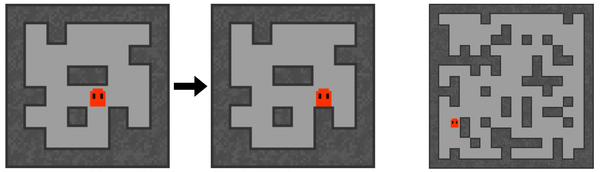
QORA: Zero-Shot Transfer via Interpretable Object-Relational Model Learning

Gabriel Stella Dmitri Loguinov

Texas A&M University

What does QORA do?

The task we are concerned with is **learning environmental dynamics from observation**. Our goal is to **efficiently** learn accurate, **interpretable** rules, that **generalize** to unseen scenarios, without relying on domain-specific information.



(a) example state transition (training)

(b) unseen state (testing)

We operate in an object-oriented framework where each environment consists of a tuple (M, C, S, B, A, T):

- M is the set of member attributes, e.g.: 2D position, 3D rgb color
- C is the set of classes, e.g.: player, wall, door
- S is the set of all possible states, each of which consists of a set of objects
- B is the distribution over initial game states, which may be parameterized
- A is the set of actions, e.g.: up, down, left, right, stay
- T is the transition probability distribution, which defines the domain's dynamics

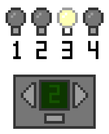
To measure prediction error, we calculate the Earth Mover's Distance (EMD) between the learner's predicted distribution and the environment's true distribution over future states.

Benchmark Environments

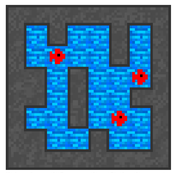
These domains allow us to test specific aspects of a learner's operation.

- **walls**: This domain contains walls and a player object. The singular player entity is controlled by directional actions; movement is blocked by walls. This tests a learner's ability to discover simple relational rules.
- **lights**: This non-gridworld domain contains lights and a re-assignable switch that can toggle the state of the light it refers to.
- **doors**: This domain adds colored doors and a new action to the walls domain. The action allows the player to toggle its color; the player can only pass through doors that are the same color as it.
- **fish**: This stochastic domain contains walls and one or more fish. On each iteration, each fish independently chooses a movement direction and attempts to move (fish are blocked by walls). To minimize error, the learner must estimate the entire probability distribution over future states (i.e., every possible set of destination positions).

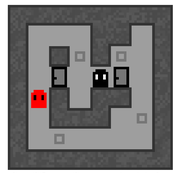
More domains are discussed in the paper.



(a) a level from the lights domain



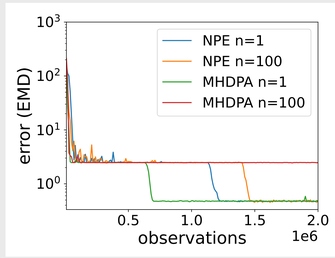
(b) a level from the fish domain



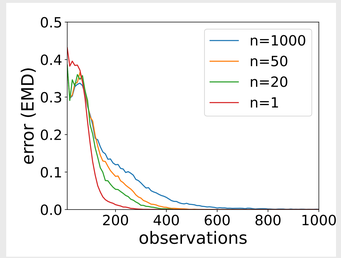
(c) a level from the $complex(4)$ domain

QORA learns over $1,000\times$ faster than neural-network baselines

In a relational domain, QORA reliably converges to *zero* error within one thousand steps, while both neural-network baselines fail to converge within two million steps.



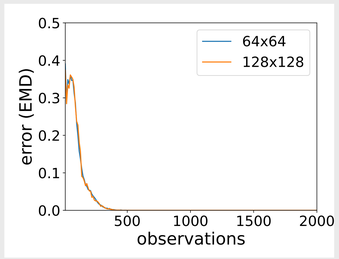
(a) neural-network baselines



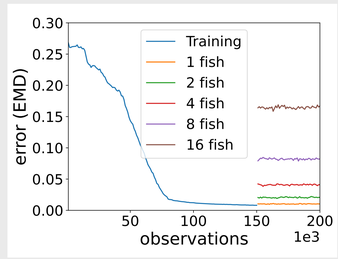
(b) QORA

QORA consistently demonstrates zero-shot transfer

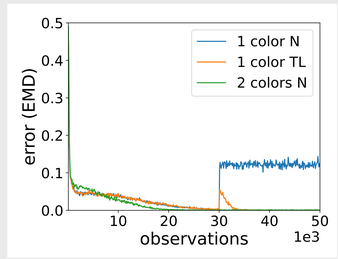
In deterministic environments, QORA is able to train in simple situations and perfectly predict outcomes in more-complex scenarios. In stochastic environments, prediction error scales optimally in the number of objects (i.e., linearly). When observing new interactions, QORA retains previously-learned partial rules, allowing for rapid adaptation.



(a) walls, transfer at 1k observations (deterministic)



(b) fish, transfer at 150k observations (stochastic)



(c) doors, transfer at 30k observations (deterministic)

QORA's learned models are easily interpretable

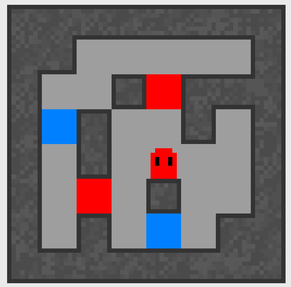
In this domain, the player character can only move if not blocked by a wall or a different-colored door.

$$r(o_1, s) = \begin{cases} (0,0) & P_1(o_1) \vee P_2(o_1) \\ (1,0) & \text{otherwise,} \end{cases}$$

where

$$P_1(o_1) \equiv \exists o_2 \in \textsf{s.walls}\colon o_2[pos] - o_1[pos] = (1,0)$$
$$P_2(o_1) \equiv \exists o_2 \in \textsf{s.doors}\colon o_2[pos] - o_1[pos] = (1,0) \wedge \neg(o_2[color] - o_1[color] = 0),$$

(a) one of QORA's learned rules in the doors domain



(b) a level from the doors domain; doors are denoted by the red and blue squares

How does QORA work?

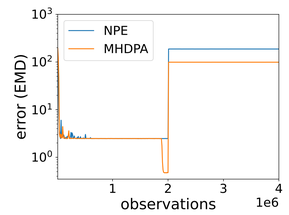
Key insights

- **Compositional framework**: our object-based representation is an effective way to *semantically* decompose states
- **Occam's razor**: we construct hypotheses iteratively to automatically scale the model's complexity to match the domain
- **Explicit modeling**: QORA learns explicit probability distributions by directly counting frequencies
- **Confidence levels**: we use predictive power and statistical significance measures to score and compare hypotheses
- **Simplicity**: QORA's single hyperparameter is easy to tune

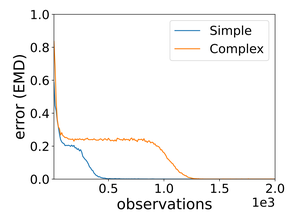
We introduce the following confidence-based score metric to evaluate hypotheses:

$$\mathcal{S}(r) = \sum_{(x,y)} \hat{P}(y|x)\hat{P}(x,y).$$

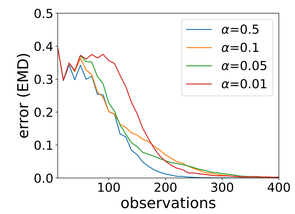
Additional Experiments



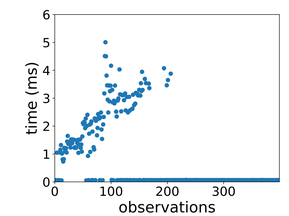
(a) neural networks fail to transfer even in relatively simple domains



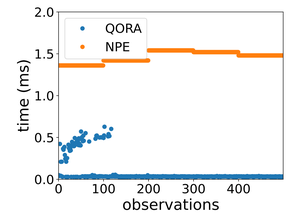
(b) QORA can be pre-trained in simpler scenarios to accelerate learning prior to transfer to more-complex tasks



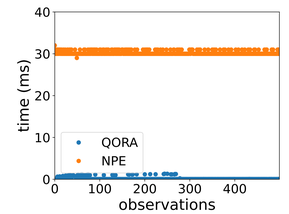
(c) QORA is robust to significant changes to its hyperparameter α



(d) once QORA's model converges, updates are nearly instantaneous



(e) QORA is several times faster than optimized neural networks, even when the networks use batching



(f) QORA is over an order of magnitude faster than neural networks if they don't use batching