Trackmania Fixed Gaussian Mixtures RL

- Formula-type game, purpose: minimize time till finish.
- Discrete inputs: `Gas` (2 values), `Brake` (2), `Steer` (3).

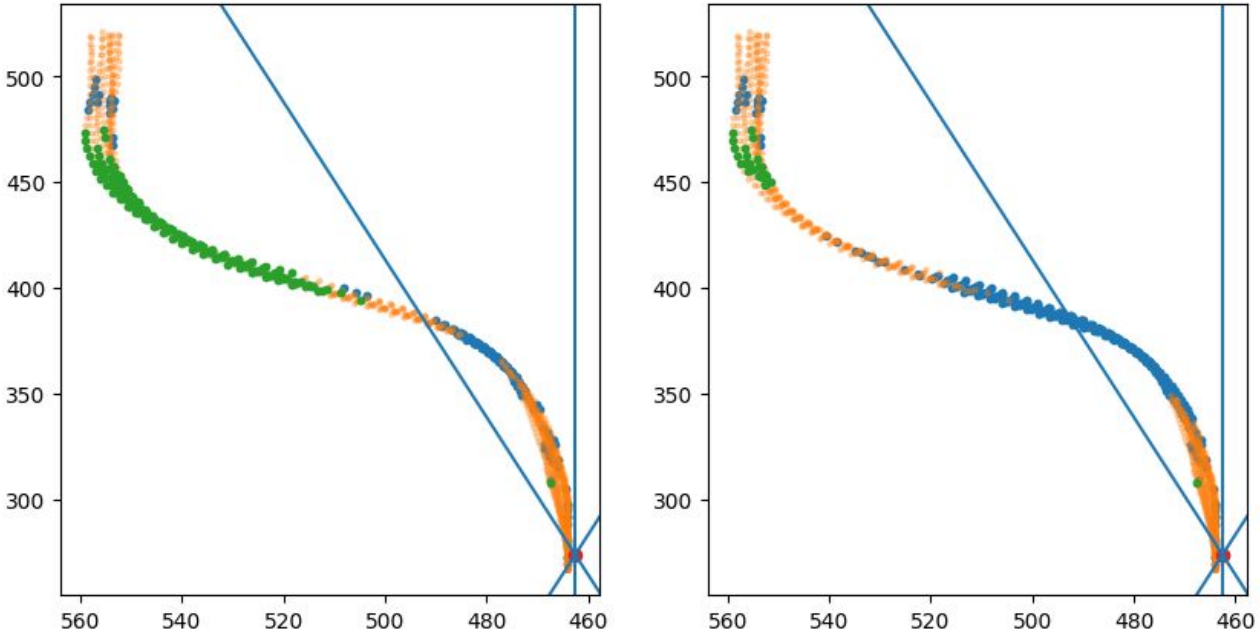
- Game pros: Very large dataset of well-driven replays on many maps.
- Game cons: Only API is reverse-engineered.

Hypothesis:

- People know the optimal racing line, but have trouble completely following it for the entirety of the track.
- Start with a large batch of top replays, and “stitch” / converge towards the best racing line.

Gaussian Mixtures

- Suppose that we know the latent Gaussians that generate the (state, action) space. They are the points from the best K replays on the map.
- Let the Gaussians have their mean fixed. We only modify their covariance and amplitude s.t. we best approximate the Q values of observed (s, a) tuples.



Gaussian Mixtures

- Approximate an observed Q(s, a) in function of latent Gaussians that share the action:

$$Q(s,a) = \mathbb{E}_{\substack{(f,a',\sigma^2,amp) \\ a'=a}} \left[\exp\left(\frac{|| \phi(s) - \phi(f) ||^2}{\sigma^2} + amp \right) \right]$$

- Our purpose is to minimize the batch TD error w.r.t. the latent Gaussians' covariances and amplitudes.

$$\min_{\sigma^2,amp} \mathbb{E}_{(s,a,r,s') \in \text{Batch}} \left[\gamma(r + \max_{a'} Q(s',a')) - Q(s,a) \right]^2$$

Representation problems

- Replay points are often bunched up on a dimension, so just one learnable std scalar per Gaussian isn't expressive enough:

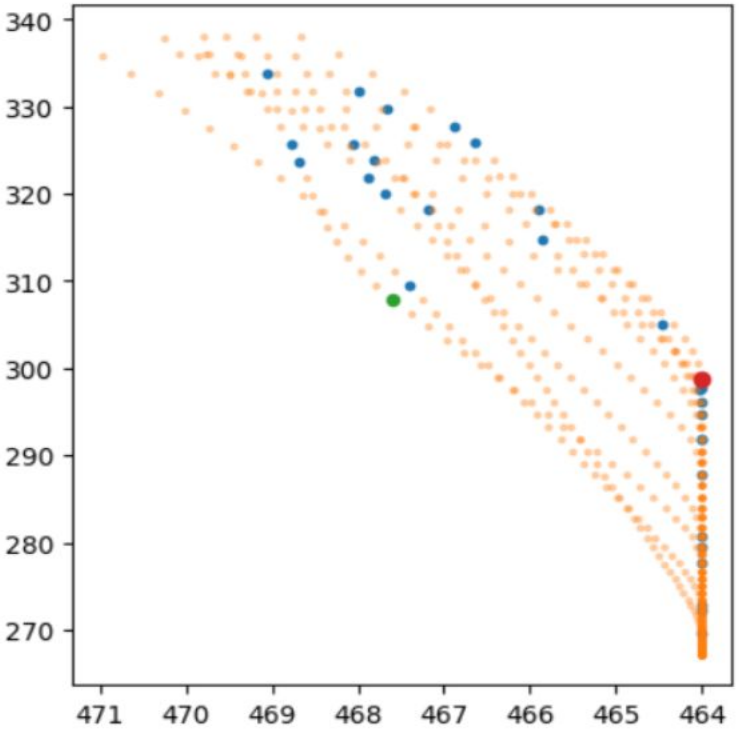
$$-\frac{||\, s_{xy} - f_{xy} \,||^2}{\sigma_{xy}^2} - \frac{||\, s_{yz} - f_{yz} \,||^2}{\sigma_{yz}^2} - \frac{||\, s_{xyz} - f_{xyz} \,||^2}{\sigma_{xyz}^2}$$

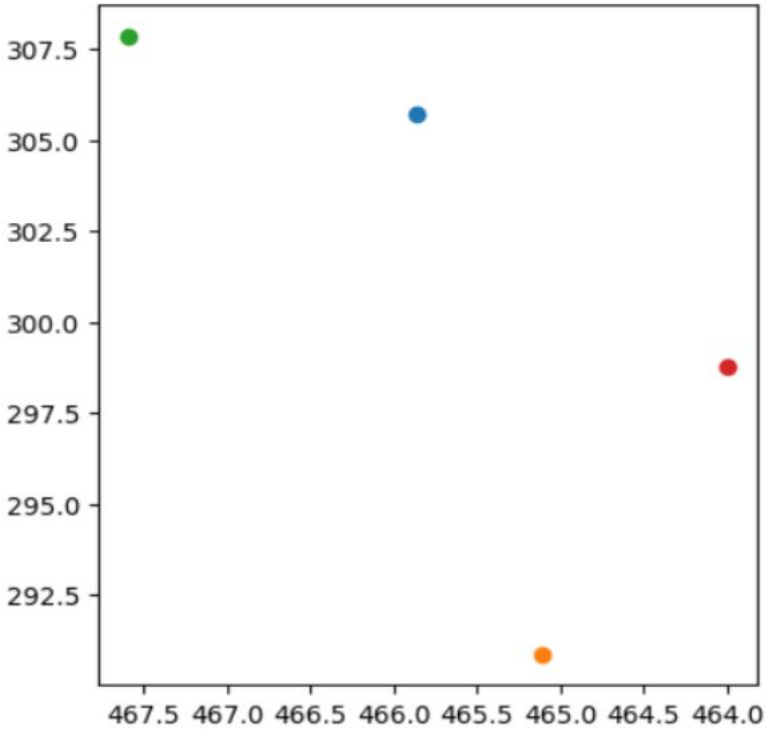
- Assuming that we should take an action just because fixed points in our vicinity take it is slightly optimistic.

- It's best to start with low stds for all fixed points and expand some, rather than start with higher stds and shrink most.

Representation problems

- Red point is our current location. Orange = STEER_FRONT, Blue = LEFT.
- We are so far ahead that almost all points should relatively show LEFT.





Fixed point’s relative action

There are enough ways to deal with this:

- Add an angle distance to the exponential term. In practice, this leaves very little points with enough influence over our current state.

- Dynamically change the fixed points’ actions if the angle distance is higher than a fixed angle (e.g. pi/6, pi/4). Used in practice, the most expensive part of the code.

- (best in hindsight) remove actions, and have each fixed point carry the orientation of the car over the next ticks. We compute an average orientation for an observed state.

Exploration policy

- For MBGD, sampling relative to the sum worked best, since the Q function estimation was visibly biased below the real value:

$$a \sim \frac{Q(s,a)}{\sum_{a'} Q(s,a')}$$

- Impulse-based descent reduced the bias significantly. Sum-sampling was too exploratory, and softmax converged too quickly. Used eps-greedy.

Reward function

- Passive component: the distance travelled between the last two frames, but only on XZ: Y is height, and we would occasionally get reward bursts for dropping off the road.

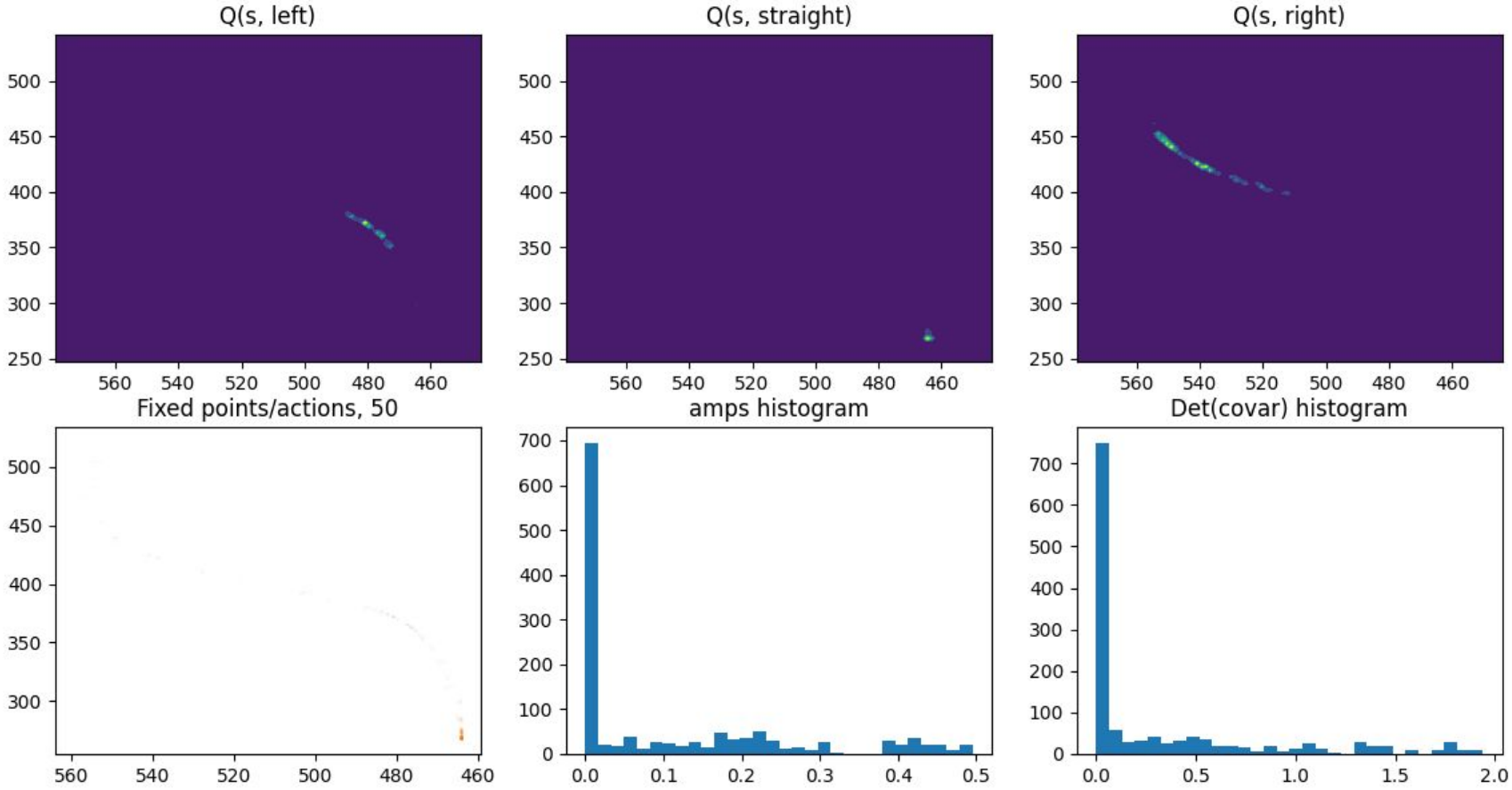
- Active component: reward if the agent reached a fixed point faster than a replay did. Need to constrain the agent angle as well as to not stumble upon a point.

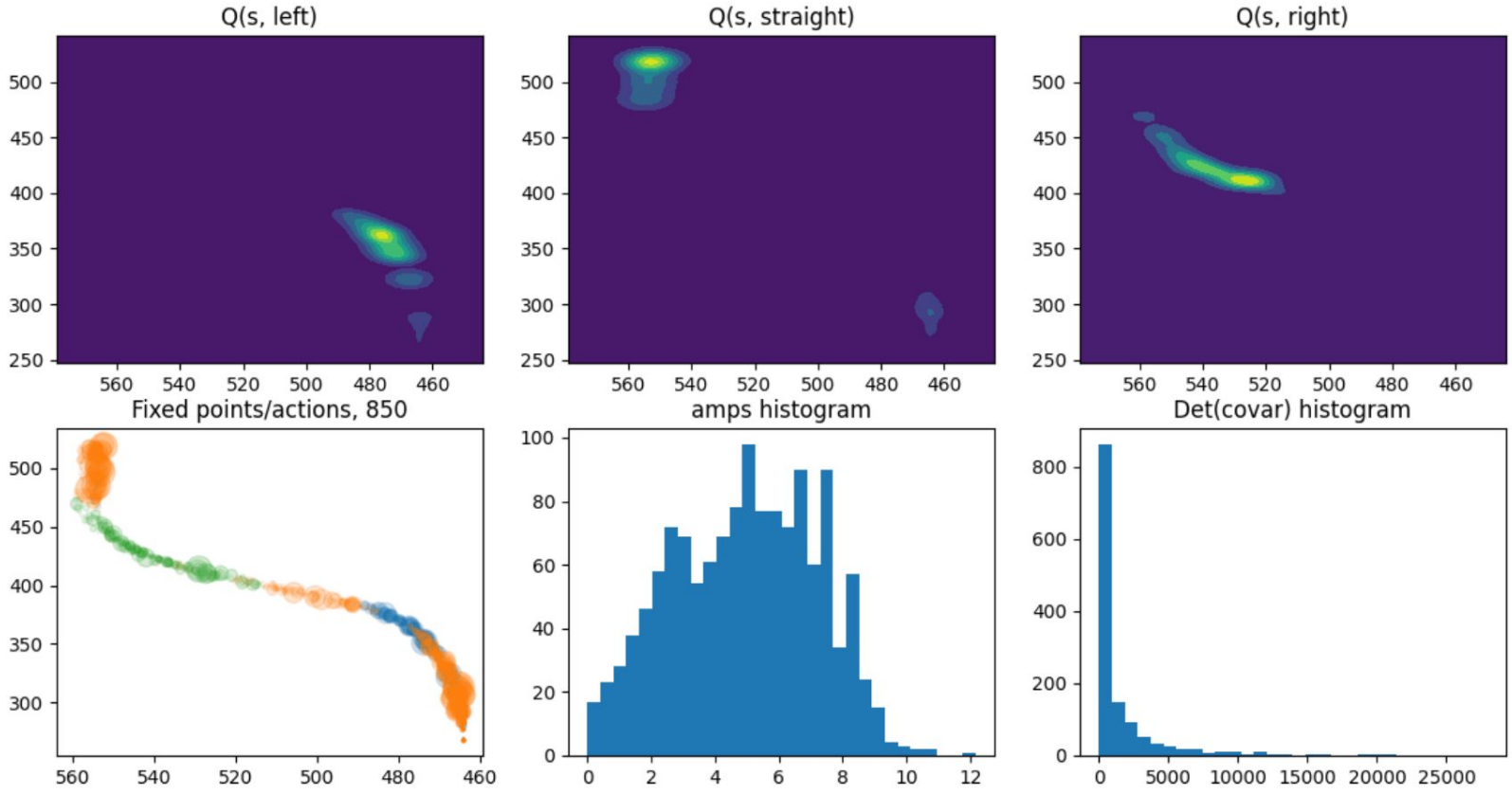
- Sparse component: finishing the map or passing a checkpoint.

Run on test map

- PB at 6s 57.
- Stable argmax at 6s 87 after 800 episodes.
- Best at 6s 62 after ~ 2700 episodes.

- We incrementally allow the agent to take actions more often as episodes progress, e.g. from 2 actions per second to ~7.


Q(s, left)
Q(s, straight)
Q(s, right)
Fixed points/actions, 50
amps histogram
Det(covar) histogram


Q(s, left)
Q(s, straight)
Q(s, right)
Fixed points/actions, 850
amps histogram
Det(covar) histogram

Others

Run on a more complicated map

- API crashes:-(
- Has relatively strong starting line. Any initial std is a very bad approximation.

(more) Deterministic exploratory policy

- Build a graph with nodes at replay intersections. Perform Dijkstra / sample from top K length paths.
- Angle/Distance threshold for intersection definition.