TaskMet: Task-driven metric learning for end-to-end model learning

Dishank Bansal, Ricky T. Q. Chen, Mustafa Mukadam, Brandon Amos

Task-driven metric learning for end-to-end model learning

Dishank Bansal **Ricky T. Q. Chen** **Mustafa Mukadam** **Brandon Amos**
Meta AI

Abstract

We address the challenge of applying trained prediction models to solve downstream tasks. When prediction models are solely trained to achieve accurate predictions, they may struggle to perform well on the desired downstream tasks, due to limited capacity or lack of task-specific information. On the other hand, several studies have shown that directly incorporating task-specific information during the training of prediction models can lead to improved task performance. However, this approach disregards the information and semantics of the original prediction task, resulting in prediction models that excel only at the specific downstream task they were trained on but are highly inaccurate overall. To address this, we propose using task information to modify only the metric that the prediction model uses for training. This approach does not alter the optimal prediction model itself, but rather encourages the model, within its limited capacity, to generate more accurate predictions specifically for inputs that are relevant to the downstream task. This enables us to achieve the best of both worlds: a prediction model trained on its original prediction problem while also being valuable for the desired downstream task. We validate our approach through experiments conducted in two main settings: (1) decision-focused learning scenarios involving portfolio optimization and budget allocation, and (2) reinforcement learning in noisy environments with distracting states.

Meta AI

Paper available at – https://fb.workplace.com/groups/831302610278251/permalink/9253830228025405/

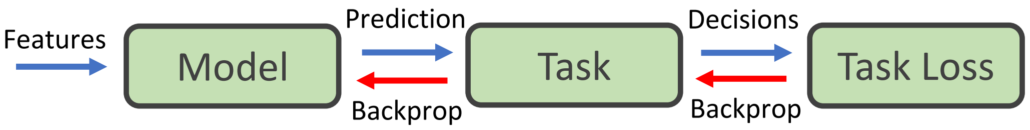
Introduction: Task-based Model Learning

No Free Lunch Theorem - **No single algorithm will solve all your machine learning problems better than every other algorithm**.

NFL in our context - **No single prediction model can be used for all downstream tasks.**

- Prediction models are used for downstream tasks.
- They are trained for maximum likelihood of data.
- In an ideal world, this will work. But we suffer from
  - Approximation error with modelling
  - Limited data
- Task-based Model Learning – Take into account the task information when training the prediction model.

Task-based Model Learning



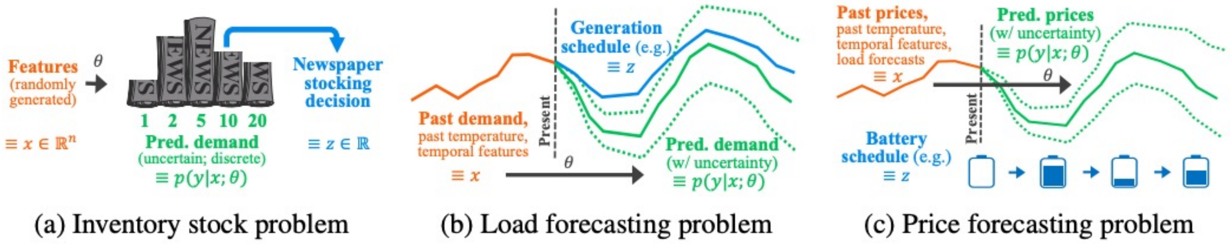
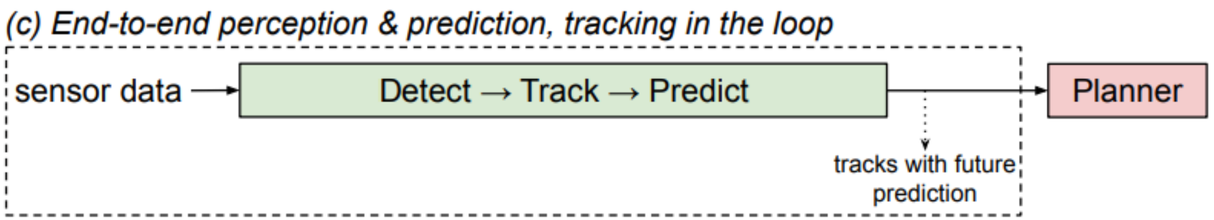
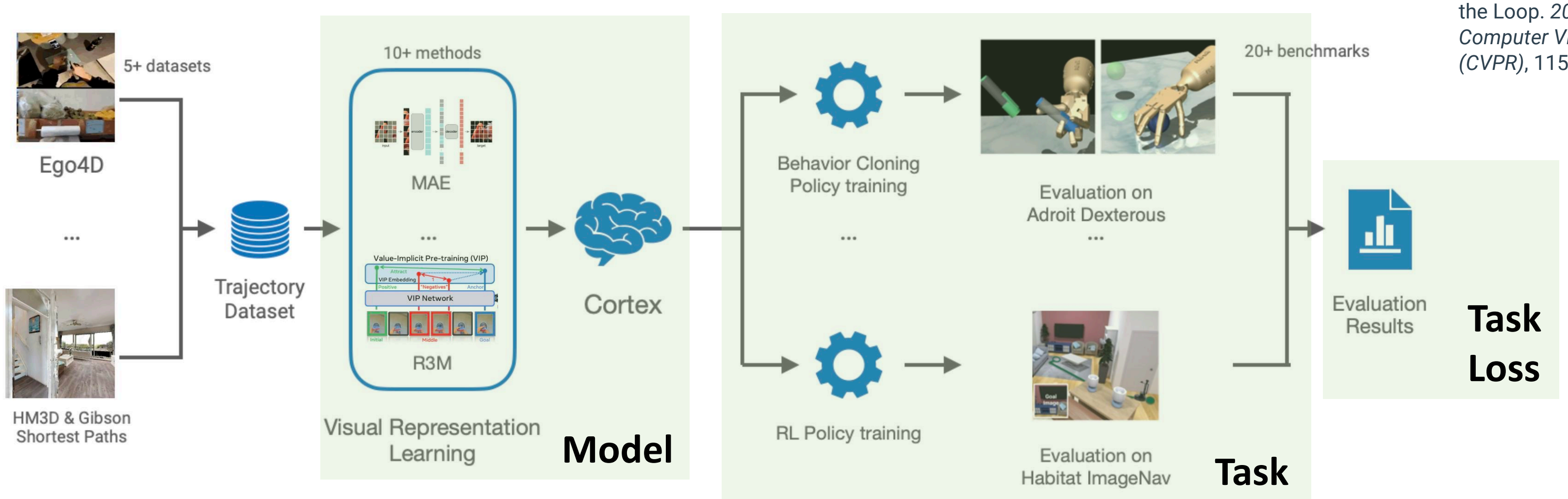


Figure 1: Features x, model predictions y, and policy z for the three experiments.

Donti, P., Amos, B., & Kolter, J. Z. (2017). Task-based end-to-end model learning in stochastic optimization. *Advances in neural information processing systems*, *30*.



Liang, M., Yang, B., Zeng, W., Chen, Y., Hu, R., Casas, S., & Urtasun, R. (2020). PnPNet: End-to-End Perception and Prediction With Tracking in the Loop. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 11550-11559.



Majumdar, A., Yadav, K., Arnaud, S., Ma, Y. J., Chen, C., Silwal, S., ... & Meier, F. (2023). Where are we in the search for an Artificial Visual Cortex for Embodied Intelligence?. *arXiv preprint arXiv:2303.18240*.

Background

- Prediction model trained for Maximum Liklihood Estimation

$$\theta^\star := \underset{\theta}{\operatorname{argmin}}\, \mathcal{L}_{pred}(\theta)$$
$$\mathcal{L}_{pred}(\theta) = \mathrm{E}_{x,y\,\sim\,D}[\|f_\theta(x) - y\|_2^2]$$

- Task-based prediction models generally trained as multi-objective problem as follows

$$\theta^\star := \underset{\theta}{\operatorname{argmin}}\, \mathcal{L}_{task}(\theta) + \alpha\mathcal{L}_{pred}\,(\theta)$$

- Problems with training as multi-objective problem
    - Overfitting to the particular task at hand, rendering the model unable to generalize to other tasks
    - Parameter α has to be tuned for every different task.
    - Task-relevant features being learned in not interpretable

TaskMet – Task-Driven Metric Learning for Model Learning

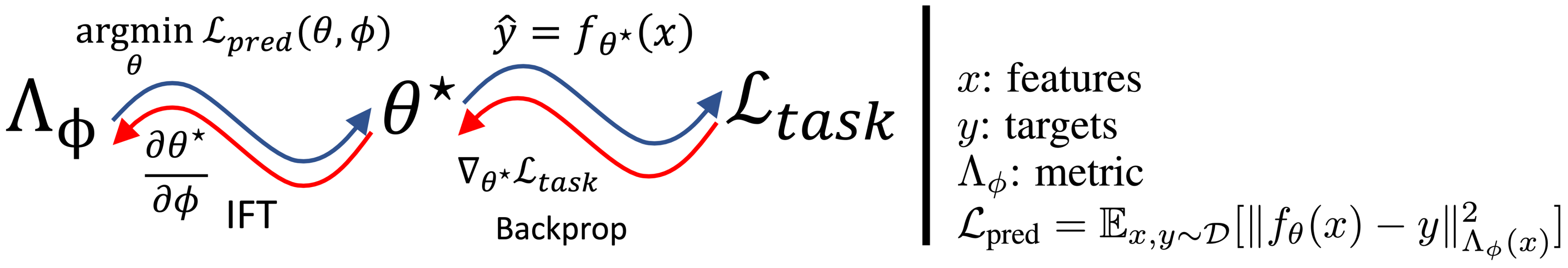


Figure 1: TaskMet: learning a metric for prediction with a downstream task loss.

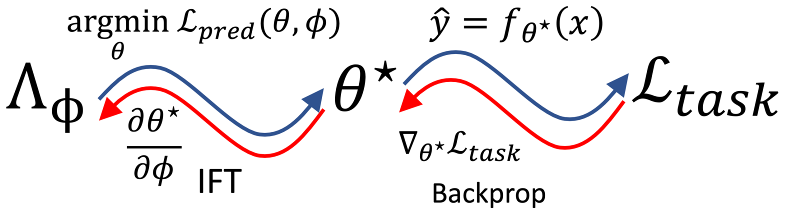
TaskMet

- Using metricized mahalanobis loss for training prediction model

$$\mathcal{L}_{pred}(\theta,\phi) \coloneqq \mathrm{E}_{x,y\sim D}\left[\|f_\theta(x)-y\|^2_{\Lambda_\phi(x)}\right] = \mathrm{E}_{x,y\sim D}\left[(f_\theta(x)-y)^T\Lambda_\phi(x)(f_\theta(x)-y)\right]$$

- Metric $\Lambda_\phi(x)$, which a PSD matrix of dim n, can capture following properties of data:
  - **Relative importance of dimensions**: the metric allows for down- or up-weighting different dimensions of the prediction space.
  - **Correlation in the prediction space**: Off diagonal elements of the matrix can capture correlation in features.
  - **Relative importance of samples**: heteroscedastic metrics Λ(x) enable different samples to be weighted differently inthe final expected cost over the dataset

TaskMet



- End-to-end metric learning for model learning

$$\phi^\star \coloneqq \operatorname{argmin} \mathcal{L}_{task}(\theta^\star(\phi))$$

$$\text{s.t.} \quad \theta^\star(\phi) = \operatorname{argmin} \mathcal{L}_{pred}(\theta, \phi)$$

- Have to calculate $\nabla_\phi \mathcal{L}(\theta^\star(\phi))$ for being able to learn ϕ using gradient descent.

$$\nabla_\phi \mathcal{L}_{task}\big(\theta^\star(\phi)\big) = \nabla_\theta \mathcal{L}_{task}(\theta)\Big|_{\theta=\theta^\star(\phi)} \cdot \underbrace{\frac{\partial\, \theta^\star(\phi)}{\partial \phi}}_{\text{Calculate using Implicit Function theorem}}$$

$$\nabla_\phi \mathcal{L}_{task}\big(\theta^\star(\phi)\big) = -\nabla_\theta \mathcal{L}_{task}(\theta) \cdot \left(\frac{\partial \mathcal{L}_{pred}(\theta, \phi)}{\partial^2\, \theta}\right)^{-1} \cdot \frac{\partial \mathcal{L}_{pred}(\theta, \phi)}{\partial \phi\, \partial \theta}\Bigg|_{\theta=\theta^\star(\phi)}$$

Experiments

Decision Oriented Model Learning

Resource Prediction -> Model, Resource Allocation -> Task.

- GT – Cubic dataset, Prediction Model – Linear

$$y_n = 10x_n^3 - 6.5x_n$$

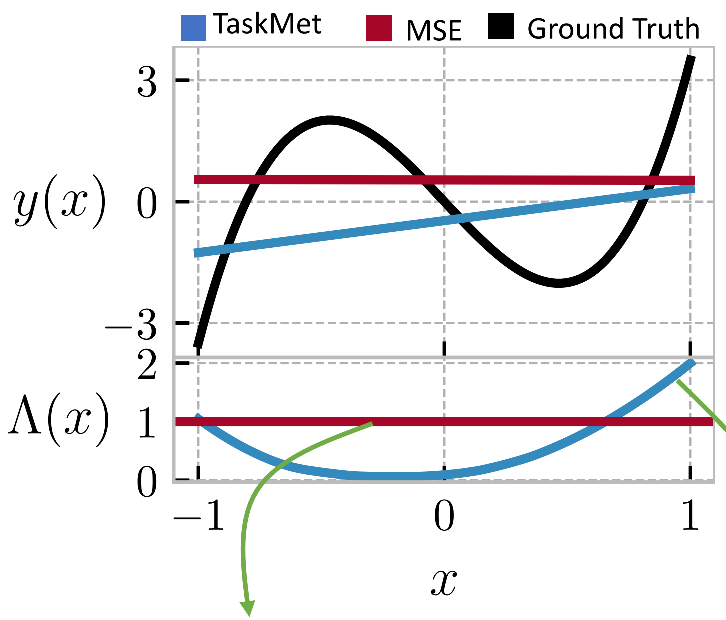
- Multiple resource utility y_n predictions.

$$\hat{y} = [y_1, \dots, y_n]$$

- Task – choose resource with maximum utility, i.e

$$\boldsymbol{z}^*(\hat{\boldsymbol{y}}) = \arg\operatorname{topk}(\hat{\boldsymbol{y}})$$

- Since there is severe modelling error, model has to choose where to make errors. High Utility points are more important for the task, hence important to model them correctly



MSE puts equal weight on all points in total loss

TaskMet prioritize accurate prediction of high utility points.

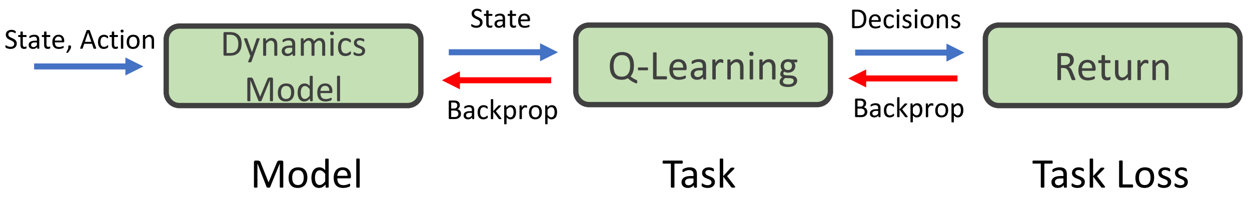
Table 1. Comparison of the normalized test decision quality (0=random, 1=oracle) on the decision oriented learning problems. α is prediction loss weight in Eq. (1)

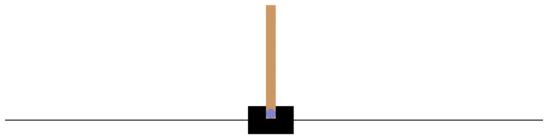
		Problems		
Method	α	Cubic	Budget	Portfolio
MSE		-0.96 ± 0.02	0.54 ± 0.17	0.33 ± 0.03
DFL	0	0.61 ± 0.74	0.91 ± 0.06	0.25 ± 0.02
DFL	10	0.62 ± 0.74	0.81 ± 0.11	0.34 ± 0.03
LODL	0	0.96 ± 0.005	0.84 ± 0.105	0.17 ± 0.05
LODL	10	-0.95 ± 0.005	0.58 ± 0.14	0.30 ± 0.03
TaskMet		0.96 ± 0.005	0.83 ± 0.12	0.33 ± 0.03

Shah, S., Wang, K., Wilder, B., Perrault, A., & Tambe, M. (2022). Decision-Focused Learning without Differentiable Optimization: Learning Locally Optimized Decision Losses.

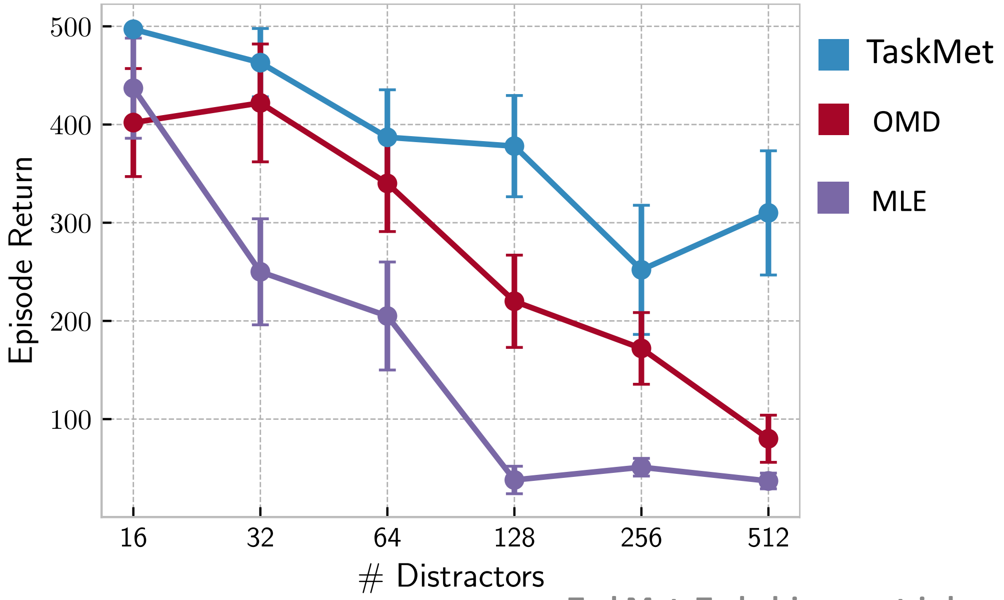
Experiments

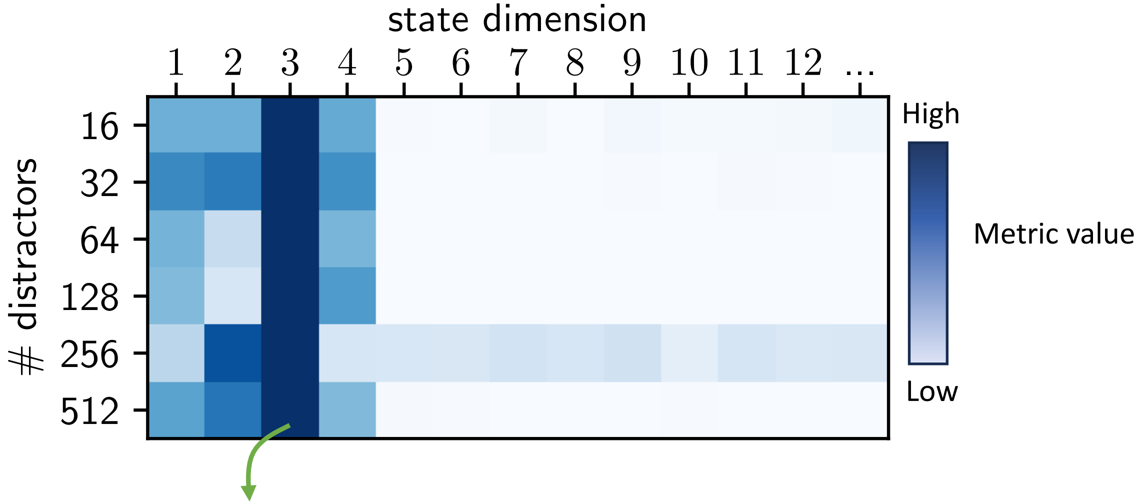
Model Based Reinforcement Learning





- Experiment on cartpole environment. 4 dimensional State space
- Added noisy/redundant dimensions to the state of the agent.
- We use diagonal Matrix as metric here. Metric helps with finding the dimension in the state space that are relevant for the task.



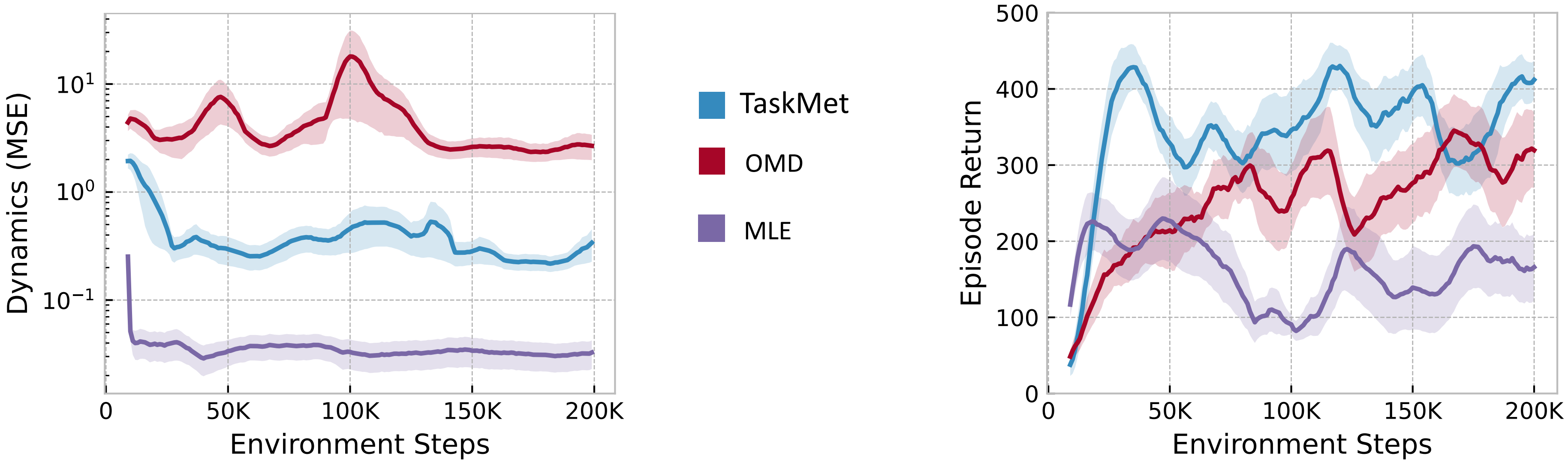


Metric value is highest for dimension 3 – pole angle – most indicative of the reward

Experiments

Model Based Reinforcement Learning - Continued

Limited model Capacity - Dynamics model is underparameterized, i.e have very limited capacity



Summary

- Prediction models needs to be tailored to downstream task they will be used for.
- Train the model using Mahalanobis loss parameterized by metric $\Lambda_{\phi}(x)$
- Learn the metric $\Lambda_{\phi}(x)$ using gradients from $\mathcal{L}_{task}$.
    - Will need to use implicit function theorem to calculate these gradients.
- Learned metric can provide information about important samples and dimensions.

Paper available at – https://fb.workplace.com/groups/831302610278251/permalink/9253830228025405/

Meta AI