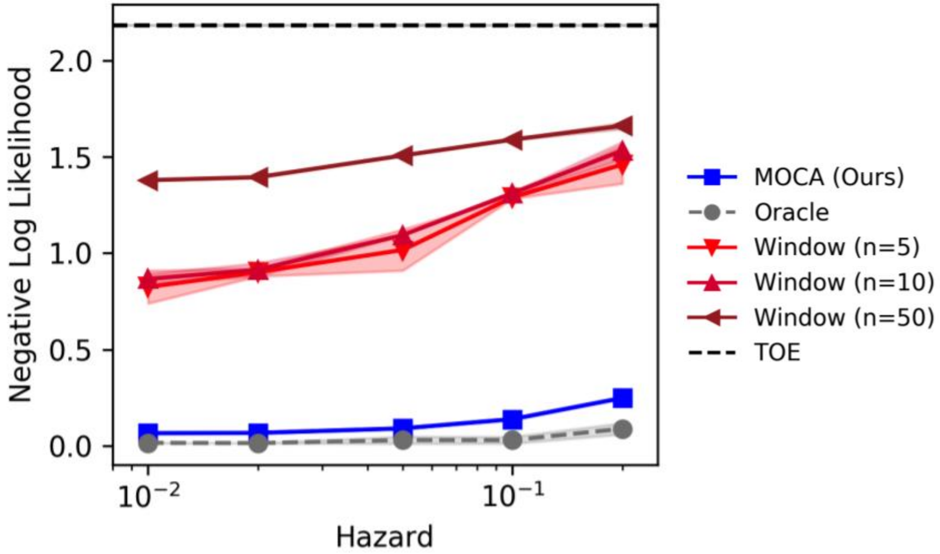
- Meta learning via online change point analysis (MOCA)
  1) η_t *updated* : $\eta_t[r] = h_\theta(x_t, y_t, \eta_{t-1}[r-1])$ *or* $\eta_t = f_\theta(x_{t-r_t+1:t}, y_{t-r_t+1:t})$
  2) y_t *estimation*: $p(y_t|x_{1:t}, y_{1:t-1}) = \sum_{r_t=0}^{t-1} p_\theta(y_t|x_t, \eta_{t-1}[r_t])\, p(r_t|x_{1:t}, y_{1:t-1})$
  3) y_t *observed* : $p(r_t|x_{1:t}, y_{1:t}) \propto p_\theta(y_t|x_t, \eta_{t-1}[r_t]) p(r_t|x_{1:t}, y_{1:t-1})$
  4) r_{t+1} *estimation* : $p(r_{t+1}|x_{1:t+1}, y_{1:t}) = \begin{cases} \lambda & \text{if } r_{t+1} = 0 \\ (1-\lambda)\, p(r_t|x_{1:t}, y_{1,t-1}) & \text{if } r_{t+1} = r_t + 1 \end{cases}$
  5) θ updated with loss $= \sum_{t=k}^{k+T} l_t = \sum_{t=k}^{k+T} -\log p_\theta(y_t|x_{1:t}, y_{1:t-1})$

Continuous meta-learning without tasks



Black : run length (higher intensity for higher probability)
Red vertical line : true change points
→ can capture the change points

Blue vertical line : time stamp of middle and right visualization
Red points : current task
Green points : previous tasks (more faded for older ones)
→ disregard the unrelated tasks
→ back to prior at change point



Thank you