SQL Injections and Reinforcement Learning:

An Empirical Evaluation of the Role of Action Structure

Nordsec 2021

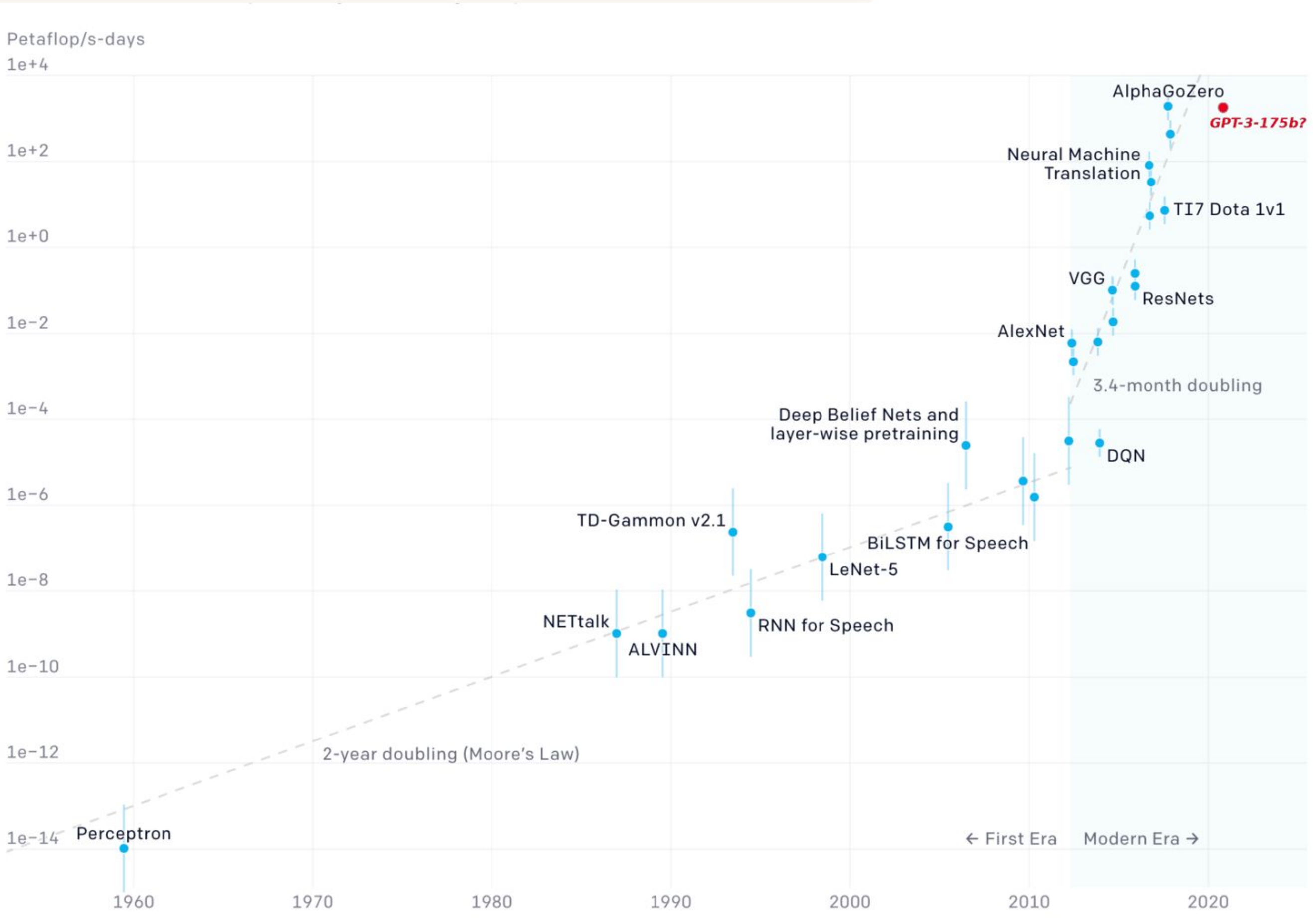
UiO : University of Oslo

Mila

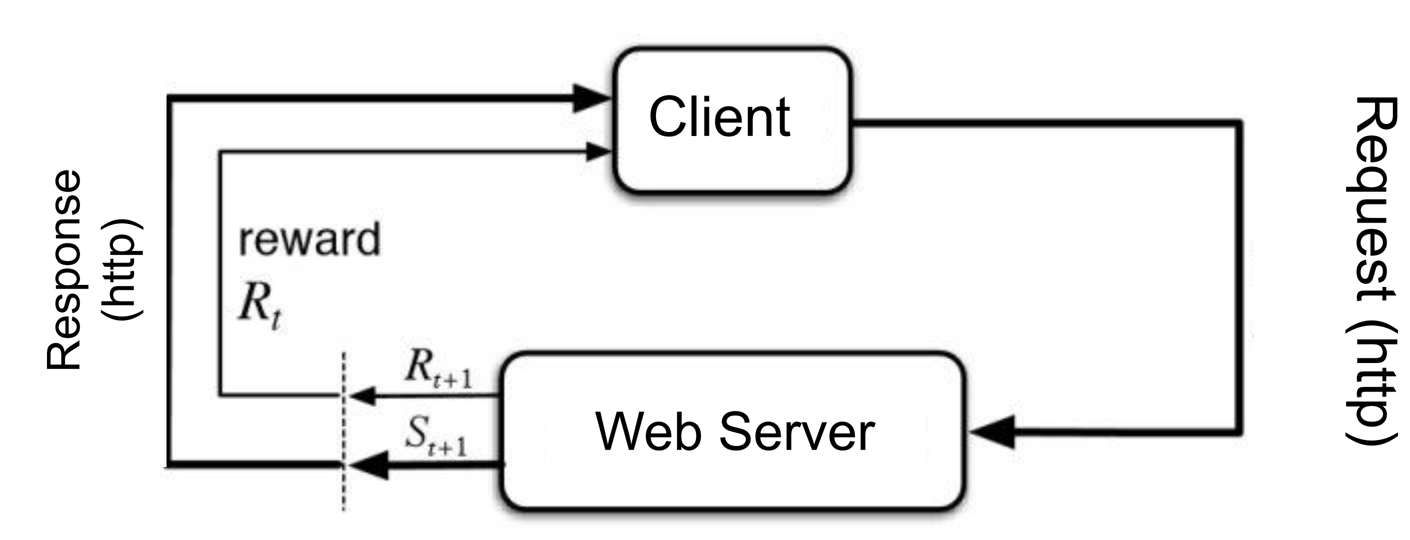
Deep Learning

- **Computational power is cheap.**
- **The exponential growth is holding.**

- **GPT3** 175 billion parameters.
- **OpenAI Codex** Code generation.
- **AlphaGo** 1,920 CPUs 280 GPUs.

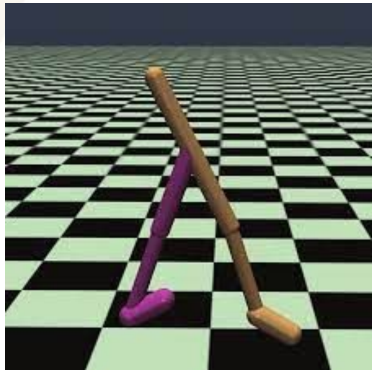


Reinforcement Learning for Web Fuzzing



- **Action space**
  - How does the agent communicate to the environment
  - GET /<payload> HTTP/1.1
  - payload can be: hardcoded, composed of a vocabulary, characters, bits.
- **Observation space**
  - How does the environment communicate with the agent
  - HTTP status code, HTML

Deep Reinforcement Learning



- **Reinforcement Learning with Deep Neural Networks**
- **The classical problem settings:**
    - Robotics, physics symulators (OpenAI gym MuJoCo-ant)
        - ~1k steps/second
        - ~10k steps/second with specialized hardware (20201, TPUs)
- **Web Server:**
    - Commercial servers:
        - 1.2 Million requests/sec per second (in 2016)
        - 5k req/s, random web server (ab -n 10000 -c 100 http://mila.quebec/)

    - Deep RL agents require a lot of environment interactions (~1M for simple problems).
        - Simulation speed is crucial

The Setup

The Environment

- **2 SQL tables**
  - User, the working table.
  - Private, which holds the flag.
- **Processed responses:**
  - Unsanitized user input field:
    - At every attempt we select one out of three possible queries
    - And a variable number of column.
    - Probing is required for both.

-Best possible solution requires at most 5 actions:

- 3 actions to guess the correct number of columns
- 2 actions to find the escape

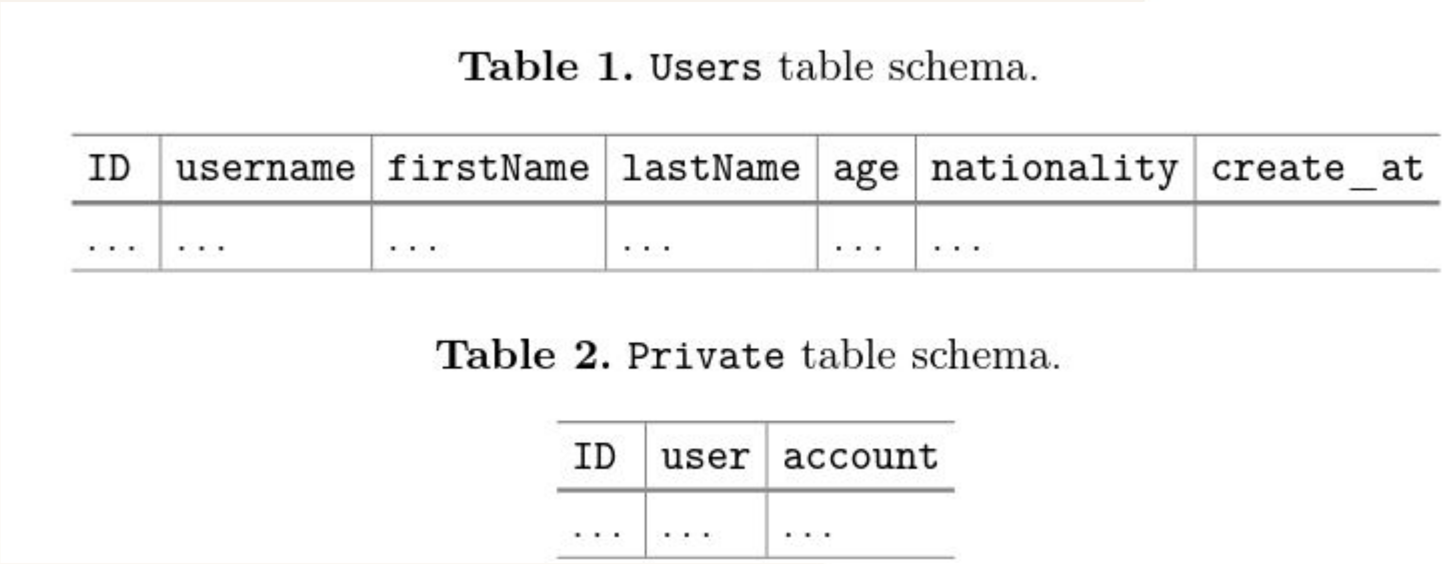
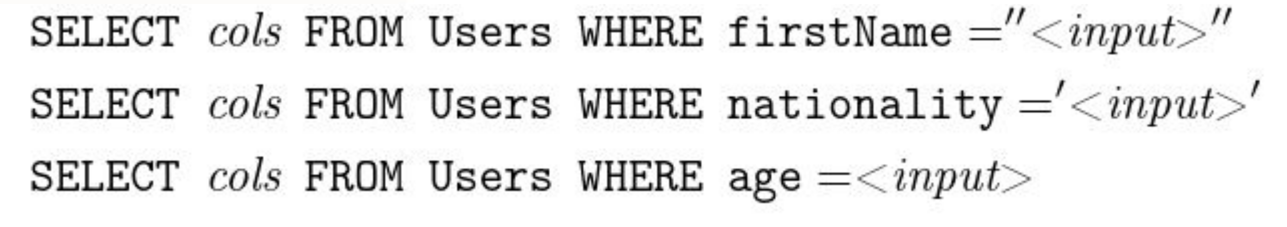


Table 1. Users table schema.

ID	username	firstName	lastName	age	nationality	create_at
...	...	...	...	...	...	

Table 2. Private table schema.

ID	user	account
...	...	...



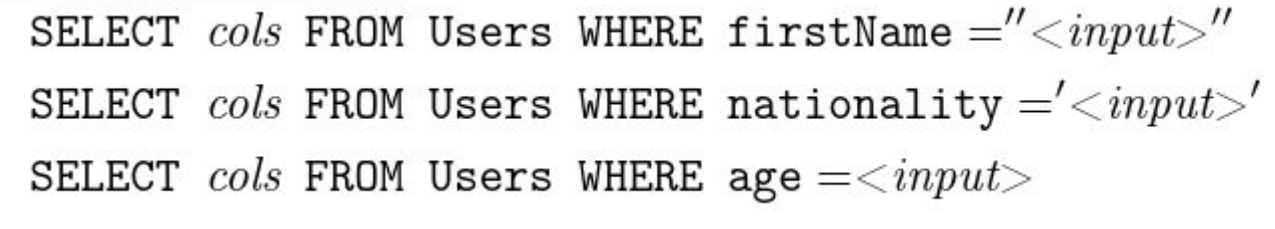
SELECT *cols* FROM Users WHERE firstName ="<*input*>"

SELECT *cols* FROM Users WHERE nationality ='<*input*>'

SELECT *cols* FROM Users WHERE age =<*input*>

The Environment

- **One Algorithm, Multiple tasks**
    - Reconnaissance and exploitation are interleaved



SELECT $cols$ FROM Users WHERE firstName $='' <input> ''$

SELECT $cols$ FROM Users WHERE nationality $=' <input> '$

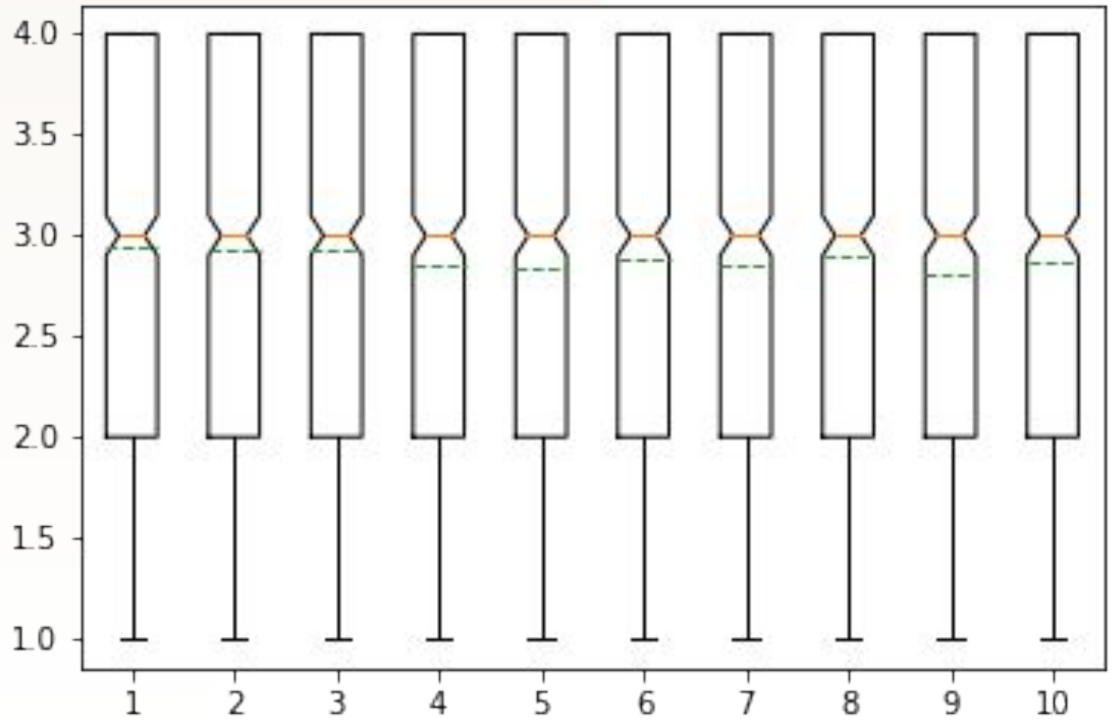
SELECT $cols$ FROM Users WHERE age $= <input>$

Structured agent

- **Hard coded requests (25 actions)**
  - 6 escape probing (e.g. 1' or 1=1 --)
  - 9 exploratory (e.g. 1" union select NULL --)
  - 9 exploit attempts (e.g. 1" union select account, NULL from private --)
  - 1 other “\0”
- **Processed responses:**
  - A list of 25x1 numbers, one for each possible action.
    - 0 at an index means that the action has been tried and an sql error was returned
    - 1 action never tried
    - 2 action tried -> no data returned
    - 3 action tried -> something
    - 4 flag found
      action 1: -> obs [2, 1, 1, 1, 1, 1, 1, ...] # valid sql, no data
      action 3: -> obs [2, 1, 3, 1, 1, 1, 1, ...] # valid sql, data was returned

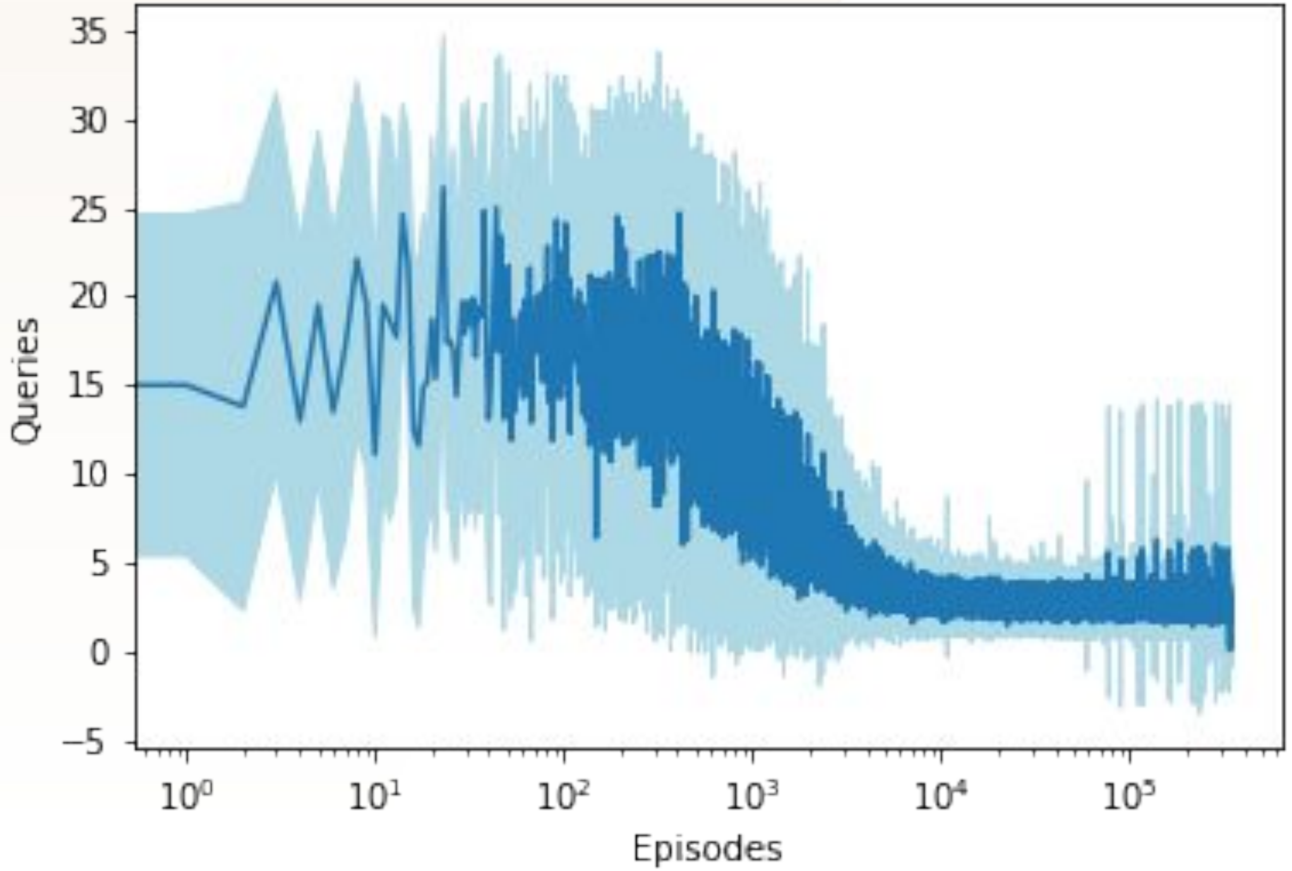
Structured agent, Final Performance

- Performance of 10 different agents:
- Orange line gives the median number of queries (~3), green mean
- 25%, 50%, 95% confidence interval for the median



Structured agent, Training

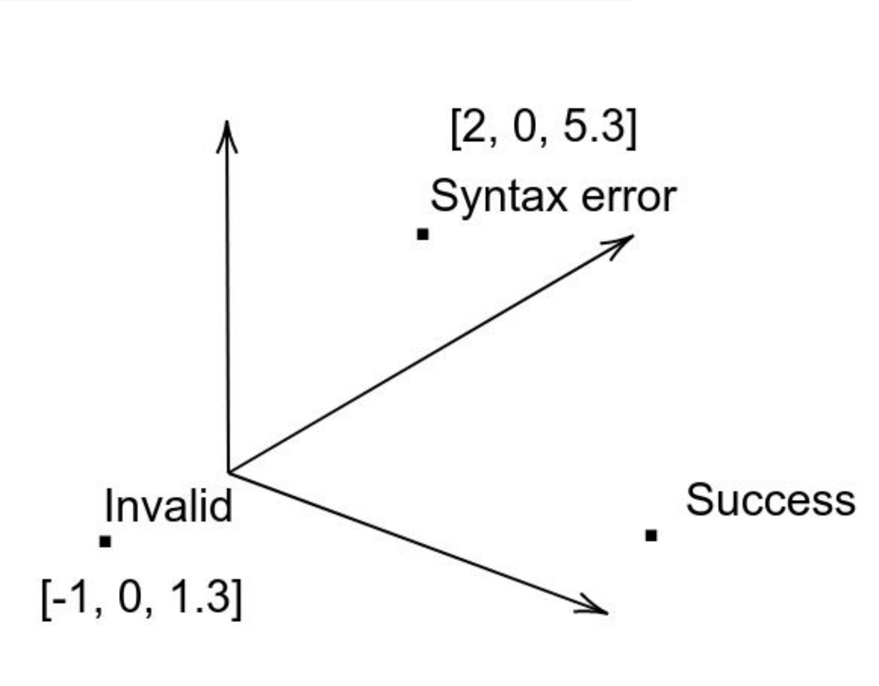
- Number of queries to capture the flag
- ~10k steps to convergence
- 3 queries per attempt
- w.r.t. 17 queries by trying randomly



Structured agent

- **Pros:**
  - Easy to encode human knowledge.
  - Higher efficiency to solve tasks.
- **Cons:**
  - As good as the encoded knowledge.
  - Unlikely to find surprising results.

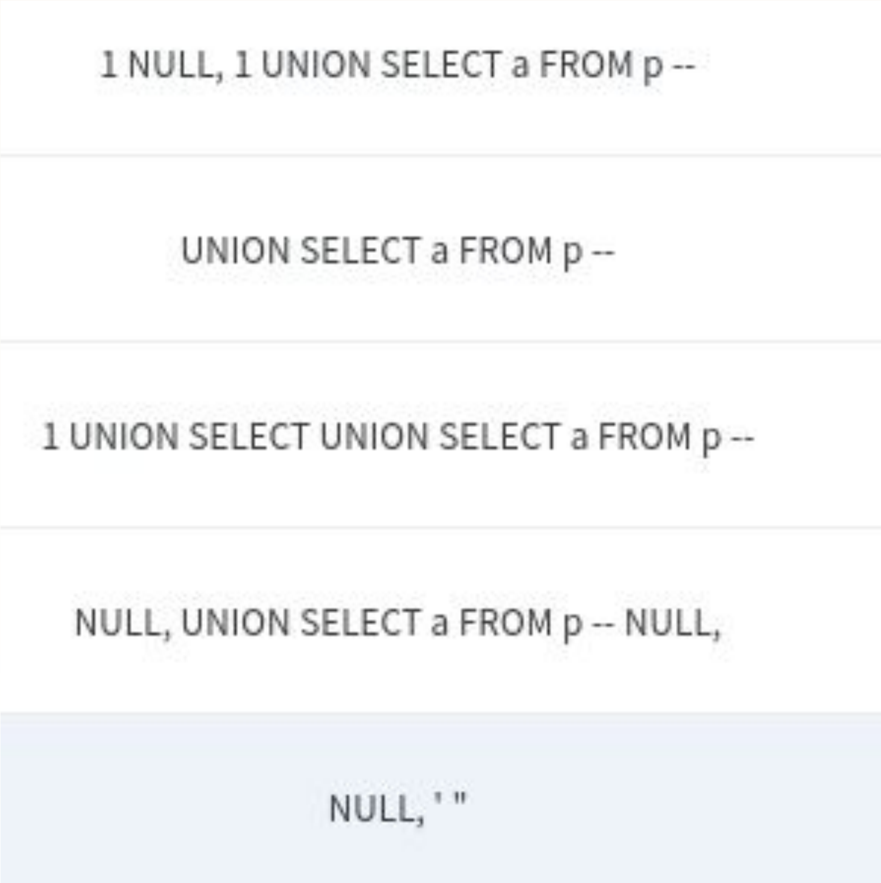
Unstructured Agent



- **Huge action space >100k distinct actions**
  - Actions are selected from a dictionary of:
    - SQL keywords
    - A “wordlist”
  - An action is a mix of the dictionary words
  - a = [34, 8, 9, 12, 0]
- **Observations**
  - A transformation of the SQLite error message
  - e.g. “SELECTs to the left and right of UNION do not have the same number of result columns” becomes “columns”
  - any variation of “syntax error” becomes “Syntax error”
  - Then the string then becomes a multidimensional embedding, [-2, 0, 5.3]

Training progress

1000 steps



1 NULL, 1 UNION SELECT a FROM p --

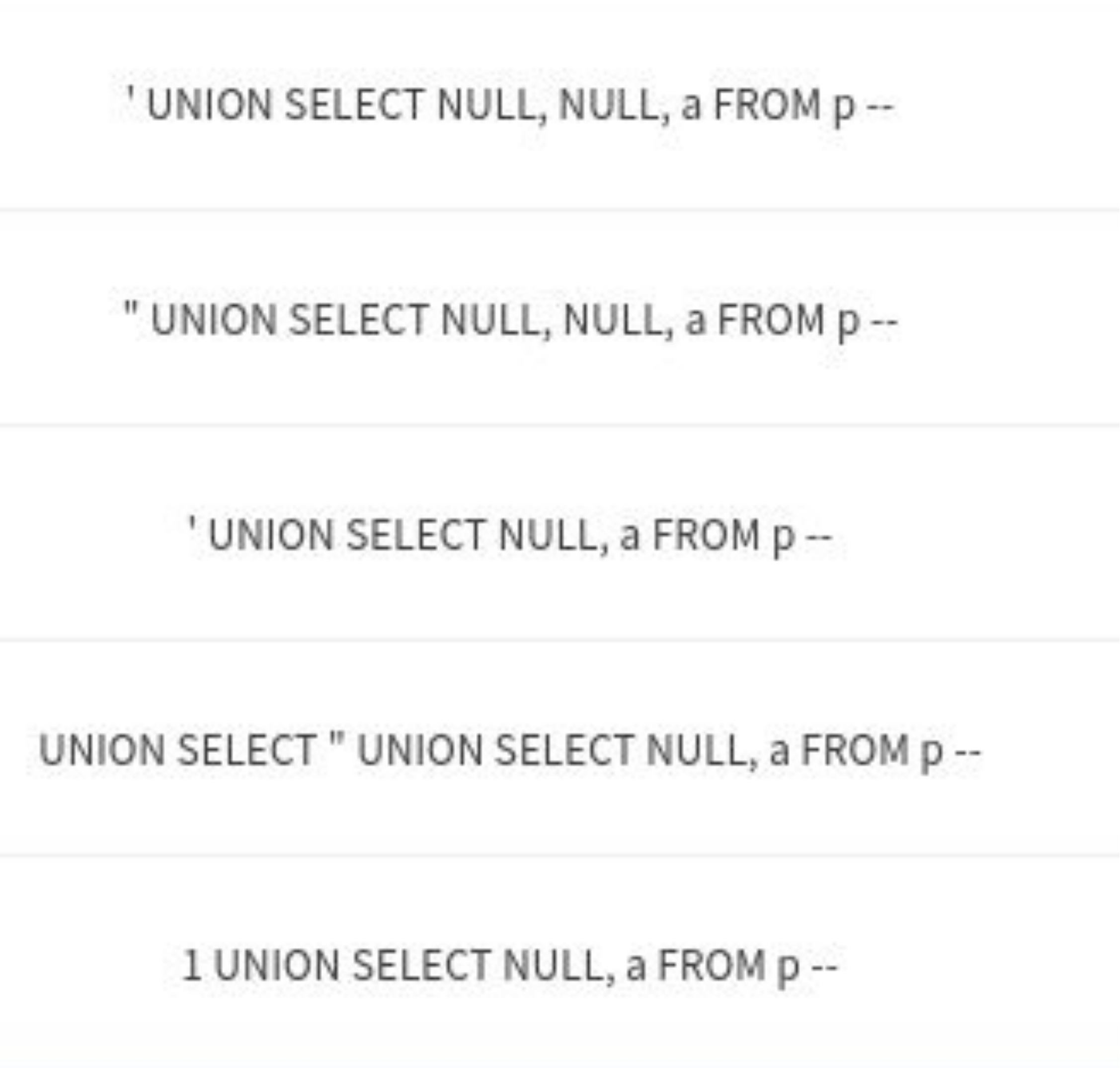
UNION SELECT a FROM p --

1 UNION SELECT UNION SELECT a FROM p --

NULL, UNION SELECT a FROM p -- NULL,

NULL, ' "

16000 steps



' UNION SELECT NULL, NULL, a FROM p --

" UNION SELECT NULL, NULL, a FROM p --

' UNION SELECT NULL, a FROM p --

UNION SELECT " UNION SELECT NULL, a FROM p --

1 UNION SELECT NULL, a FROM p --

Structured agent

- **Pros:**
  - Does not rely on human knowledge (and limitations).
  - Scales with compute, not with man-hours.
  - Can find surprising solutions.
- **Cons:**
  - Complex algorithms to write.
  - Requires Reinforcement Learning + Security + Natural Language Processing understanding.
  - Young field, no turnkey solutions.
  - Good solutions emerge after 1-2 Million interactions.

Why Reinforcement Learning now

- **We have the resources and algorithms:**
  - Web servers are fast.
  - These algorithms scale with compute speed.
  - Can find surprising solutions.
  - Can handle as many edge cases as there are.
- **Experts define the problem.**
  - What makes a successful solution.
  - How does the program solution look like (neural net architecture)
  - Compute finds the program.
- **Exponential scale gives exponential results**
  - Often is just a matter of **more compute**.
  - DeepMind, OpenAI showed we did not yet hit the limit of scale

Questions?
Contacts

- Manuel Del Verme (manuel.delverme@gmail.com)
- Åvald Åslaugson Sommervoll (aavalds@ifi.uio.no)
- Laszlo Erdodi
- Simone Totaro
- Fabio Massimo Zennaro

Mila

McGill

UiO University of Oslo