Attend our workshop

AMLD EPFL

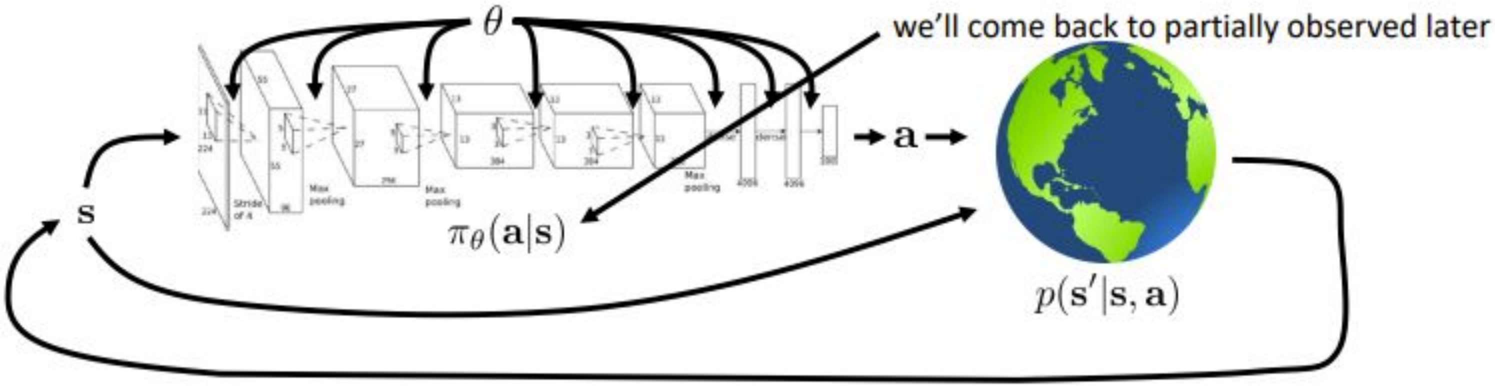
Deep Reinforcement Leaning for Satellite Constellation Planning and Routing

24 March
9:00 to 17:30

SwissTech Convention Center EPFL, Lausanne, Switzerland

Applied Machine Learning Days

The goal of Reinforcement Learning



$$\underbrace{p_\theta(\mathbf{s}_1, \mathbf{a}_1, \ldots, \mathbf{s}_T, \mathbf{a}_T)}_{p_\theta(\tau)} = p(\mathbf{s}_1) \prod_{t=1}^{T} \pi_\theta(\mathbf{a}_t | \mathbf{s}_t) p(\mathbf{s}_{t+1} | \mathbf{s}_t, \mathbf{a}_t)$$

$$p((\mathbf{s}_{t+1}, \mathbf{a}_{t+1}) | (\mathbf{s}_t, \mathbf{a}_t)) =$$
$$p(\mathbf{s}_{t+1} | \mathbf{s}_t, \mathbf{a}_t) \pi_\theta(\mathbf{a}_{t+1} | \mathbf{s}_{t+1})$$

$$\theta^\star = \arg\max_\theta E_{\tau \sim p_\theta(\tau)} \left[\sum_t r(\mathbf{s}_t, \mathbf{a}_t) \right]$$

Space and Reinforcement Learning

Challenges in Space Exploration

- Harsh, complex environment demanding high-precision mission planning.
- Economic imperative to minimize failure risks.
- Traditional planning: Time-consuming, reliant on complex math, often faces NP-hard problems.

Deep Reinforcement Learning for the win!

- Learns optimal actions by interacting with complex environments by trial and error.
- Environment retains complexity but is virtua so lows iteration cost.
- Reduced computational load in production.

Use Case 1: Operational Application of DRL in Earth Observation Satellite Scheduling

Context & Challenge

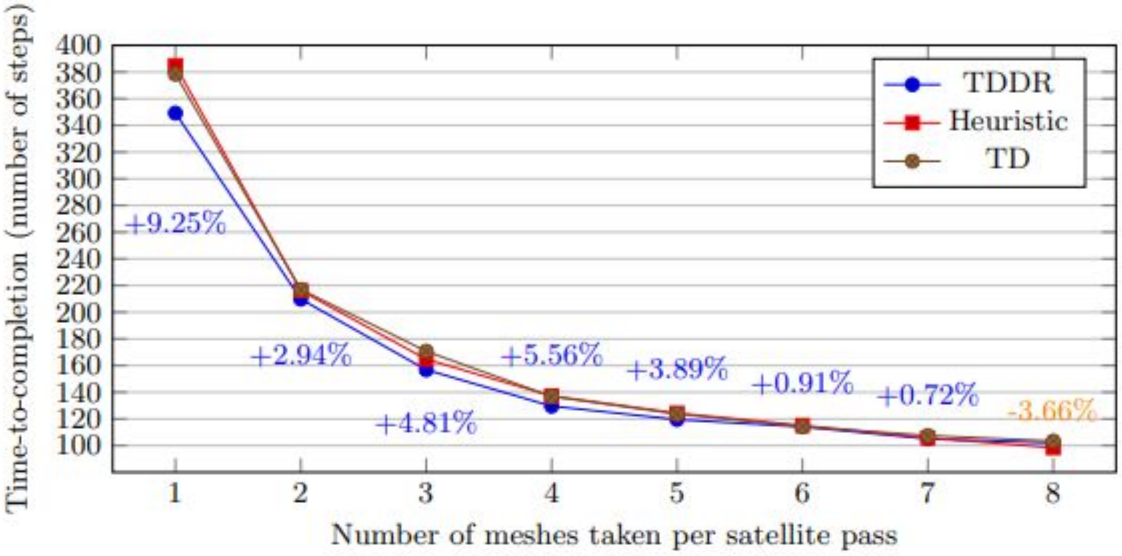
- Scheduling agile Earth Observation satellites (AEOS) is complex:
  - NP-hard problem with high combinatorial complexity.
  - Significant impact of weather uncertainties on image acquisition.
  - Large area coverage requests (countries/continents) extending over months or years, making long-term strategies crucial.

- Approach:

  - Utilizes the Actor Critic (A2C) algorithm enhanced with Transfer Learning, Domain Knowledge, and Domain Randomization (TDDR) via a mix of real weather forecast and generated one.

Use Case 1: Operational Application of DRL in Earth Observation Satellite Scheduling

- Benefits:
  - Effectively addresses the issue of weather forecast uncertainties.
  - Optimizes large area image acquisitions with strategic long-term planning.
  - Demonstrates superior performance over traditional heuristics in various weather conditions.



Use Case 2: Neural Combinatorial Optimization with Reinforcement Learning, *Bello et al.*

1. **Combinatorial Optimization:** given a finite set, one is tasked with finding an optimal object within that set.
2. **Examples**: Knapsack Problem, Travelling Salesman Problem. Both are NP hard to solve.
3. **Basic Solutions**: use approximation algorithms and predefined heuristics.
4. **Reinforcement Learning**: deriving suboptimal solutions without predefined heuristics: let the model find its own!

Use Case 2: Neural Combinatorial Optimization with Reinforcement Learning, *Bello et al.*

The travelling salesman problem: given a set of N cities, find the permutation that yields the shortest path.

Applications in satellite scheduling:

- Earth observation satellite scheduling + data transfer → challenge akin to the Traveling Salesman Problem (TSP)
- Satellites must optimally schedule imaging tasks and data transmission to ground stations, minimizing time and resources usage.
- We apply DRL to develop algorithms that learn to navigate the complexities of satellite orbits, weather uncertainties, and data transfer windows to find efficient solutions to the TSP, but on a global and dynamic scale.

Use Case 2: Neural Combinatorial Optimization with Reinforcement Learning, *Bello et al.*

- Given a permutation, one considers the associated tour length

$$L(\pi \mid s) = \left\| \mathbf{x}_{\pi(n)} - \mathbf{x}_{\pi(1)} \right\|_2 + \sum_{i=1}^{n-1} \left\| \mathbf{x}_{\pi(i)} - \mathbf{x}_{\pi(i+1)} \right\|_2 ,$$

- One aims to learn a policy which given a sequence of points, assigns high probability to short tours and low probabilities to long tours.

Use Case 2: Neural Combinatorial Optimization with Reinforcement Learning, *Bello et al.*

- We factorize the policy according to the following. Each of the product's term is represented as a nonparametric softmax module

$$p(\pi \mid s) = \prod_{i=1}^{n} p\left(\pi(i) \mid \pi(< i), s\right)$$

- Inspired by traditional sequence to sequence models trained on conditional log-likelihood. Two differences:

1. One does not want to be entangled with a fixed size vocabulary. Instead, one wants to be able to generalize beyond N cities
2. One would need truth labels, which would both be expensive to generate, bound the learning by predefined heuristics.

Use Case 2: Neural Combinatorial Optimization with Reinforcement Learning, *Bello et al.*

- Instead, we use Pointer Networks!
- Pointer Networks allow to point at a position in an input sequence, instead of predicting an index value in a fixed size set. This allows for size generalization
- Encoder/decoder architecture with **attention**

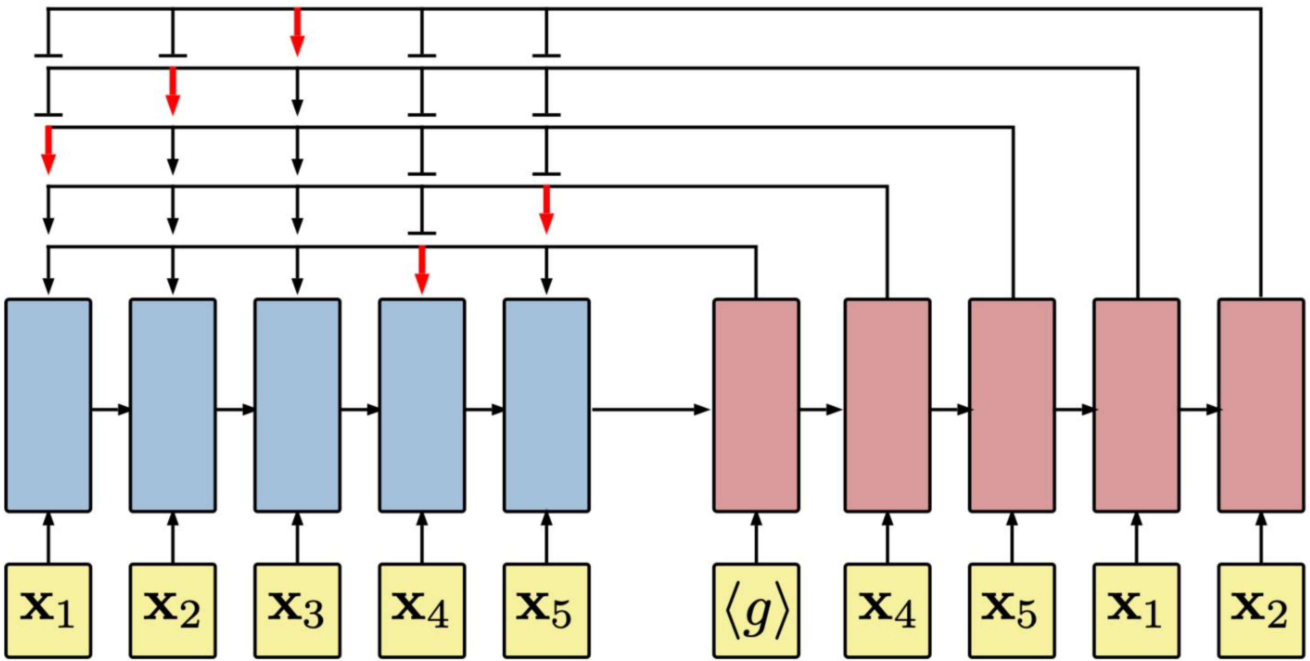


Figure 1: A pointer network architecture introduced by (Vinyals et al., 2015b).

Use Case 2: Neural Combinatorial Optimization with Reinforcement Learning, *Bello et al.*

- Embed the graph in high-dimensional space
- Instantiate two LSTM layers: **encoder** and **decoder**.
- The **encoder** reads one point at a time, encoding it in a latent space
- The **decoder** uses a **pointing mechanism** (attention) to output a distribution over the next point to visit. The selected point selected is then passed as input to the decoder. The first point to be passed is a trainable parameter.

Use Case 2: Neural Combinatorial Optimization with Reinforcement Learning, *Bello et al.*

Attention is all you need: predicts a distribution over the set of k references.

$$u_i = \begin{cases} v^\top \cdot \tanh\left(W_{ref} \cdot r_i + W_q \cdot q\right) & \text{if } i \neq \pi(j) \text{ for all } j < i \\ -\infty & \text{otherwise} \end{cases} \text{ for } i = 1, 2, \ldots, k$$

$$A(ref, q; W_{ref}, W_q, v) \stackrel{\text{def}}{=} softmax(u).$$

$$p(\pi(j)|\pi(< j), s) \stackrel{\text{def}}{=} A(enc_{1:n}, dec_j).$$

Use Case 2: Neural Combinatorial Optimization with Reinforcement Learning, *Bello et al.*

- Attention function represents the degree to which the model points to reference i, upon seeing query q.
- **Additional step:** use glimpses. Dot product between attention probabilities and references.

Use Case 2: Neural Combinatorial Optimization with Reinforcement Learning, *Bello et al.*

- **Loss function:** Given a sequence of points, minimize the expected tour length. We start at a first point, we make a decision about the second point etc… We minimize the expected total length.

$$J(\boldsymbol{\theta} \mid s) = \mathbb{E}_{\pi \sim p_{\theta}(.|s)} L(\pi \mid s) \, .$$

Use Case 2: Neural Combinatorial Optimization with Reinforcement Learning, *Bello et al.*

- Use Policy Gradients with baseline (EMA of rewards)

$$\nabla_\theta J(\theta \mid s) = \mathbb{E}_{\pi \sim p_\theta(.|s)} \left[\big(L(\pi \mid s) - b(s) \big) \nabla_\theta \log p_\theta(\pi \mid s) \right]$$

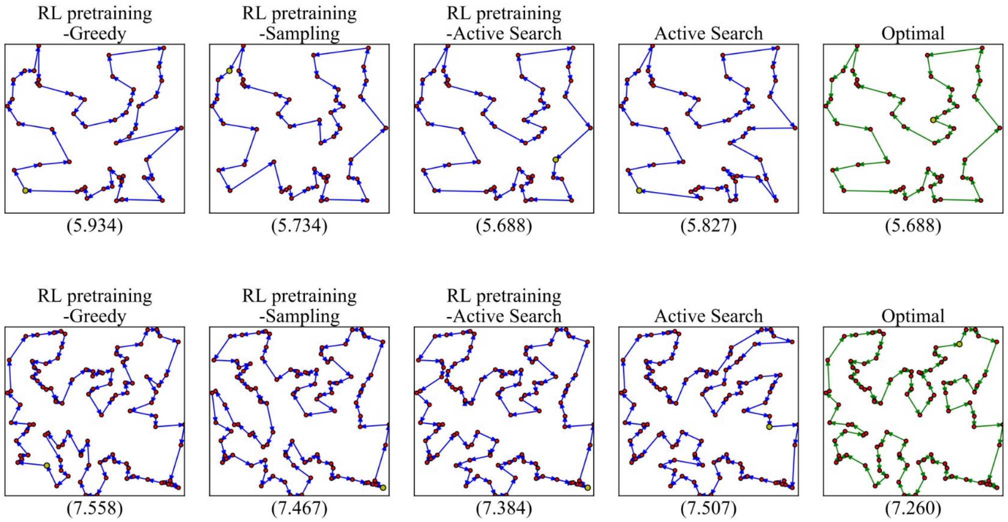
Neural Combinatorial Optimization with Reinforcement Learning, *Bello et al.*

Algorithm 1 Actor-critic training

procedure $\textsc{Train}$(training set S, number of training steps T, batch size B)
  Initialize pointer network params θ
  Initialize critic network params θ_v
  for $t = 1$ to T **do**
    $s_i \sim \textsc{SampleInput}(S)$ for $i \in \{1, \ldots, B\}$
    $\pi_i \sim \textsc{SampleSolution}(p_\theta(.|s_i))$ for $i \in \{1, \ldots, B\}$
    $b_i \leftarrow b_{\theta_v}(s_i)$ for $i \in \{1, \ldots, B\}$
    $g_\theta \leftarrow \frac{1}{B}\sum_{i=1}^{B}(L(\pi_i|s_i) - b_i)\nabla_\theta \log p_\theta(\pi_i|s_i)$
    $\mathcal{L}_v \leftarrow \frac{1}{B}\sum_{i=1}^{B} \|b_i - L(\pi_i)\|_2^2$
    $\theta \leftarrow \textsc{Adam}(\theta, g_\theta)$
    $\theta_v \leftarrow \textsc{Adam}(\theta_v, \nabla_{\theta_v}\mathcal{L}_v)$
  end for
  return θ
end procedure

Neural Combinatorial Optimization with Reinforcement Learning, *Bello et al.*

- At test time, we sample greedily!
- **Other strategies:** active search (we will not cover this)



Connect with us!

https://www.linkedin.com/in/alexandre-carlhammar-852844240/

https://www.linkedin.com/in/theo-le-fur-469639265/

Questions?