Continual Slow-and-Fast Adaptation of Latent Neural Dynamics (CoSFan): Meta-Learning What-How & When to Adapt

Ryan Missel & Linwei Wang {rxm7244,Linwei.Wang}@rit.edu
GCCIS, Rochester Institute of Technology

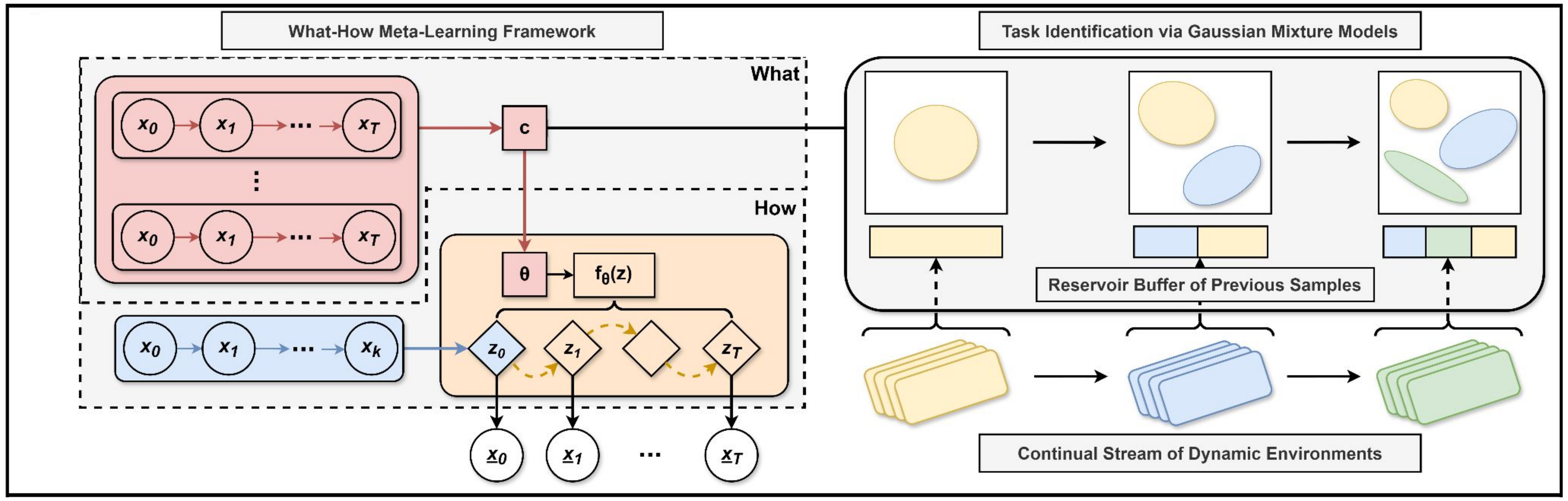
Paper Code

Overview

- Forecasting from high-dimensional time-series requires adapting to systems with varying underlying dynamics. Standard training on non-stationary systems risks catastrophic forgetting when dynamics shift over time.
- Present **CoSFan**, a continual meta-learning framework that enables both slow-and-fast adaptation of latent dynamics functions to few-shot samples.

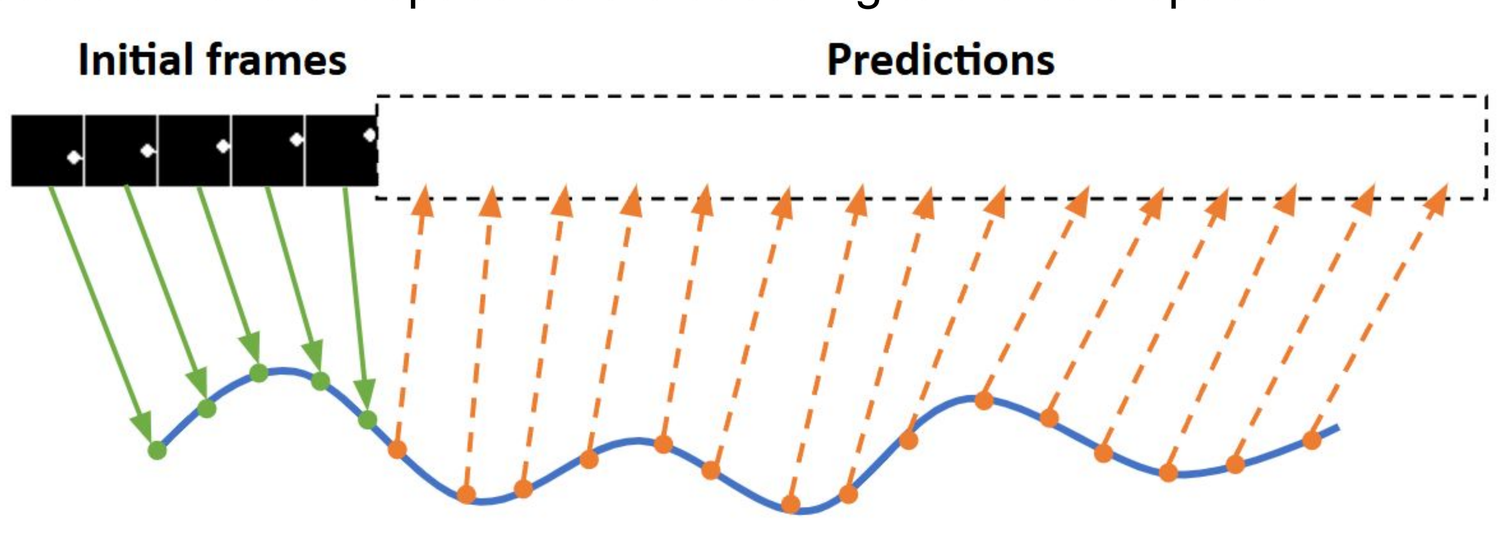
Novel framework:

- A feed-forward hyper-network meta-model that infers *what* system is observed and *how* to adapt the latent dynamics.
- A continual learning strategy that detects *when* task shifts occur and identifies the relation to prior tasks via task-relational reservoir sampling.

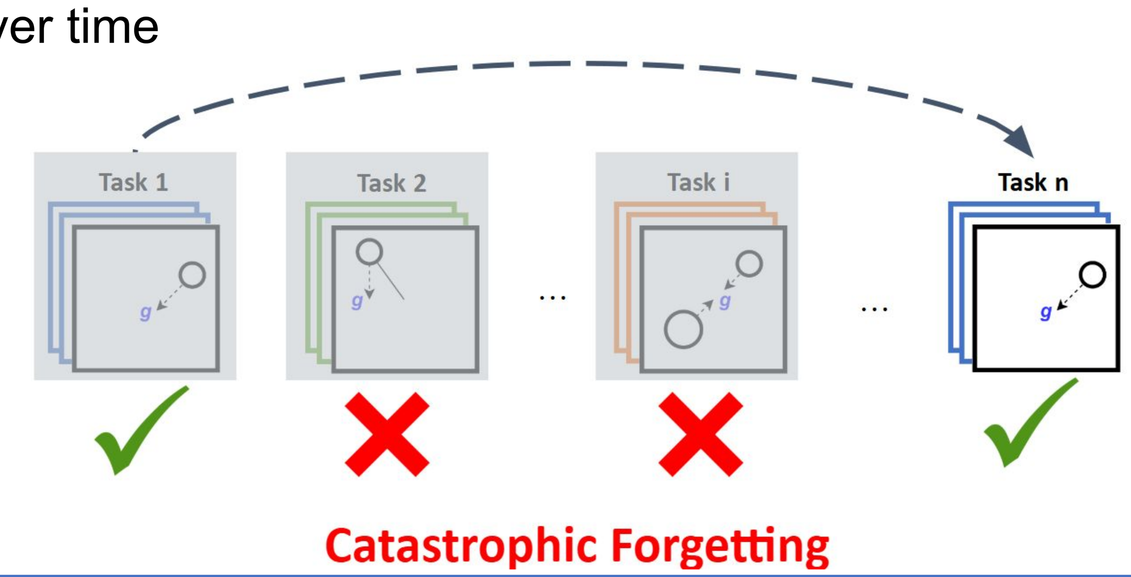


Problem Setting

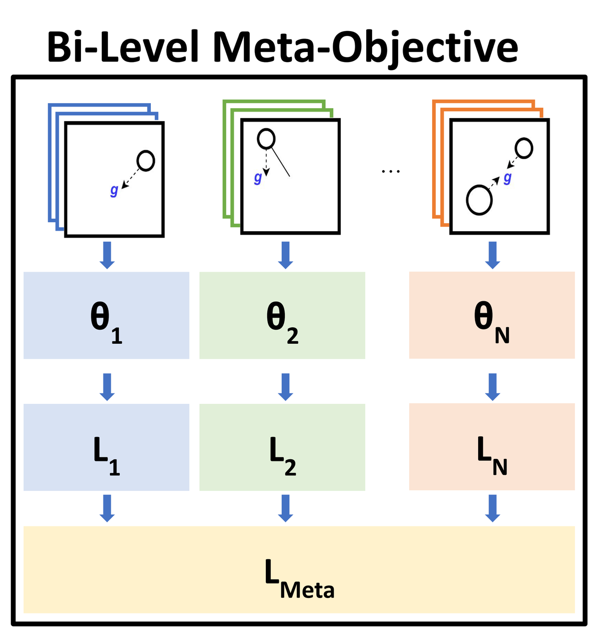
- Given a few initial frames, leverage a Sequential Latent Variable Model (sLVM) to forecast it in a latent space before decoding to the data space.



- Dynamical systems stream in over time
- Task **boundaries** are unknown
- Task **labels** are unknown
- Old tasks can re-emerge later
- Assume **local stationarity,** a minimum amount of time tasks stay the same

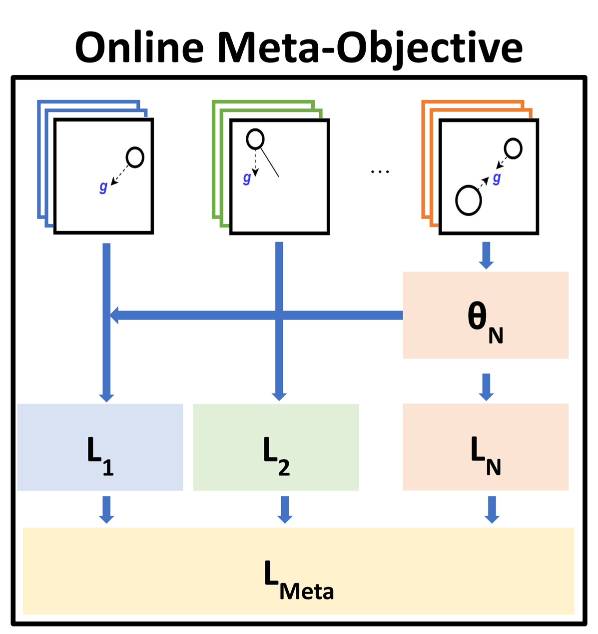


Limitation of CML



Without labels for context-query pairing, we cannot use standard meta-optimization.

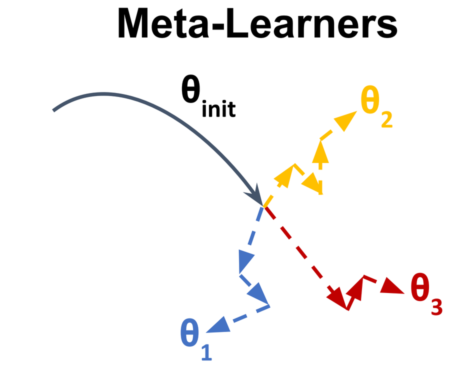
We must use online objective approximations, which are insufficient in broader domains.



Algorithmic Priors

Existing Work

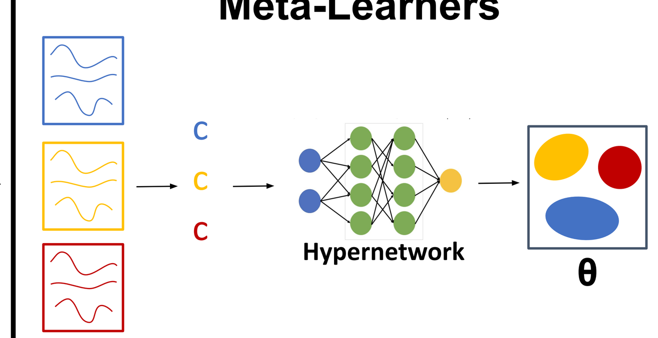
Gradient-Based Meta-Learners



- Shared initial parameter set
- Not suited for all domains
- Slow to adapt

Proposed Work

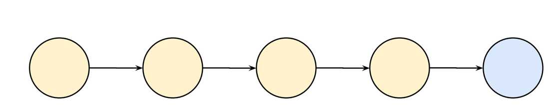
Feed-Forward Meta-Learners



- Complex non-linear mappings
- Fast & parallelizable
- General algorithm

Task ID & Relational-Modeling

Detecting Boundaries

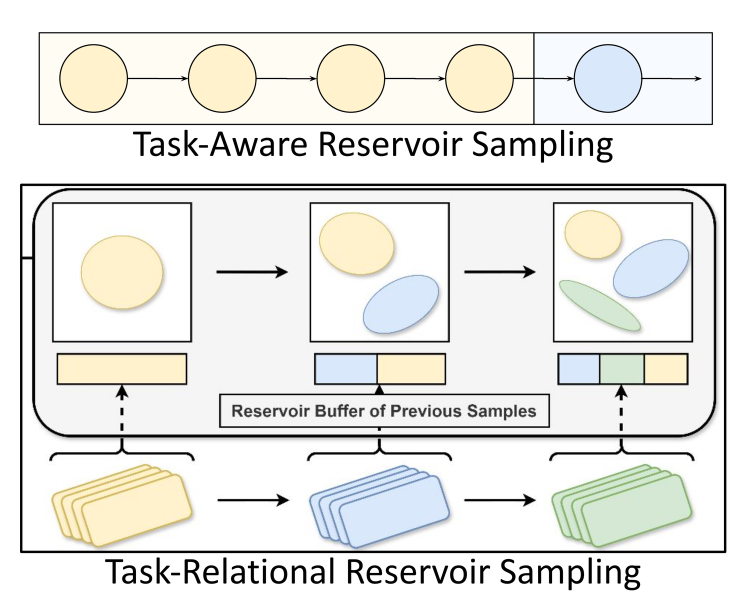


$$||\mathcal{L}_{j,n} - \mathcal{L}_{j,n-1}|| > \nu$$

(Caccia, NeurIPS 2020)

- Use previous timesteps as context
- Performance dip when task swaps
- Threshold ν on per-step difference

Task Identification



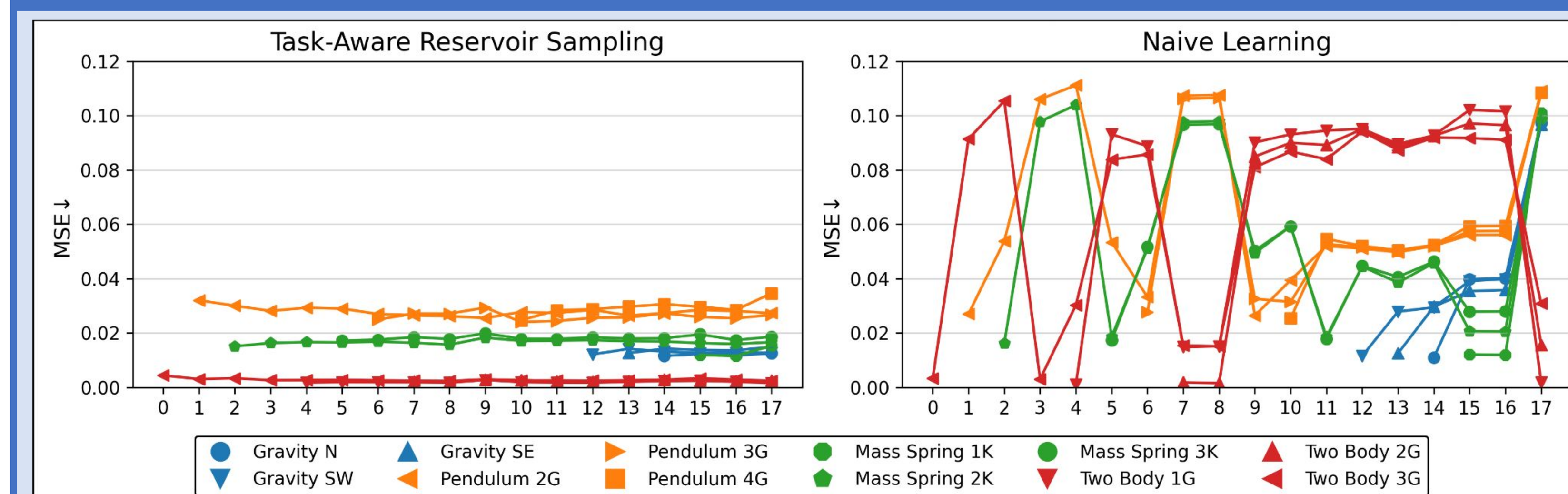
Adaptation Efficiency

Table 1: CML adaptation efficiency comparison. All methods were adapted on 1200 batches. MAML-X refers to the X number of inner-steps.

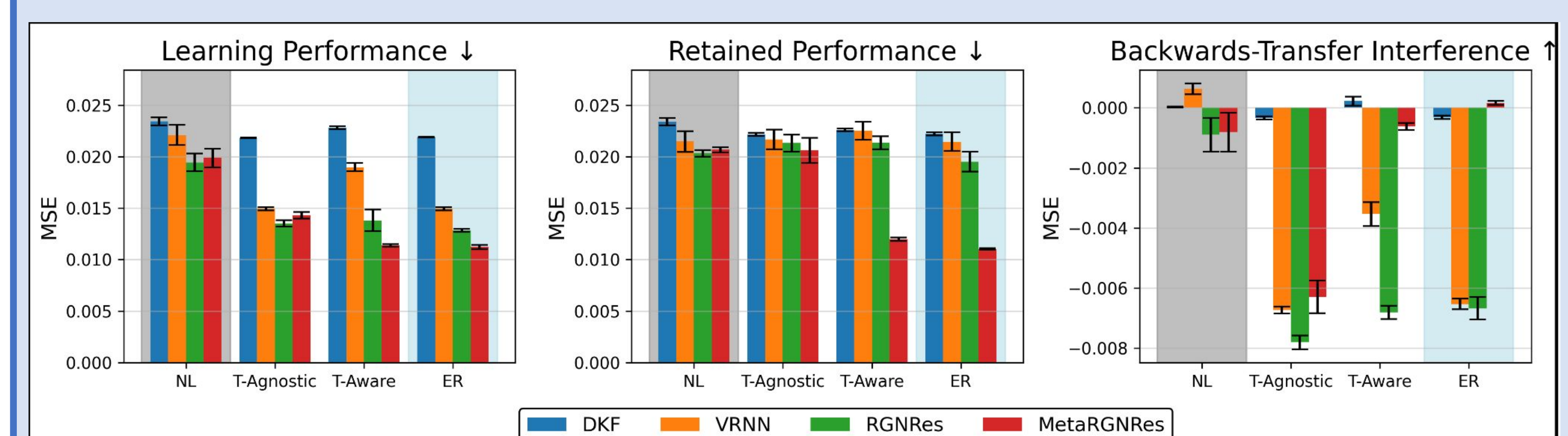
Model	Metric	T-Agnostic	T-Aware
MAML-1	TTA-1 [s]	*0.0148(0.0009)*	*0.0150(0.0080)*
	TTA-12 [s]	*0.0149(0.0069)*	*0.1602(0.0058)*
	TTT [min]	*38.8(0.2)*	*116.3(11.0)*
MAML-5	TTA-1 [s]	0.0618(0.0077)	0.0670(0.0096)
	TTA-12 [s]	0.0636(0.0074)	0.7475(0.0413)
	TTT [min]	91.5(0.7)	302.8(22.7)
FF	TTA-1 [s]	**0.0018(0.0048)**	**0.0017(0.0043)**
	TTA-12 [s]	**0.0018(0.0043)**	**0.0017(0.0005)**
	TTT [min]	**30.0(0.3)**	**85.2(5.5)**

- Feed-Forward adaptation is magnitudes faster than gradient-based.
- Gradient-based scales poorly to the number of simultaneous tasks while Feed-Forward is agnostic in speed.

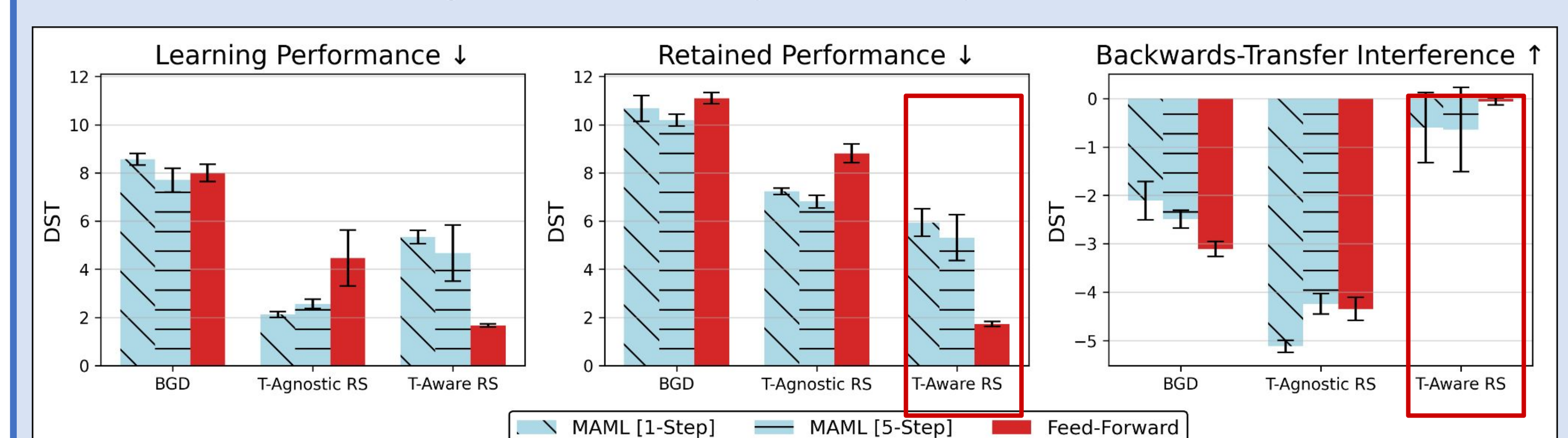
Experiments



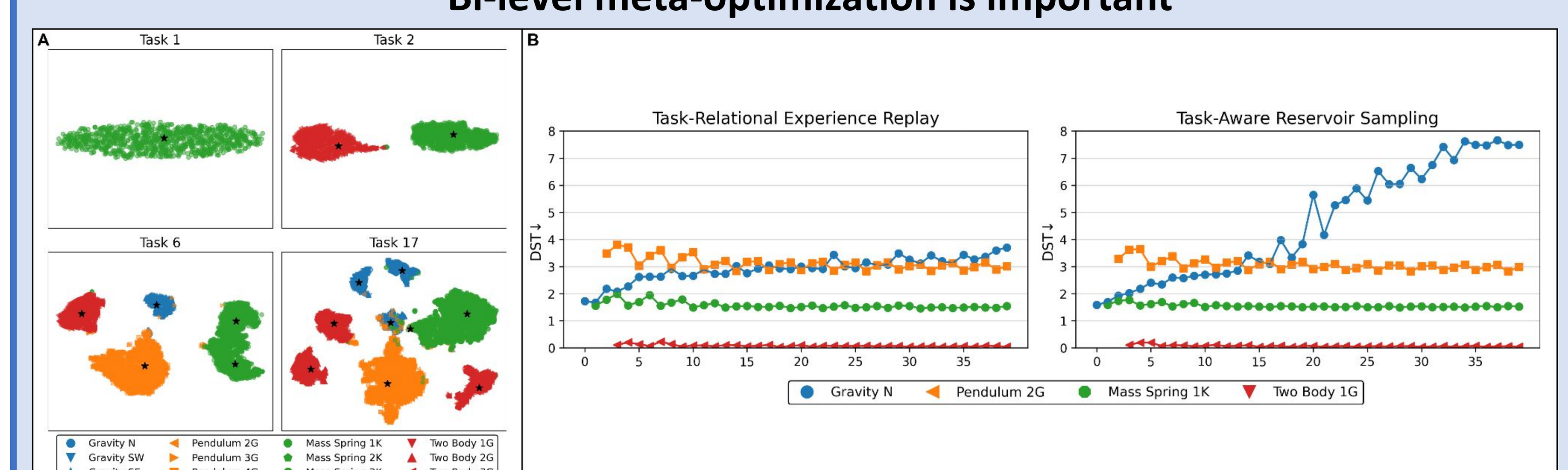
Naive meta-models catastrophically forget



Heterogeneous latent dynamics require meta-models



Bi-level meta-optimization is important



GMM task identification is robust to re-appearing tasks