APPENDIX A
DERIVATION OF HESSIANS FROM THE BACKWARD PASS

PODDP computes a second-order approximation $\tilde{Q}$ to the Q-function defined in the main text, by taking first- and second-derivatives of the dynamics and cost functions with respect to perturbations of the belief state δs and controls δu. For reasons of space and brevity, we derive the Hessian matrices here.

Note that following the standard iLQR approach [3], we discard the terms involving Hessians of the dynamics $\frac{\partial^2 s'_z}{\partial \delta s^2}$, $\frac{\partial^2 s'_z}{\partial \delta s \partial \delta u}$, and $\frac{\partial^2 s'_z}{\partial \delta u^2}$.

The Hessians of the $\tilde{Q}$-function are:

$$Q_{ss} = \sum_{z \in \mathcal{Z}} \left[\frac{\partial^2 b_z}{\partial \delta s^2} \left(l_z + V'_z\right) + 2 \frac{\partial b_z}{\partial \delta s} \left(\frac{\partial l_z}{\partial \delta s} + \frac{\partial s'_z}{\partial \delta s}^\mathsf{T} \frac{\partial V'_z}{\partial s'_z} \right)^\mathsf{T} + b_z \left(\frac{\partial^2 l_z}{\partial \delta s^2} + \frac{\partial s'_z}{\partial \delta s}^\mathsf{T} \frac{\partial^2 V'_z}{\partial s'^2_z} \frac{\partial s'_z}{\partial \delta s} + \frac{\partial V'_z}{\partial s'_z}^\mathsf{T} \frac{\partial^2 s'_z}{\partial \delta s^2} \right) \right]$$

$$Q_{su} = \sum_{z \in \mathcal{Z}} \left[\frac{\partial b_z}{\partial \delta s} \left(\frac{\partial l_z}{\partial \delta u} + \frac{\partial s'_z}{\partial \delta u}^\mathsf{T} \frac{\partial V'_z}{\partial s'_z} \right)^\mathsf{T} + b_z \left(\frac{\partial^2 l_z}{\partial \delta s \partial \delta u} + \frac{\partial s'_z}{\partial \delta s}^\mathsf{T} \frac{\partial^2 V'_z}{\partial s'^2_z} \frac{\partial s'_z}{\partial \delta u} + \frac{\partial V'_z}{\partial s'_z}^\mathsf{T} \frac{\partial^2 s'_z}{\partial \delta s \partial \delta u} \right) \right]$$

$$Q_{uu} = \sum_{z \in \mathcal{Z}} \left[b_z \left(\frac{\partial^2 l_z}{\partial \delta u^2} + \frac{\partial s'_z}{\partial \delta u}^\mathsf{T} \frac{\partial^2 V'_z}{\partial s'^2_z} \frac{\partial s'_z}{\partial \delta u} + \frac{\partial V'_z}{\partial s'_z}^\mathsf{T} \frac{\partial^2 s'_z}{\partial \delta u^2} \right) \right]$$

APPENDIX B
DERIVATION OF VALUE FUNCTION RECURSION FROM THE BACKWARD PASS

We can plug δu^* from (8) in the main text back into $\tilde{Q}$, to calculate the approximate quadratic model of the value function V:

$$\begin{aligned} \Delta V &\approx -\frac{1}{2} k^\mathsf{T} Q_{uu} k \\ \frac{\partial V}{\partial s} &\approx Q_s - \frac{1}{2} K^\mathsf{T} Q_{uu} k \\ \frac{\partial^2 V}{\partial s^2} &\approx Q_{ss} - \frac{1}{2} K^\mathsf{T} Q_{uu} K. \end{aligned} \tag{10}$$

In Algorithm 2, the OPTIMIZECONTROL function returns $\Delta = \left\langle V, \frac{\partial V}{\partial s}, \frac{\partial^2 V}{\partial \delta s^2} \right\rangle$.

APPENDIX C
SUPPLEMENTARY EXPERIMENTAL METHODS AND RESULTS

A. Dynamic Bicycle Model

Agents in all experiments follow a 4-dimensional dynamic bicycle model. The state is defined as $[x, y, \phi, v]$, where x is the longitudinal position, y is the longitudinal position, ϕ is the orientation, and v is the velocity. The control is defined as $[\omega, a]$, where ω is the steering angle, and a is the longitudinal acceleration.

The dynamics are:

$$\begin{aligned} x(t+\delta t) &= x(t) + v\cos(\phi)\delta t + \epsilon_x \\ y(t+\delta t) &= y(t) + v\sin(\phi)\delta t + \epsilon_y \\ \phi(t+\delta t) &= \phi(t) + \frac{v}{L}\tan(\omega)\delta t + \epsilon_\phi \\ v(t+\delta t) &= v(t) + a\delta t + \epsilon_v, \end{aligned}$$

where L is the vehicle length, and ϵ_x, ϵ_y, ϵ_ϕ, and ϵ_v are additive Gaussian noise terms.

B. Experiment 1

1) Scenario formulation: In the T-Maze environment, the observability of the latent state improves as the agent moves longitudinally down the corridor. Given an uncertainty level ξ the standard derivation of the observation is formulated as

$$\sigma(y) = 0.1 + (s(-(y-c))/c) \cdot \xi$$

where y is the coordinate of the agent along the hallway, $c = 18.0$ defines the border of the region with noisy observations, and $s(x) = \left(\sqrt{x^2+1} + x\right)/2$. An observation o conditioned on the latent state z is then sampled from the observation function given by:

$$o \sim O(\cdot|z, y) = \begin{cases} \mathcal{N}(-1, \sigma(y)) \text{ if } z = \textit{Left} \\ \mathcal{N}(1, \sigma(y)) \text{ if } z = \textit{Right} \end{cases}$$

where $\mathcal{N}$ is a normal distribution function. The goal on the $Left$ is located at $(-25.0, 25.0)$ and the goal on the $Right$ is located at $(25.0, 25.0)$.

2) Model evaluation: To quantitatively evaluate the model, we ran 100 executions in thirteen different environments, each with a different level of observation uncertainty in figure 3(a). The observation uncertainty levels used were $[0.0, 1.0, 2.0, \ldots, 12.0]$. Each execution sampled the ground truth world state from the prior (set to $b_0(\textit{Left}) = 0.51$), and sampled observations from the observation distribution at each observation timestep. For this experiment, the timestep was $\delta t = 0.1$, and the planning horizon was $T = 60$. We set the number of segments for hierarchical PODDP to be $k = 3$; therefore there were two observations over the planning horizon, at $\tau_1 = 20$ and $\tau_2 = 40$. All algorithms replanned after each observation. The vehicle dynamics were assumed to be deterministic, because they were independent of the latent state value, and not relevant for the task.

3) Supplementary results: Fig. 6(a) and (b) show sampled trajectories from MLDDP and PWDDP, respectively. Both algorithms fail to explore as aggressively as PODDP, which accelerates more rapidly in the beginning of the trajectory to get as reliable an observation as possible. The possible MLDDP trajectories split at each observation point, depending on the maximum-likelihood belief state following the observation. These trajectories commit strongly to the maximum-likelihood goal location, and when a bad observation occurs, they are unable to recover. The PWDDP trajectories also commit early, based on minimizing the goal costs to both locations simultaneously, weighted by their probability. Only when the initial observation is very strong can PWDDP can commit strongly.

Algorithm 1: FORWARDPASS $(x_0, b_0, U_{\text{nom}}, S_{\text{nom}}, k, K, \alpha, \mathcal{Z}, T)$

$U \leftarrow [\,]$; // initialize control map indexed by histories
$S(\text{'Root'}) \leftarrow [\mathbf{x}_0, \mathbf{b}_0]$; // initialize belief state map indexed by histories
FORWARDTREE $(\text{'Root'}, U, S, U_{\text{nom}}, S_{\text{nom}}, \varnothing, \varnothing, \alpha, \mathcal{Z}, T, 1)$; // trajectory tree recursion
return U, S ; // return updated trajectory tree
Procedure FORWARDTREE$(H, U, S, U_{nom}, S_{nom}, k, K, \alpha, \mathcal{Z}, T, d)$
  if $k(H), K(H) \neq \varnothing$ **then** // apply control updates
    $U(H) \leftarrow U_{\text{nom}}(H) + \alpha k(H) + K(H)(S(H) - S_{\text{nom}}(H))$;
  else
    $U(H) \leftarrow U_{\text{nom}}(H)$;
  for $z \in \mathcal{Z}$ **do**
    $[\mathbf{x}_H, \mathbf{b}_H] \leftarrow S(H)$;
    $\mathbf{x}_{ML} = \arg\max_{\mathbf{x}} p(\mathbf{x}|\mathbf{x}_H, U(H), z)$; // assume ML state transition
    $\mathbf{o}_{ML} = \arg\max_{\mathbf{o}} p(\mathbf{o}|\mathbf{x}_{ML}, z)$; // assume ML observation
    $\mathbf{b}' = \text{BELIEFUPDATE}(\mathbf{o}_{ML}, \mathbf{x}_{ML}, U(H), \mathbf{x}_H, \mathbf{b}_H)$;
    $S([H, z]) \leftarrow [\mathbf{x}_{ML}, \mathbf{b}']$; // append new belief state to history
    if $d < T$ **then**
      FORWARDTREE $([H, z], U, S, U_{\text{nom}}, X_{\text{nom}}, k, K, \alpha, \mathcal{Z}, T, d+1)$; // recurse

Algorithm 2: BACKWARDPASS $(U, S, \mathcal{Z}, T)$

$k, K \leftarrow [\,]$; // initialize control update maps indexed by histories
BACKWARDTREE $(\text{'Root'}, k, K, U, S, \mathcal{Z}, T, 1)$; // compute control updates recursively
return k, K ; // return control updates
Procedure BACKWARDTREE$(H, k, K, u, S, \mathcal{Z}, T, d)$
  $\Delta \leftarrow [\,]$; // initialize backward derivatives map
  for $z \in \mathcal{Z}$ **do**
    if $d < T$ **then** // recursively compute backward derivatives and updates
      $\Delta(z) \leftarrow \text{BACKWARDTREE}([H, z], k, K, U, S, \mathcal{Z}, T, d)$;
    else // set backward derivatives as empty
      $\Delta(z) \leftarrow \varnothing$;
  $k_H, K_h, \Delta_H \leftarrow \text{OPTIMIZECONTROL}(U(H), S(H), \Delta, \mathcal{Z})$;
  $k(H), K(H) \leftarrow k_H, K_H$; // update control update maps
  return Δ_H ; // return backward derivatives

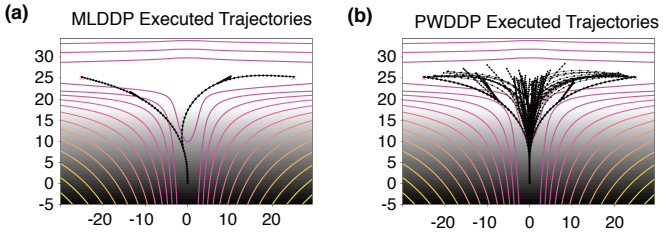

Fig. 6: Experiment 1 results. (a) 100 sampled MLDDP executions. (b) 100 sampled PWDDP executions. Compare to Main Text, Fig. 3(c).

C. Experiment 2

1) Scenario formulation: To capture the rough terrain in this experiment, we assume that the terrain exerts a location dependent resistive force on the vehicle. We model this force as a velocity-dependent deceleration r, such that:

$$r = \rho * \tanh(v),$$

which the vehicle must overcome with its own acceleration, incurring cost. The parameter ρ varies according to the location. In our scenario, ρ is high for rough terrain, and zero for smooth terrain. The transition in our environment is sigmoidal in the x-dimension.

For this experiment, the timestep was $\delta t = 0.1$, and the planning horizon was $T = 60$. We set the number of segments for hierarchical PODDP to be $k = 3$; therefore there were two observations over the planning horizon, at $\tau_1 = 20$ and $\tau_2 = 40$.

In this scenario, because all information about the latent state comes from the state transitions, we assume additive

Gaussian noise at each timestep for all state variables to make the belief state dynamics nontrivial.

2) Supplementary results: Fig. 7(a) and (b) show sampled trajectories from MLDDP and PWDDP, respectively. Both algorithms fail to explore as aggressively as PODDP, which veers left into the potentially smooth area to obtain as reliable an observation as possible. MLDDP initially assumes that $z =$ *Rough*, and heads directly for the goal location. If the maximum-likelihood belief after the first observation is $z =$ *Smooth*, it moves into the smooth region. PWDDP actually moves away from the smooth region, because this trajectory allows it to minimize distance from the goal with the same control sequence under both dynamics. This is an interesting and subtle feature of PWDDP. It also moves into the smooth region if the belief updates in favor of $z =$ *Smooth*, but it does not explore or plan for future observation contingencies.

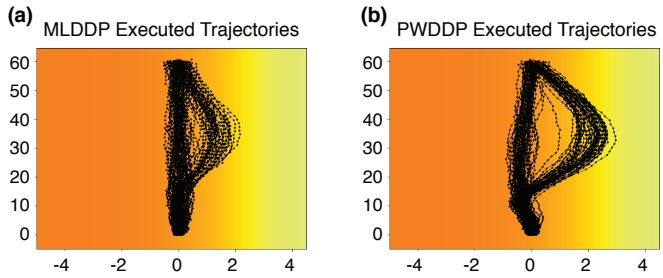

Fig. 7: Experiment 2 results. (a) 100 sampled MLDDP executions. (b) 100 sampled PWDDP executions. Compare to Main Text, Fig. 4(b).

D. Experiment 3

1) Scenario formulation: In this scenario, we assume that both vehicles start from a velocity of 10 m/s. The dynamics of the other vehicle follows a modified IDM, which shares the same acceleration control principle of the standard IDM, while considering the leading vehicle on an adjacent lane by modifying the distance metric between the agent and the leading vehicle from $s(l, a) = |y_l - y_a|$ to

$$s'(l, a) = \sigma_{c,\delta}\left(|x_l - x_a|\right) \cdot s(l, a)$$

where x is the coordinate perpendicular the the lanes and y is the coordinate along the lanes with l indicating the leading vehicle and a indicating the IDM agent, and sigmoid function $\sigma_{c,\delta}(x) = \left(1 + \exp^{-\delta(x-c)}\right)^{-1}$. For the sigmoid function σ, parameter c is assigned with the value of half the lane width, and δ is an adjustable steepness parameter.

For this experiment, the timestep was $\delta t = 0.1$, and the planning horizon was $T = 30$. We set the number of segments for hierarchical PODDP to be $k = 3$; therefore there were two observations over the planning horizon, at $\tau_1 = 10$ and $\tau_2 = 20$.

In this scenario, because all information about the latent state comes from the other vehicle's state transitions, we assume additive Gaussian noise at each timestep for all state variables to make the belief state dynamics nontrivial.

2) Supplementary results: Please see our GitHub page (https://davidqiu1993.github.io/poddp-paper) for supplementary videos.