Efficient Reinforcement Learning by Discovering Neural Pathways

NeurIPS 2024

Samin Yeasar Arnob [1,2], Riyasat Ohib [3], Sergey Plis [4], Amy Zhang [5], Alessandro Sordoni [6], Doina Precup [1,2]

McGill University [1], Mila Quebec AI Institute [2],
Georgia Institute of Technology [3], Georgia State University [4],
University of Texas, Austin [5], Microsoft Research [6]

Motivation:

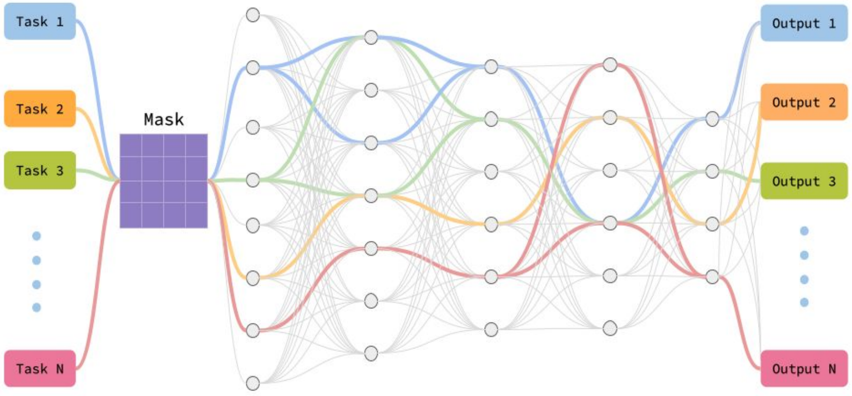
- The human brain:
    - ***continuously learns*** new things without catastrophic forgetting due to its ***plasticity*** [1, 2, 3, 4]
    - ***strengthens*** more frequently used synaptic connections and eliminates synaptic connections that are rarely used, a phenomenon called ***synaptic pruning*** [5]
    - ***creates neural pathways to transmit information***; different neural pathways [6, 7] are used to complete different tasks.

- We propose a **novel approach** in deep reinforcement learning to form **distinct neural pathways for different tasks** within one neural network.

[1] Karl Zilles. **Neuronal plasticity as an adaptive property of the central nervous system**. Annals of Anatomy-Anatomischer Anzeiger, 174(5):383–391, 1992.
[2] Daniel Drubach. **The brain explained**. Pearson, 2000.
[3] Jill Sakai. **Core concept: How synaptic pruning shapes neural wiring during development and, possibly, in disease**. Proceedings of the National Academy of Sciences, 117(28):16096–16099, 2020. ISSN 0027-8424. doi: 10.1073/pnas.2010281117. URL https://www.pnas.org/ content/117/28/16096.
[4] Lucy B. Rorke. **Central Nervous System Plasticity and Repair**. Journal of Neuropathology & Experimental Neurology, 44(5):530–530, 09 1985. ISSN 0022- 3069. doi: 10.1097/00005072-198509000-00008. URL https://doi.org/10.1097/ 00005072-198509000-00008.
[5] Irwin Feinberg. **Schizophrenia: caused by a fault in programmed synaptic elimination during adolescence?** Journal of psychiatric research, 17(4):319–334, 1982.
[6] Peter H Rudebeck, Mark E Walton, Angharad N Smyth, David M Bannerman, and Matthew FS Rushworth. **Separate neural pathways process different decision costs**. Nature neuroscience, 9(9): 1161–1168, 2006.
[7] Tomáš Paus, Alex Zijdenbos, Keith Worsley, D Louis Collins, Jonathan Blumenthal, Jay N Giedd, Judith L Rapoport, and Alan C Evans. **Structural maturation of neural pathways in children and adolescents: in vivo study**. Science, 283(5409):1908–1911, 1999.

Objective:

- We want to maximize learning capacity of parameter space for RL agent.
- Our approach aims to identify the important connections among the neurons in a deep neural network that allow accomplishing a specific task.



Background:

- We leverage insights from recent *lottery ticket hypothesis* **[1, 2, 3, 4]** literature to construct *task-specific neural pathways* in multitask reinforcement learning in both online and offline settings.

- Scoring function **[2]** based on *connection sensitivity*:

$$\mathbf{S}(\theta_q) = \lim_{\epsilon \to 0} \left| \frac{\mathcal{L}(\theta_0) - \mathcal{L}(\theta_0 + \epsilon\delta_q)}{\epsilon} \right| = \left| \theta_q \frac{\partial \mathcal{L}(\theta_0)}{\partial \theta_q} \right|$$

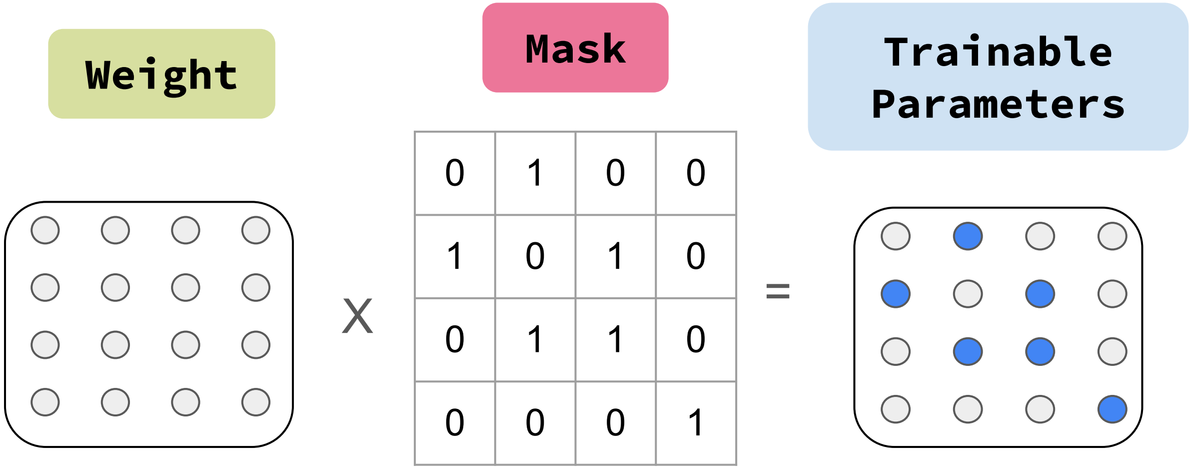
  - We measure the effect of weight θ_q on loss function $\mathcal{L}(\theta_0)$
  - δ_q is a vector whose q^{th} element equals θ_q and all other elements are 0.

[1] Jonathan Frankle and Michael Carbin. **The lottery ticket hypothesis: Finding sparse, trainable neural networks**, 2019.
[2] Namhoon Lee, Thalaiyasingam Ajanthan, and Philip HS Torr. **Snip: Single-shot network pruning based on connection sensitivity**. arXiv preprint arXiv:1810.02340, 2018.
[3] Hidenori Tanaka, Daniel Kunin, Daniel L. K. Yamins, and Surya Ganguli. **Pruning neural networks without any data by iteratively conserving synaptic flow**. CoRR, abs/2006.05467, 2020. URL https://arxiv.org/abs/2006.05467.
[4] Chaoqi Wang, Guodong Zhang, and Roger Grosse. **Picking winning tickets before training by preserving gradient flow**. arXiv preprint arXiv:2002.07376, 2020a.

Task-specific Subnetwork

$$m = \mathcal{T}_k\Big(\mathbf{S}(\theta; D)\Big)$$

$$\theta_m = \theta * m$$



$\mathcal{T}_k$: selects top k parameters

m: mask allows training task-specific subnetwork

Neural Pathway

- Neural Pathway (NP):
  - Let's define a neural network as $f(x, \theta)$
  - Apply mask m to compute neural pathway as $f(x, \theta * m)$
- For Actor-Critic Network:
  - Actor-Network: $\pi(\theta)$
  - Critic-Network: $Q(\phi)$
  - For n^{th} task compute two masks:
    - m_θ^n , m_ϕ^n
    - Actor-network: $\pi(\theta * m_\theta^n)$
    - Critic-Network: $Q(\phi * m_\phi^n)$

Data Adaptive Pathway Discovery (DAPD)

Scoring Function: $\mathbf{S}(\theta_q, D) = \left| \theta_q \frac{\partial \mathcal{L}(\theta_0; D)}{\partial \theta_q} \right|$

Adaptive learning:

1. Use the most recent data $D^{t-L:t} = \{(s, a, s', r)\}_{l=0}^{L}$

   $\mathbf{S}^{j}(\theta, D^{t-L:t})$

2. Stabilize parameter space update with K moving average:

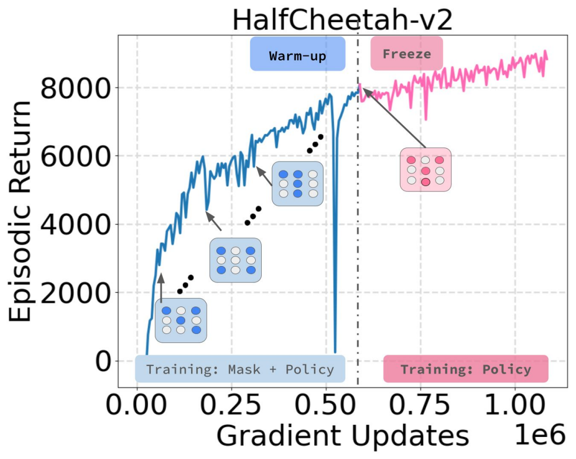
   $\frac{1}{K}\sum_{k=0}^{K-1} \mathbf{S}^{j-k}(\theta, D^{t-L:t})$

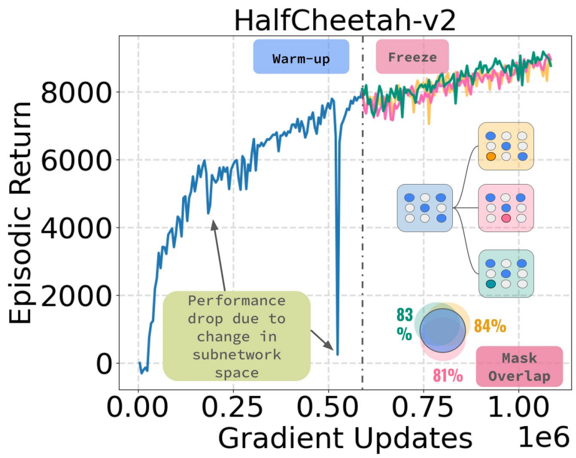
Updated Mask:

$$m = \mathcal{T}_k\left(\frac{1}{K}\sum_{k=0}^{K-1} \mathbf{S}^{j-k}(\theta, D^{t-L:t})\right)$$

Empirical Proof of Many Lottery Subnetwork Hypothesis:

- DAPD switch in-between multiple subnetwork during *warm-up* phase.
- It is essential to *Freeze* the sub-network once reached a *good-performance* (episodic reward, a hyper-parameter).
- **Fig 2** supports our hypothesis:
  - **There exists many sub-networks, which when trained separately can reach to almost equivalent performance.**





Data Adaptive Pathway Discovery (DAPD)

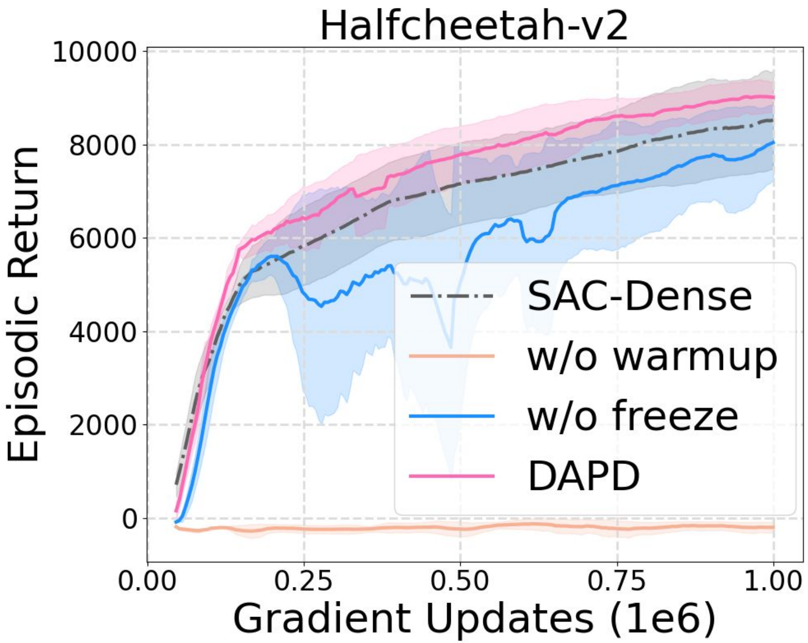
We show the importance of having a having **warm-up** and **freeze**, **two stages** of training in Fig (a).

Warm-up and Freeze:

- *Warm-up* : apply the adaptive mask
- *Freeze*: Keep the mask fixed for rest of the training once achieved a *threshold performance*, a hyper-parameter

Multitask setup:

- *Warm-up*: update the mask and corresponding weights independently
- *Freeze*: Fix the mask and merge of the weights.
- Compute *gradient average* for *overlapped mask*



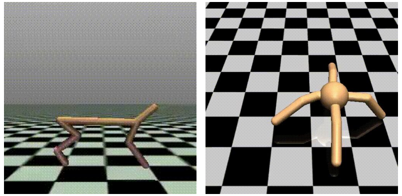
(a) Learning Curve

Experimental Setup:

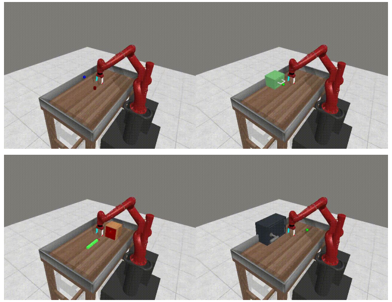
- **Environments**:
  - Continuous Control:
    - MuJoCo **[1]:** HalfCheetah, Walker2d, Ant, Hopper
    - MetaWorld **[2]**: MT10 tasks
- **Training step**: 1 million gradient step.
- **Evaluation**:
  - For MuJoCo we compute **episodic return**
  - For MetaWorld we compute the **success-rate** of task completion
  - For Offline RL setup we also report **normalized score [3]** w.r.t. training data performance:

$$normalized\ score = \left(\frac{\text{score - random score}}{\text{expert score - random score}} * 100\right).$$

  - We report the mean and standard-deviation over 5 seeds.



(a)MuJoCo



(b) MetaWorld

1. E Todorov Mujoco: A physics engine for model-based control, 2012
2. Tianhe Yu, Meta-World: A Benchmark and Evaluation for Multi-Task and Meta Reinforcement Learning, 2019
3. Justin Fu, D4RL: Datasets for Deep Data-Driven Reinforcement Learning, 2021

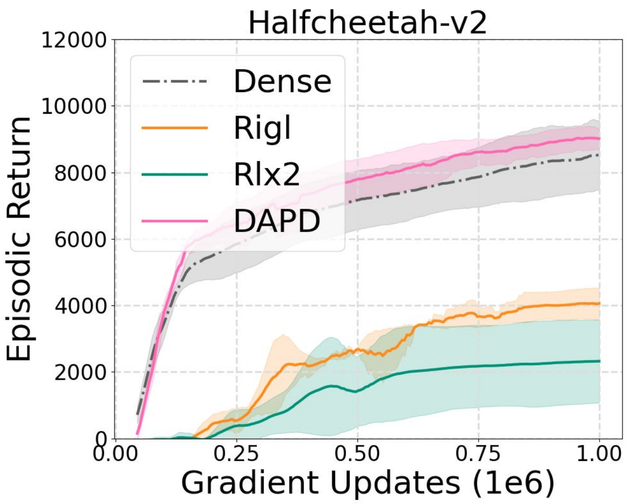
MuJoCo Benchmark:

- We compare DAPD at 95% sparsity with Dense network along with *topology based sparse methods* for RL RiGL**[1]** and Rlx2 **[2]** on MuJoCo tasks.
  - Topology based sparse method, randomly *grow* and *prune* fixed % of parameters
  - Very fragile to specific network sparsity ratio of actor and critic network
- We present the average episodic return over the last 10 evaluations over 5 seeds after 1 million training steps.
- We show DAPD exceeds other sparse training as well as the Dense network performance

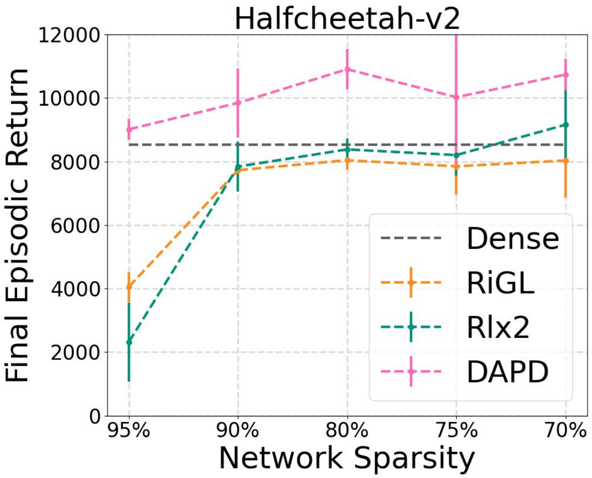
Environment	SAC-Dense	RiGL	Rlx2	SAC-DAPD
HalfCheetah-v2	8568.1 ± 1043.56	4043.95 ± 467.88	2333.31 ± 1241.16	**9028.02 ± 276.31**
Walker2d-v2	2972.49 ± 1691.47	260.3 ± 31.16	518.45 ± 205.16	**3846.3 ± 459.82**
Hopper-v2	3228.5 ± 301.88	174.89 ± 8.12	198.29 ± 10.39	**3359.88 ±46.57**
Ant-v2	3538.22 ± 654.76	954.2 ± 14.4	963.68 ± 6.96	**3916.65 ± 502.82**

1. Laura Graesser et al. The State of Sparse Training in Deep Reinforcement Learning. 2022.
2. Yiqin Tan et al. RLx2: Training a Sparse Deep Reinforcement Learning Model from Scratch. 2023

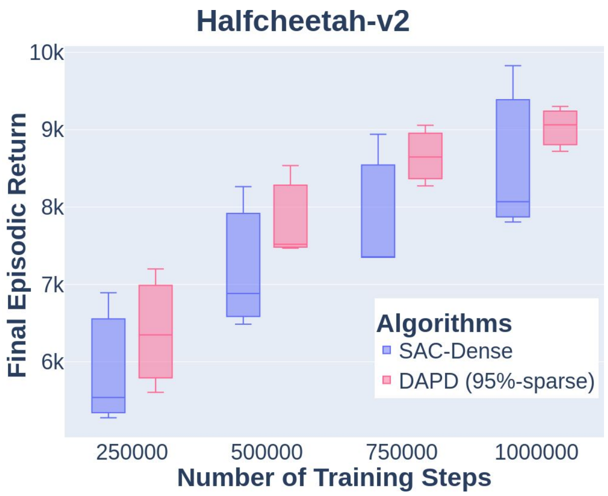
Performance Comparison



(a) Learning Curve



(b) Performance under Different Sparsity



(c) Sample Efficiency

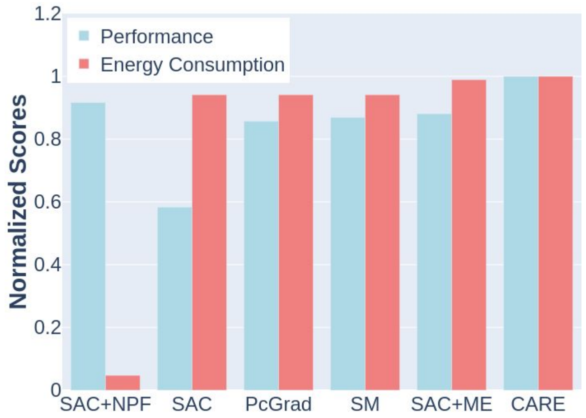
MetaWorld Benchmark:

- We compare performance of DAPD in MetaWorld multitask benchmark with various multitask algorithms.
  - We report the performance in following Table (a)

- We share the normalized performance and corresponding energy consumption in Fig (b)
  - DAPD can *potentially* safe **20x** energy consumption , under the assumption that *compute energy is proportional to FLOP counts.*

Experiments	SAC-DAPD	SAC-Dense	PCGrad	SM	SAC+ME	CARE
MT10 tasks	77 ± 1.3	49.0 ± 7.3	72.0 ± 2.2	73 ± 4.3	74 ± 4.3	$\mathbf{84 \pm 5.1}$
Parameter Counts	**17k**	340k	340k	135k	344k	486k
FLOPs	**16.9k**	339K	339K	78K	363K	368K
Energy Consumption, *Jules*	k	$20k$	$20k$	$20k$	$21.02k$	$21.25k$

(a) MetaWorld Benchmark



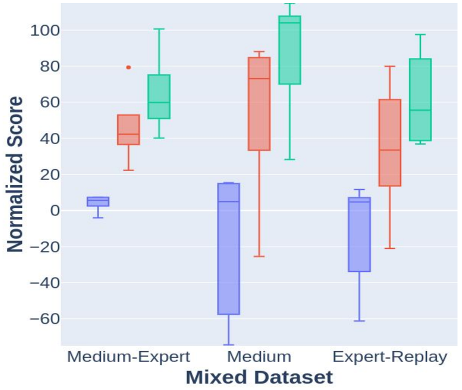
(b) Normalized Performance and Energy Consumption

Offline Benchmark:

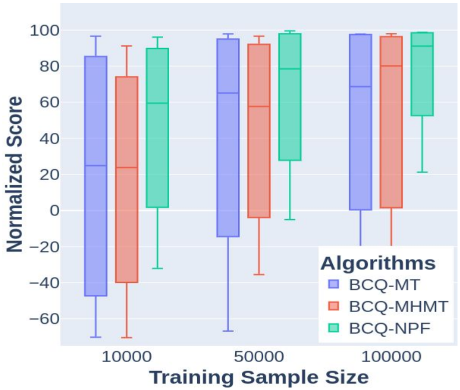
- Similar to supervised learning, we can determine the *lottery subnetwork* for Offline RL in a single-shot **[1]**.
- We compare the performance of NPF with Multitask (MT) and Multihead-Multitask (MHMT) baselines in BCQ **[2]**, IQL**[3]** offline RL algorithms in Table (a)
  - We provide the mean and standard deviation computer over 5 seeds
- We further compare the performance for BCQ-NPF under (b) mixed datasets and (c) varying number of training sample
- The results show NPF to be robust in performance.

(a) MetaWorld Benchmark

Experiment	**NPF**		Offline MT		Offline MHMT	
	BCQ	IQL	BCQ	IQL	BCQ	IQL
MT-10 tasks	**100 ± 0.0**	**97.3 ± 7.17**	81.5 ± 24.15	79.1 ± 26.81	95.9 ± 10.44	96.5 ± 7.10
Parameter Counts	**67k**	**54k**	1.34M	1.01M	1.38M	1.12M
FLOPs	**29.4K**	**53.6k**	589K	1073K	629k	1128k
Energy Consumption, *Joules*	k	k	$20k$	$20k$	$21.25k$	$21.02k$



(b)



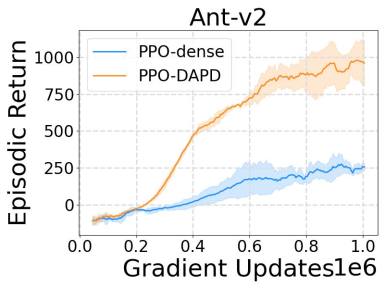
(c)

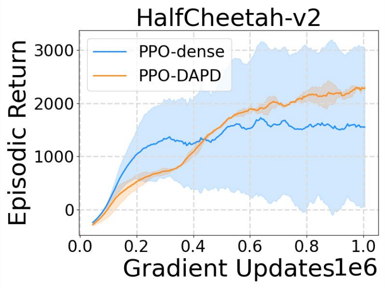
1. Single-Shot Pruning for Offline Reinforcement Learning, S Y Arnob, R Ohib, S Plis, D Precup, 2021
2. Off-Policy Deep Reinforcement Learning without Exploration, Scott Fujimoto, David Meger, Doina Precup, 2019
3. Offline Reinforcement Learning with Implicit Q-Learning, Ilya Kostrikov, Ashvin Nair, Sergey Levine, 2021

Empirical Proof of generalization:

Algorithmic Generalization:

- DAPD is effective with PPO in continuous control tasks.





Domain Generalization:

- To prove domain generalization, we show performance of DQN in Atari domain

Environment	DQN-dense (mean ± std)	DQN DAPD (mean ± std)
DemonAttack-v4	17670.33 ± 2829.91	**20803.33 ± 3273.07**
BreakoutNoFrameskip-v4	346.66 ± 12.21	**384.0 ± 15.80**
PongNoFrameskip-v4	**20.36 ± 0.58**	19.09 ± 0.77

We summarize our contributions as follows:

- We showcase **how to train multiple neural pathways for multi-task RL** where the **objective** is to **improve energy efficiency and reduce the carbon footprint associated with both offline and online RL training.**

- We introduce **Data Adaptive Pathway Discovery (DAPD)**, which **leverages network sensitivity** to adjust pathways in response to the **data distribution shifts encountered in online RL**. This capability enables us to **identify pathways at high levels of sparsity** and surpass competitive sparse training baselines .

- We demonstrate **superior sample efficiency** and **performance** in both single and multi-task RL compared to dense counterpart. The sparsity in the model induces **20x increase in energy efficiency** compared to alternative approaches, achieved through FLOP count reduction and the utilization of Sparse Matrix Multiplication (SpMM).

- Please check out our paper for more experimental results and discussion.