A curious property of randomized log-space search algorithms is that their outputs are often longer than their workspace. One consequence is that there is no clear way to reproduce the same output when running the algorithm twice on the same input. It is not feasible to store the random bits (or the output) of the previous run in log-space, and using new random bits in another execution can result in a different output. This leads to the question: how can we reproduce the results of a randomized log space computation of a search problem?

We will give a precise definition of this notion of "reproducibility". Then we will show that every problem in search-RL has a randomized log-space algorithm where the output can be reproduced. Reproducibility can be thought of as an extension of pseudo-determinism. Indeed, for some problems in search-RL we show pseudo-deterministic algorithms whose running time significantly improve on known deterministic algorithms.

Joint work with Yang Liu.