Lifelong Learning Using Eigentasks

Aswin Raghavan, Jesse Hostetler, Indranil Sur, Abrar Rahman
SRI International

ELECTRONICS RESURGENCE INITIATIVE SUMMIT & MTO Symposium

SRI International

DARPA Lifelong Learning Machines (L2M)

Artificial Intelligence

Background

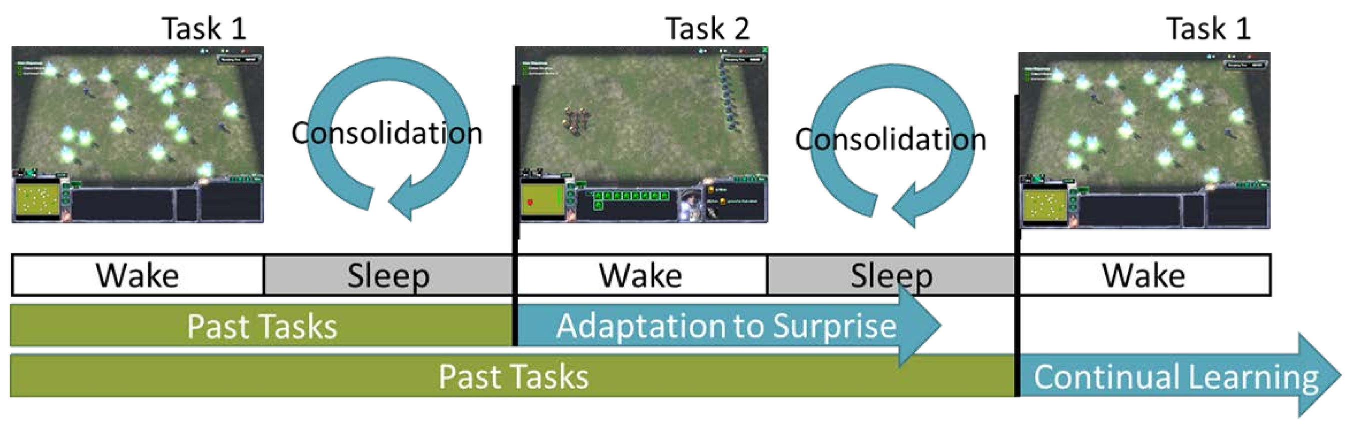
As part of the DARPA Lifelong Learning Machines (L2M) program, SRI International has developed the Eigentask framework. Eigentasks combine the strength of **generative memory** (i.e. using a machine learning model capable of generating or “re-playing” samples of previous tasks), with the strength of **mixture-of-experts** models (i.e. partitioning tasks along the skills required to solve them). Lifelong Learning has three core components [1]:

1. **Continual Learning**: The system learns a nonstationary stream of tasks (both novel and recurring) without distinct training and testing phases.
2. **Transfer and Adaptation**: As learning progresses, the system performs better on average on the next task, for both novel tasks (**forward transfer**) and known tasks (**backward transfer**).
3. **Sustainability**: The system continues learning for an arbitrarily long lifetime using limited resources (e.g., memory, time) in a scalable way.

Our approach can achieve all three objectives in both supervised learning and reinforcement learning settings.

L2 Problem Settings

Lifelong RL in Starcraft 2: The agent experiences a sequence of “mini-games” with different objectives, where old tasks can recur along with new tasks. The agent must learn to solve new tasks, using knowledge of old tasks to speed it up (forward transfer) while remembering how to solve old tasks (backward transfer).

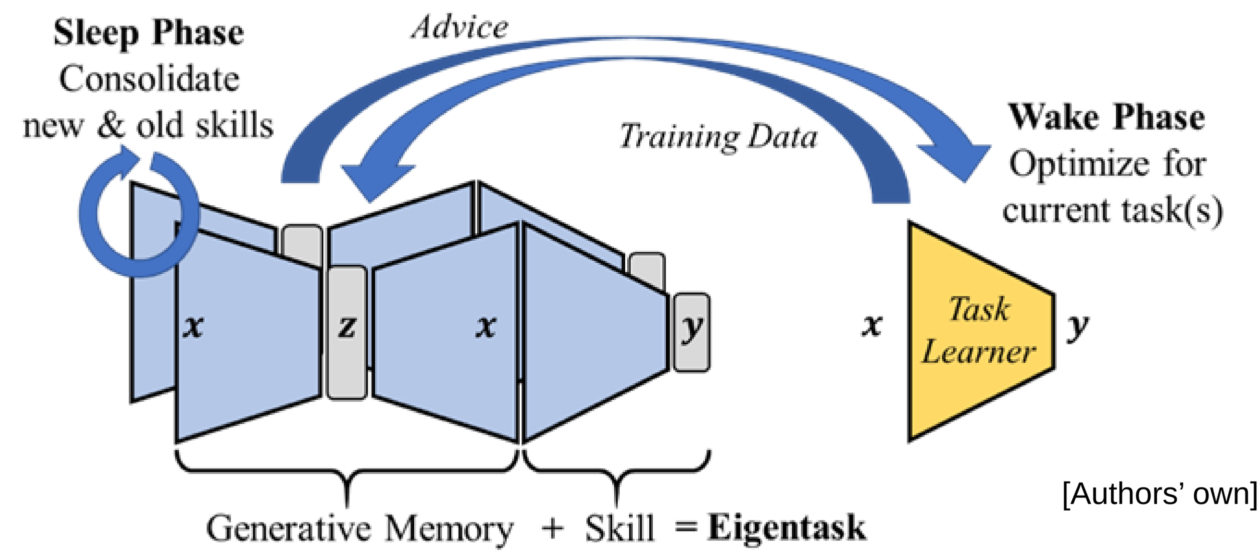


[Authors’ own]

Lifelong Supervised Learning: The agent learns to classify images (e.g. MNIST digits) where data is presented for only a few classes at a time. The agent must learn to classify new classes while remembering old classes.

Approach

Eigentask framework:



[Authors’ own]

An **eigentask** is a machine learning model that 1) can solve a set of tasks, and 2) can sample from its input distribution. 3) can determine when the eigentask is applicable. See our paper [3] for all technical details.

Eigentask: $E = (g, f, \tau)$ comprising a **generator** probability distribution $g(\epsilon)$ on the inputs X, and a **skill** $f: X \mapsto \Delta(Y)$ that maps inputs to outputs. $\tau: X \mapsto [0,1]$ measures the applicability of the eigentask for a given input.

Wake-Sleep Phases:

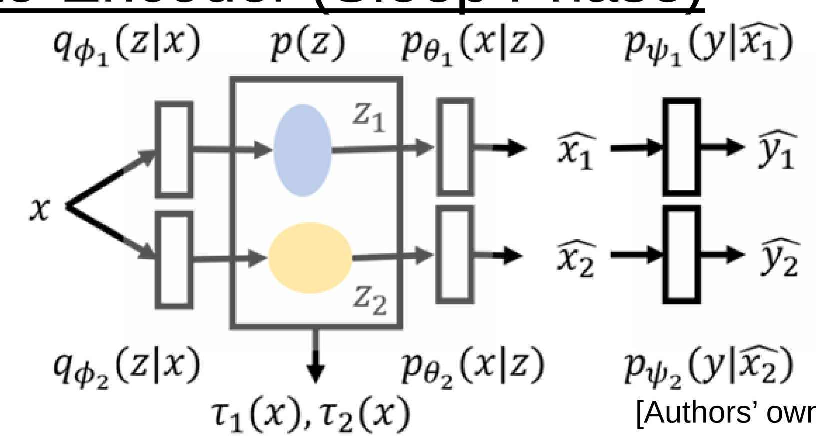
The eigentasks are trained on a data stream in alternating phases:

Wake Phase: A separate task learner learns the current task or tasks, using forward transfer from previously-learned eigentasks.

Sleep Phase: Knowledge learned from recent tasks is consolidated into eigentasks using generative replay to avoid forgetting.

Open-World Variational Auto-Encoder (Sleep Phase)

We realize the eigentask framework with **Open-World Variational Auto-Encoder (OWVAE)**. Each generator is a VAE [2]. The skills are classifiers or RL policies. The training objective is:



[Authors’ own]

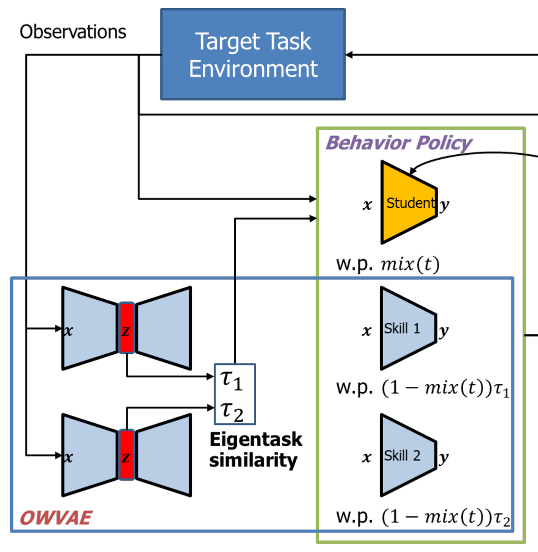
$$\min_{\theta,\phi,\psi} E_{x,y}[E_\tau[L_{\text{VAE}}(x;\theta,\phi) + \log p_\psi(y|\hat{x})],$$

which combines losses for the generators and skills. The losses from each eigentask are weighted by a likelihood ratio:

$$\tau_i(x) = \text{softmax}_\beta \frac{p_{\theta_i}(x|z)}{\max_j p_{\theta_j}(x|z)} \approx \text{softmax}_\beta \frac{\Phi(z_i)}{\max_j \Phi(z_j)}$$

The OWVAE simultaneously learns to partition a data stream of tasks into eigentasks via τ and to solve each eigentask.

Rejection sampling: reject samples based on confidence of skill (VAug), create balanced dataset (Baug), or both (VBAug).



Lifelong RL with Eigentasks:

Goal: forward transfer: wake phase
Idea: explore new task using relevant skills identified by OWVAE
Reduces sample complexity if some previous task is like target task.

Importance weighting for off-policy correction

For each eigentask i, $b(s) = \begin{cases} \pi(s), & 1-\mu(t) \\ p_{\psi_i}(s), & w.p.\ \mu(t)\tau(s)_i \end{cases}$

Results and Impact

Continual Learning for Supervised Classification:

We tested our method on the “SplitMNIST” benchmark and a new benchmark “SplitMNIST+FashionMNIST” in the “Class Incremental” setting:

- Examples presented from two classes at a time
- The agent learns to label all the classes

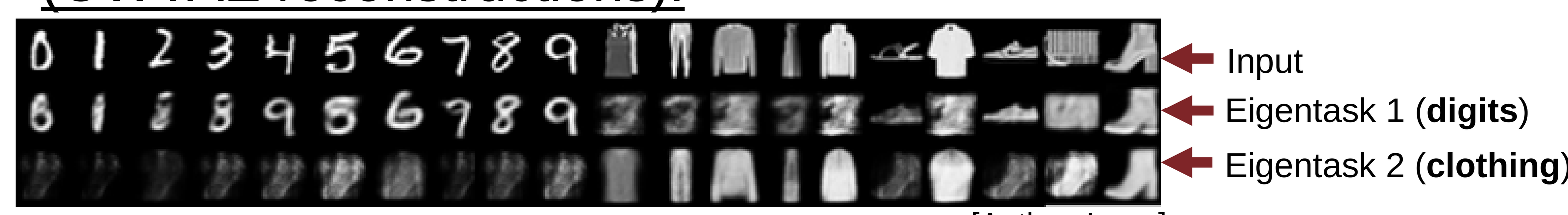
Approach	Method	D1	D2
Baselines	None - lower bound	19.90	10.22
	Offline - upper bound	97.94	90.89
Regularization	EWC	20.01	10.00
	Online EWC	19.96	10.00
	SI	19.99	10.00
Replay	LwF	23.85	10.07
	DGR	90.79	73.36
	DGR x2	91.83	65.82
	DGR+distill	91.79	72.40
	DGR+distill x2	94.01	67.37
	RtF	92.56	61.15
	RtF x2	92.86	61.41
Replay+Exemplars	iCaRL	94.57	82.69
Replay+Eigentask	ET1-BaseAug	87.68	69.29
	ET1-BAug	90.99	74.11
	ET1-VAug	87.33	63.34
	ET1-VBAug	90.69	77.43
	ET2-BaseAug	88.93	57.91
	ET2-BAug	**91.27**	69.95
	ET2-VAug	82.08	69.55
	ET2-VBAug	90.25	**76.81**

[Authors’ own]

- SplitMNIST (D1): our method is competitive with state-of-the-art
- SplitMNIST+FashionMNIST(D2): our method is superior than current approaches because:

1. Able to deal with multiple modes in the input distribution
2. Successfully separates MNIST and Fashion MNIST as eigentasks, learns one skill each

Automatic partitioning of data stream into coherent eigentasks (OWVAE reconstructions):



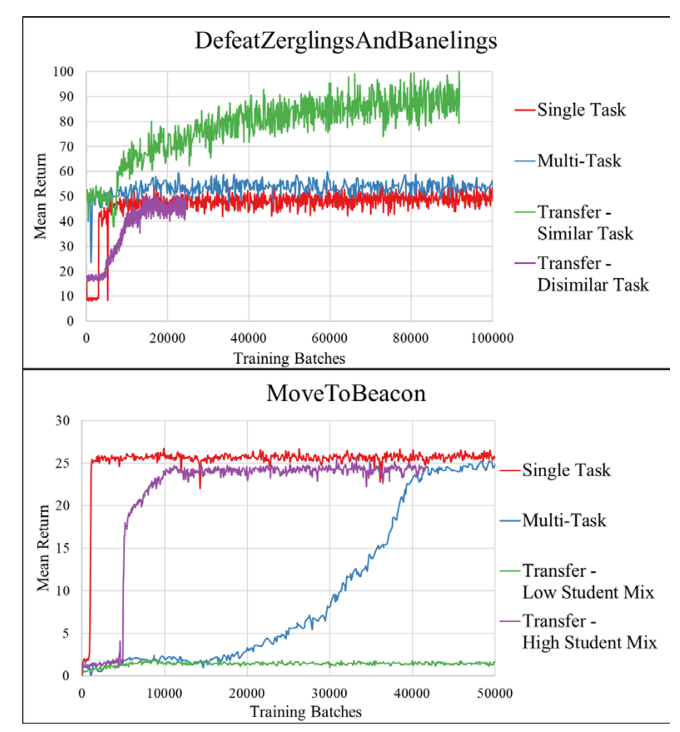
[Authors’ own]

Forward Transfer in Reinforcement Learning:

We train a model with two eigentasks and measure forward transfer to a novel task using the eigentasks as “teachers”

When transferring to a similar task (Top), transfer gives superior few-shot performance and convergence to a better policy than single-task learning.

When transferring to a dissimilar task (Bottom), transfer can slow learning, but we can modulate the “strength” of transfer to mitigate this effect.



[Authors’ own]

Impact

We present a unified approach to lifelong learning that is applicable to supervised, unsupervised, and reinforcement learning settings. The method supports both forward and backward transfer and discovers related tasks without knowledge of task identity. Many approaches from the literature can be incorporated into the eigentask framework.

References
[1] L2M Working Group on Definitions and Scenarios (2020). Technical Report.
[2] Kingma, D. P., & Welling, M. (2013). Auto-encoding variational Bayes. *arXiv:1312.6114*.
[3] Raghavan, A., Hostetler, J., Sur, I., Rahman, A., & Divakaran, A. (2020). Lifelong learning using eigentasks: Task separation, skill acquisition, and selective transfer. *ICML 2020 Workshop on Lifelong Learning.* https://www.youtube.com/watch?v=IsO2Yz4z43Q

This research was developed with funding from the Defense Advanced Research Projects Agency (DARPA).