When does reinforcement learning stand out in quantum control? A comparative study on state preparation (arXiv:1902.02157)

Xiao-Ming Zhang, Zezhu Wei, **Raza Asad**, Xu-Chen Yang, Xin Wang

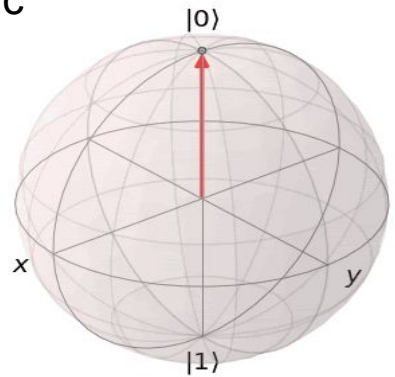
Asad Raza,
Department of Mathematics,
City University of Hong Kong

Outline

- Problem: State preparation
- Introduction to the algorithms used
- Review of Reinforcement Learning
- Control Dynamics and Constraints
- Results under various constraints

State Preparation

- **Task**: Steer the qubit from |0> to |1> under certain constraints, with fidelity being the evaluation metric

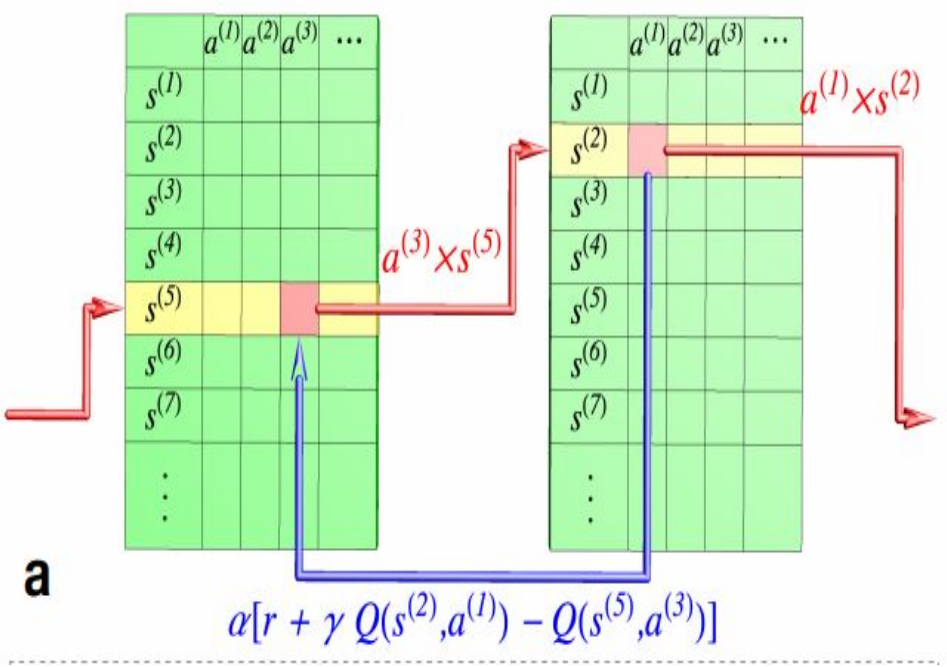


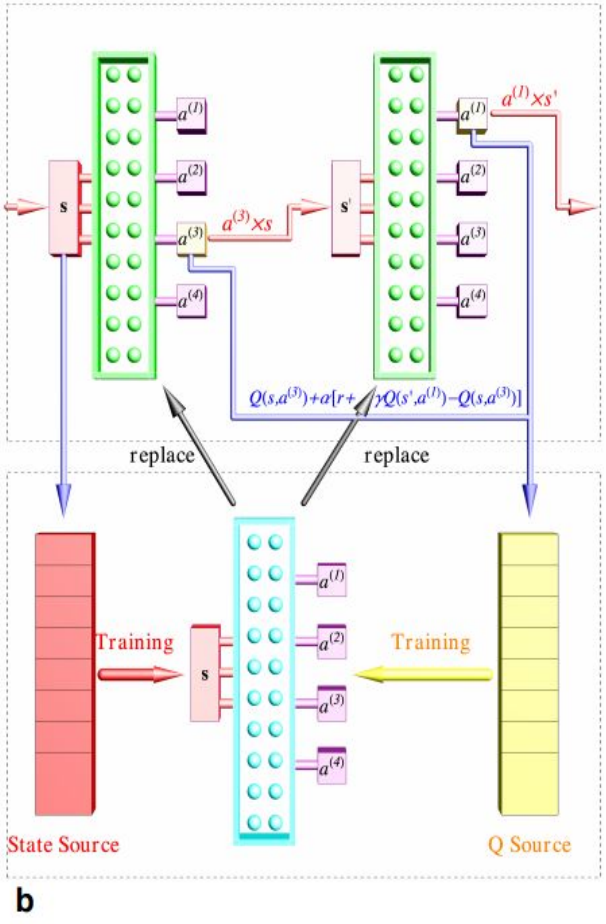
- Under different scenarios of constraints, which algorithm works best? We compare the efficacy of Stochastic Gradient Descent (SGD), Krotov, Tabular Q-Learning (TQL), Deep Q-Learning (DQL) and Policy Gradient (PG).

Introduction to the algorithms used

- SGD: Gradient based algorithm that updates control field using the gradient of the fidelity cost function
- Krotov: The evolved state is projected to the target state, defining a co-state encapsulating the mismatch between the two. Then the co-state is propagated backward to the initial state, during which process the control fields are updated. Repeat until co-state is identical to the target
- TQL: Each time the agent takes an action, a reward r is generated according to the distance between the resulting state and the target, which updates the Q-table
- DQL: Same as TQL, except that Q-table is replaced by a neural network
- PG: Similar to TQL/DQL, wherein PG takes the state as the input, the network outputs the probability of choosing each action

TQL and DQL


a
$a^{(1)}$ $a^{(2)}$ $a^{(3)}$ ...
$s^{(1)}$ $s^{(2)}$ $s^{(3)}$ $s^{(4)}$ $s^{(5)}$ $s^{(6)}$ $s^{(7)}$
$a^{(3)} \times s^{(5)}$
$a^{(1)} \times s^{(2)}$
$\alpha[r + \gamma\, Q(s^{(2)}, a^{(1)}) - Q(s^{(5)}, a^{(3)})]$


s
s'
$a^{(1)}$ $a^{(2)}$ $a^{(3)}$ $a^{(4)}$
$a^{(3)} \times s$
$a^{(1)} \times s'$
$Q(s,a^{(3)}) + \alpha[r + \gamma Q(s',a^{(1)}) - Q(s,a^{(3)})]$
replace
replace
Training
Training
State Source
Q Source
b

Reinforcement learning: Markov Decision Process

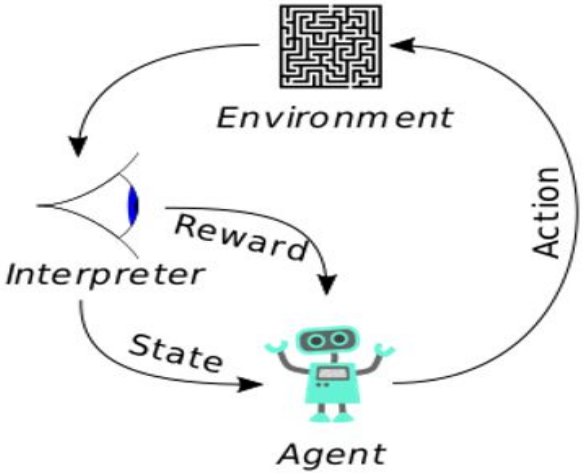
(1) states **S**, (2) actions **A**, (3) reward *r(s)*

(4) policy $\pi(a|s)$: probability of choosing each action at states s

states $\longrightarrow$ quantum states

actions $\longrightarrow$ Hamiltonian operations

rewards $\longrightarrow$ functions of fidelities



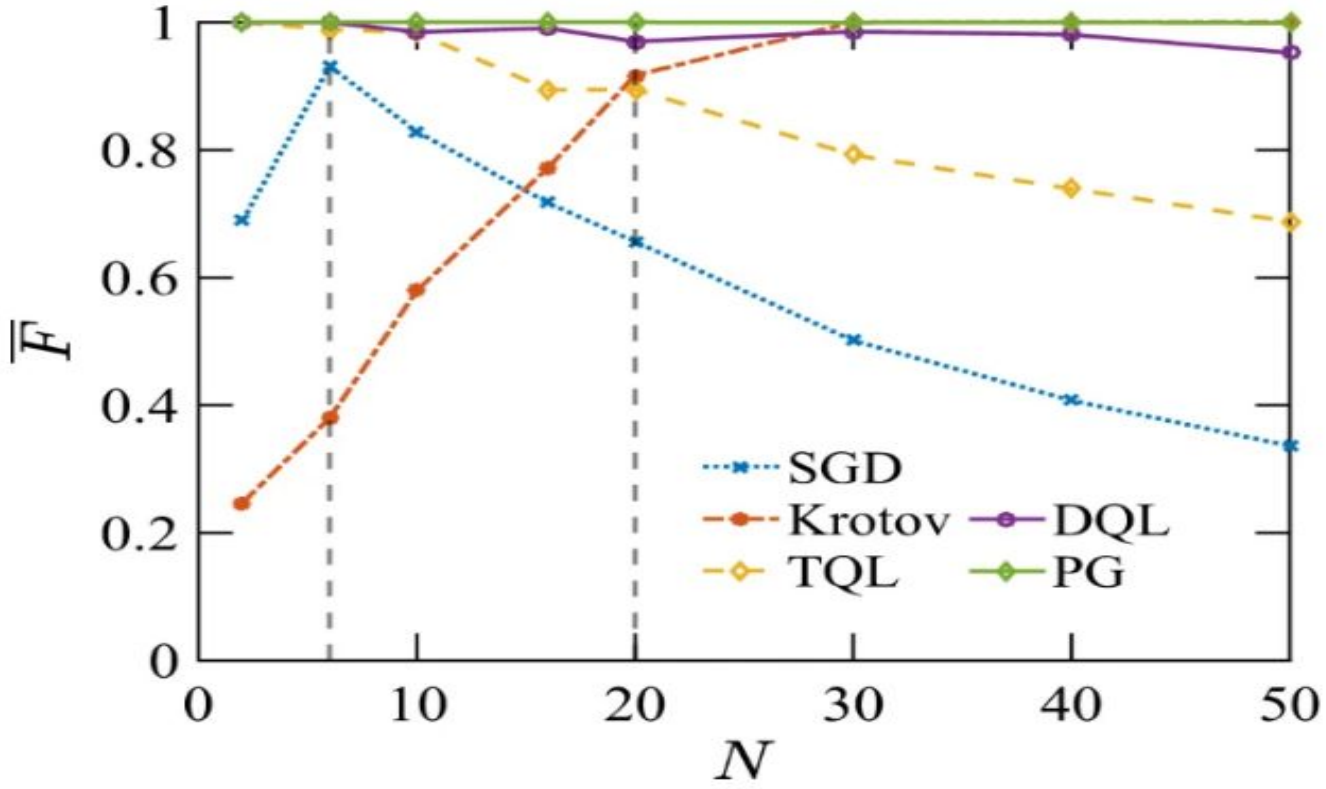
https://en.wikipedia.org/wiki/Reinforcement_learning

Dynamics of Control

$$H[J(t)] = 4J(t)\sigma_z + h\sigma_x\,,$$

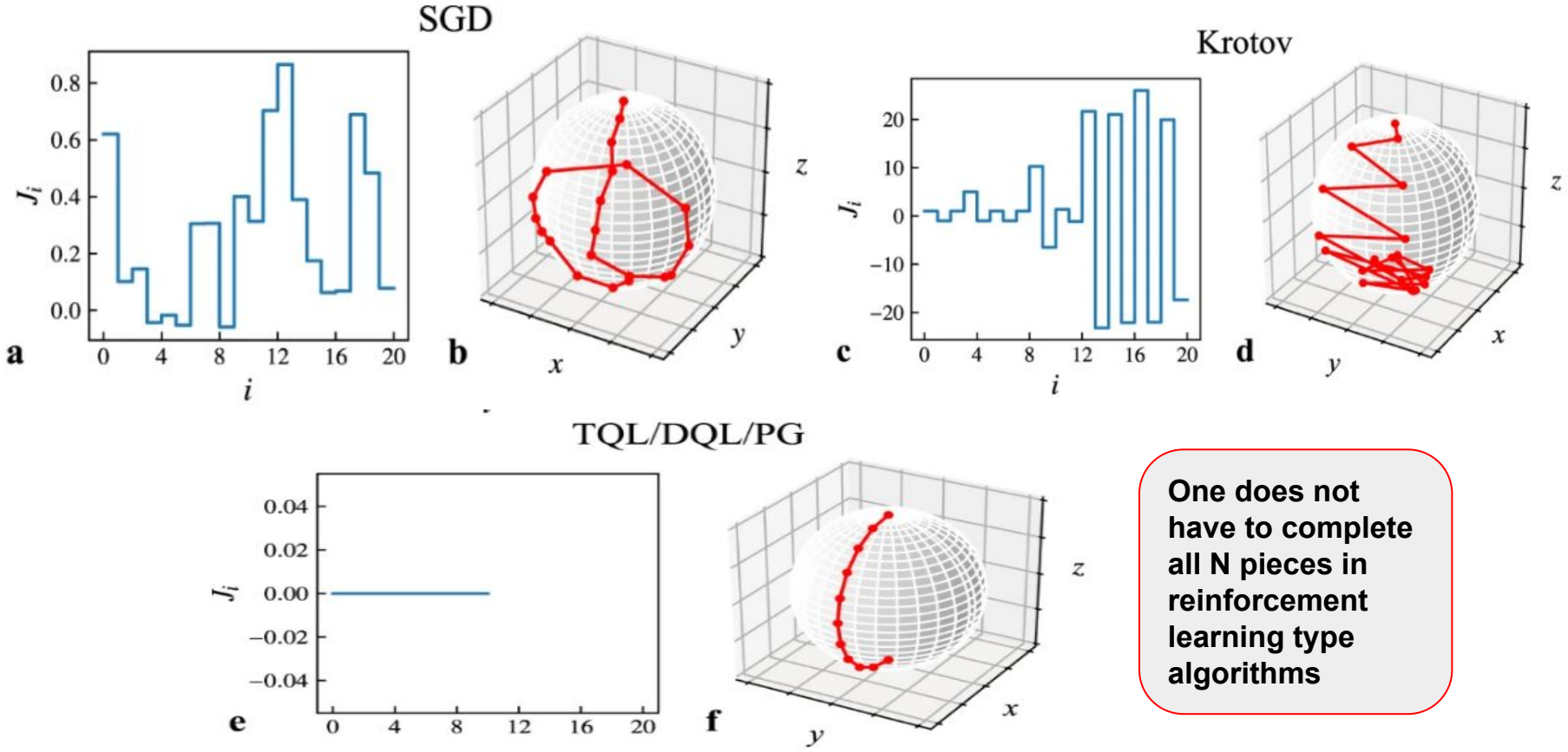
- Task: Find J(t) s.t. The fidelity between the initial state (|0>) and the final state (|1>) is as high as possible in time T = 2π
- Control is performed by N piecewise constant pulses, each with the same duration T/N.
- On the ith step $J(t) = J_i$ and $|\psi_i\rangle = U_i|\psi_{i-1}\rangle$, where $U_i = \exp\{-iH(J_i)dt\}$

Results



Among all five algorithms, PG is consistently the best. Apart from PG, DQL gives the highest fidelity for N < 30, but due to its nonzero failure probability, it is slightly outperformed by Krotov for N > 30.

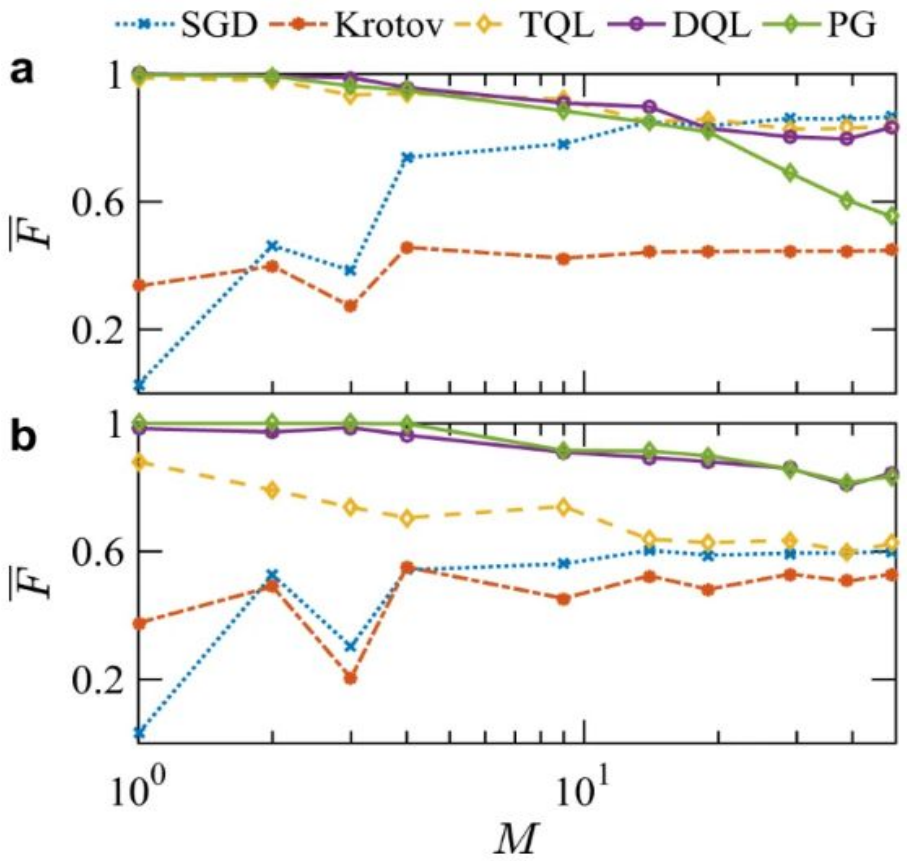
Pulse Profiles



One does not have to complete all N pieces in reinforcement learning type algorithms

Results with M+1 allowed control field values

Note that the averaged fidelities of three reinforcement learning algorithms decreases with M, albeit not considerably. This is because TQL, DQL and PG favor bounded and concrete sets of actions, and more choices will only add burden to the searching process, rendering the algorithms inefficient



Summary

- RL works well in discrete settings and SGD in continuous ones
- Performance of SGD, TQL and DQL deteriorates with increasing N. However, the performance of SGD deteriorates quickly

Thank you for listening!
Questions?