APPENDIX

A. Variational Inference for Finite-Horizon Stochastic Optimal Control

The variational posterior over trajectories is defined by the dynamics and the variational posterior over actions:

$$\begin{aligned} q(\tau|x_0) &= q(X,U|x_0) \\ &= p(X|U,x_0)q(U) \\ &= q(U)\prod_{t=0}^{T} p(x_{t+1}|x_t,u_t) \end{aligned} \tag{15}$$

We will omit the dependence on the initial state x_0 for convenience.

$$\begin{aligned} &\mathcal{KL}\left(q(\tau)||p(\tau|o=1)\right) = \int q(\tau)\log\frac{q(\tau)}{p(\tau|o=1)}d\tau \\ &= \int q(X,U)\log\frac{p(X|U)q(U)p(o=1)}{p(o=1|X,U)p(X|U)p(U)}dX\,dU \end{aligned} \tag{16}$$

Since $p(o=1)$ on the numerator does not depend on U, when we minimize the above divergence it can be dropped. The result is minimizing the below quantity, the *variational free energy* $\mathcal{F}$.

$$\mathcal{F} = \int q(X,U)\log\frac{q(U)}{p(o=1|X,U)p(U)}dX\,dU \tag{17}$$

$$= -\mathbb{E}_{q(X,U)}\left[\log p(o|X,U) + \log p(U) - \log q(U)\right] \tag{18}$$

$$= \mathbb{E}_{q(X,U)}\left[J(X,U)\right] + \mathcal{KL}(q(U)||p(U)) \tag{19}$$

$$= \mathbb{E}_{q(X,U)}\left[\hat{J}(X,U) + \log q(U)\right] \tag{20}$$

For the last two expressions we have used our formulation that the $p(o=1|X,U) = \exp(-J(X,U))$, where J is the trajectory cost, and we have incorporated the deviation from the prior into the cost function. For example, a zero-mean Gaussian prior on the controls can be equivalently expressed as a squared cost on the magnitude of the controls.

B. Training & Architecture Details

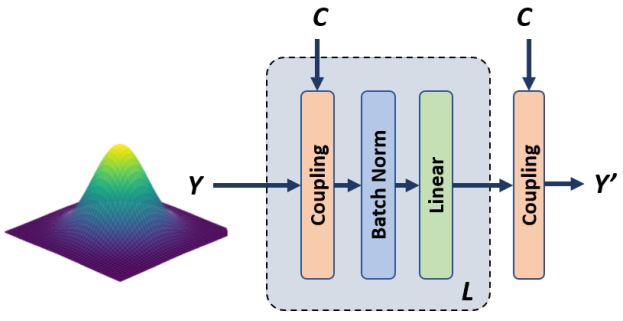

Fig. 6. The architecture for both the prior flow and the control sequence posterior flow, based on [8] and [37], showing a mapping from arbitrary Y to Y'. Each flow consists of L chained transformation blocks. A transformation block consists of a conditional coupling layer, a batch norm layer, and a linear layer. There is a final conditional coupling layer on the output. For the vae prior, there is no context therefore we use standard coupling layers and not conditional coupling layers.

Variable	Planar Navigation	3D 12DoF Quadrotor
control peturbation Σ_e	$1 - \frac{\text{epoch}}{\#\text{epochs}}$	
α	$500\frac{\text{epoch}}{\#\text{epochs}}$	
β	1	1
# epochs	1000	1000
Initial learning rate	1×10^{-3}	1×10^{-3}
Learning rate decay	0.9 every 50 epochs	
# Training environments	10000	20000
# (x_0, x_G) per training env.	100	100
h dim	64	256
a	5	5
b	$\frac{1}{64}$	$\frac{1}{1024}$
VAE training epochs	100	100
$p_\phi(h)$ flow depth L	4	4
f_ζ flow depth L	10	10

TABLE III
TRAINING AND ARCHITECTURE HYPERPARAMETERS FOR EACH EXPERIMENT.

1) Hyperparameter Tuning: There are several hyperparameters to tune in our approach. The scalar a in equation 11 was tuned so that $a\mathcal{L}_{VAE}$ and $\mathcal{L}_{flow}$ were of approximately similar magnitude. The scalar b in equation 14 was selected to be equal to the dimensionality of the SDF observation divided by the dimensionality of the latent environment embedding. This value was chosen initially to make the projection loss similar across the quadcopter and the double integrator, and we found this automatic tuning worked well in practice. Hyperparameters α, β together control the trade-off between entropy and optimality. We kept β fixed and tuned only α. To tune α, for each experiment we performed a grid search and selected the value of α that resulted in the best performance in the training environment when used with FlowMPPI.

C. Environment details

The environments are $4m \times 4m$, and generated as occupancy grids, from which we compute the SDF. For each training environment, we randomly sample 100 start & goal pairs such that they are always collision free, and within the bounds of the voxel grid. We sample start velocities from a Normal distribution, and set the goal velocity to be zero. During evaluation, for both the in-distribution and out-of-distribution environments, we sample 100 start, goal and environment tuples and evaluate all methods on these tuples. The exception to this is the real-world environments, where we keep the environments fixed and sample 100 start and goal pairs per real-world environment and evaluate all methods on these pairs. To ensure the navigation problem is non-trivial, we sample starts and goals that are at least $4m$ away.

1) Real-world environments: The two real world environments are taken from area 3 from the 2D-3D-S dataset [1]. To generate the two environments, we used the 3D mesh from the dataset and defined a subset of the area to be the environment. We then generated an occupancy grid by densely sampling the mesh, which we then used to compute the SDF.

2) Planar Navigation: The dynamics for the planar navigation system are

$$\begin{bmatrix} x \\ y \\ \dot{x} \\ \dot{y} \end{bmatrix}_{t+1} = \begin{bmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & 0.95 & 0 \\ 0 & 0 & 0 & 0.95 \end{bmatrix} \begin{bmatrix} x \\ y \\ \dot{x} \\ \dot{y} \end{bmatrix}_t + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ \Delta t & 0 \\ 0 & \Delta t \end{bmatrix} \mathbf{u} \quad (21)$$

3) 12DoF Quadrotor: The dynamics for the 12DoF quadrotor are from Sabatino [29] and are given by

$$\begin{bmatrix} x \\ y \\ z \\ p \\ q \\ r \\ \dot{x} \\ \dot{y} \\ \dot{z} \\ \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix}_{t+1} = \begin{bmatrix} x \\ y \\ z \\ p \\ q \\ r \\ \dot{x} \\ \dot{y} \\ \dot{z} \\ \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix}_t + \Delta t \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \dot{p} + \dot{q}s(p)t(q) + \dot{r}c(p)t(q) \\ \dot{q}c(p) - \dot{r}s\dot{p} \\ \dot{q}\frac{s(p)}{c(q)} + \dot{r}\frac{c(p))}{c(q)} \\ -(s(p)s(r) + c(r)c(p)s(q))K\frac{u_1}{m} \\ -(c(r)s(p) - c(p)s(r)s(q))K\frac{u_1}{m} \\ g - c(p)s(q))K\frac{u_1}{m} \\ \frac{(I_y - I_z)\dot{q}\dot{r} + Ku_2}{I_x} \\ \frac{(I_z - I_x)\dot{p}\dot{r} + Ku_3}{I_y} \\ \frac{(I_x - I_y)\dot{p}\dot{q} + Ku_4}{I_z} \end{bmatrix}_t \quad (22)$$

Where $c(p), s(p), t(p)$ are $\cos, \sin, \tan$ functions respectively. We use a parameters $m = 1, I_x = 0.5, I_y = 0.1, I_z = 0.3, K = 5, g = -9.81$. The quadrotor geometry is modeled as a cylinder with radius $0.1m$ and height $0.05m$.

D. Controller details

Variable	Planar Navigation	12DoF Quadrotor
Control Horizon H	40	40
Trial length T	100	100
Control prior σ	1	4
Dynamics Δt	0.05	0.025

TABLE IV
CONTROLLER AGNOSTIC PARAMETERS USED FOR THE EVALUATIONS

.

Controller	Variable	Planar Navigation	12DoF Quadrotor
MPPI	λ	1	1
	Σ	0.9	0.5
	iterations	1	4
SVMPC	Σ	1	0.5
	# particles	4	4
	Learning rate	1	0.5
	iterations	4	4
	warm-up iterations	25	25
iCEM	Σ	0.75	0.5
	noise parameter	2.5	3
	% elites	0.1	0.1
	% kept elites	0.3	0.5
	iterations	4	4
	momentum	0.1	0.1
FlowMPPI	λ	1	1
	Σ	1	0.75
	iterations	1	2
	M	10	10
	Proj. learn. rate	1×10^{-2}	1×10^{-2}

TABLE V
CONTROLLER HYPERPARAMETERS USED FOR THE EXPERIMENTS FOR BOTH OUR PROPOSED METHOD AND THE BASELINES.

E. Algorithms

Algorithm 2 Sample from Control Sequence Posterior with Perturbation

function SAMPLEPERTU(C, Σ_ϵ, K)
  for i $\in \{k, ..., K\}$ **do**
    $Z_k \sim \mathcal{N}(0, I)$
    $\epsilon_k \sim \mathcal{N}(0, \Sigma_\epsilon)$
    $U_k \leftarrow f_\zeta(Z_k, C) + \epsilon_k$
    $\hat{Z}_k \leftarrow f_\zeta^{-1}(U_k, C)$
    $q_\zeta(U_k|C) \leftarrow$ from $\hat{Z}_k$ via eq. (5)
  return $\{U_k, q_\zeta(U_k|C)\}_{k=1}^K$

Algorithm 3 Flow Training

Inputs: N iterations, K samples, $\Theta^1 = \{\theta^1, \psi^1, \phi^1, \omega^1, \zeta^1\}$ initial parameters, control perturbation covariance Σ_ϵ, learning rate η, loss hyperparameters (α, β)

for n $\in \{1, ..., N\}$ **do**
  $h \leftarrow q_\theta(h|E)$
  $\hat{E} \leftarrow p_\psi(E|h)$
  Compute $\log p_\phi(h)$ via eq. (5)
  Compute $\mathcal{L}_{VAE}$
  $C \leftarrow g_\omega(x_0, x_G, h)$
  $\{U_k, q_\zeta(U_k|C)\}_{k=1}^K \leftarrow$ SAMPLEPERTU(C, Σ_ϵ, K)
  $\mathcal{L} \leftarrow \mathcal{L}_{VAE}$
  for k $\in \{1, ..., K\}$ **do**
    $w_k \leftarrow$ from $(\{U_i, \log q_\zeta(U_i|C)\}_{i=1}^K, \alpha, \beta\})$ via (9)
    $\mathcal{L} \leftarrow \mathcal{L} - w_k \cdot \log q_\zeta(U_k|C)$
  $\Theta^{n+1} \leftarrow \Theta^n - \eta \frac{\partial \mathcal{L}}{\partial \Theta}$

Algorithm 4 Projection

Inputs: N iterations, K samples, $\theta, \phi, \omega, \zeta$ parameters, control perturbation covariance Σ_ϵ, learning rate η, loss hyperparameters (α, β)

$h^1 \leftarrow q_\theta(h|E)$
for n $\in \{1, ..., N\}$ **do**
  Compute $\log p_\phi(h^n)$ via eq. (5)
  $C \leftarrow g_\omega(x_0, x_G, h^n)$
  $\{U_k, q_\zeta(U_k|C)\}_{k=1}^K \leftarrow$ SAMPLEPERTU(C, Σ_ϵ, K)
  $\mathcal{L} \leftarrow -p_\phi(h^n)$
  for k $\in \{1, ..., K\}$ **do**
    $w_k \leftarrow$ from $(\{U_i, \log q_\zeta(U_i|C)\}_{i=1}^K, \alpha, \beta\})$ via (9)
    $\mathcal{L} \leftarrow \mathcal{L} - w_k \cdot \log q_\zeta(U_k|C)$
  $h^{n+1} \leftarrow h^n - \eta \frac{\partial \mathcal{L}}{\partial h}$

F. Additional Results

	K=256		K=512		K=1024	
Projection loss	Success	Cost	Success	Cost	Success	Cost
$\mathcal{L}_{OOD} + \mathcal{L}_{flow}$	**0.71**	**3688**	**0.83**	**3443**	**0.93**	**3200**
$\mathcal{L}_{OOD}$	0.52	3859	0.63	3704	0.89	3371
$\mathcal{L}_{flow}$	0.6	3758	0.72	3489	0.87	3226

TABLE VI
ABLATION OF THE DIFFERENT LOSS TERMS IN FLOWMPPIPROJECT FOR DIFFERENT SAMPLING BUDGETS FOR THE 12DOF QUADROTOR OUT-OF-DISTRIBUTION ENVIRONMENT