Improving Sample-based MPC with Normalizing Flows & Out-of-distribution Projection

Thomas Power[1] and Dmitry Berenson[1]

***Abstract*— We propose a sample-based Model Predictive Control (MPC) method for collision-free navigation that uses a conditional normalizing flow as a sampling distribution, conditioned on the start, goal and environment. This representation allows us to learn a distribution that accounts for both the dynamics of the robot and complex obstacle geometries. We can then sample from this distribution to produce control sequences which are likely to be both goal-directed and collision-free, and use this sampling distribution to improve the performance of sample-based MPC controllers. The representation of the environment is important, since when deploying this method, the robot may encounter an out-of-distribution (OOD) environment, i.e. one which is radically different from those used in training. In such cases, the learned flow cannot be trusted to produce low-cost control sequences. To generalize our method to OOD environments we also present an approach for learning the environment representation that enables us to perform *projection* on this representation as part of the MPC process. This projection changes the environment representation to be more in-distribution while also optimizing trajectory quality in the true environment. We demonstrate our approach for control of a 7DoF manipulator in several simulated environments, as well as one environment generated from real-world data.**

I. INTRODUCTION

Sample-based approaches for MPC such as the Cross Entropy Method (CEM) and Model Predictive Path Integral Control (MPPI) [1], [2], [3] have proven popular in robotics due to their ability to handle uncertainty, their minimal assumptions on the dynamics and cost function, and their parallelizable sampling. However, these methods struggle when randomly-sampling low-cost control sequences is unlikely and can become stuck in local minima, for example when a robot must find a path through a cluttered environment.

Several recent papers have considered the finite-horizon Stochastic Optimal Control (SOC) problem as Bayesian inference, and proposed methods of performing variational inference to approximate the distribution used to sample control sequences with a mixture-of-Gaussians [4] and a particle representation [5], [6]. These distributions are initially uninformed and must be iteratively improved during deployment. Instead, our proposed method uses a normalizing flow to represent this distribution, and we learn the parameters for this model from data. We use the learned distribution to improve two sampling-based MPC controllers; iCEM [2] and MPPI [3].

To generalize our method to novel environments, we present an approach that performs *projection* on the representation of the environment as part of the MPC process. This projection changes the environment representation to be more in-distribution while also optimizing trajectory quality in the true environment. In essence, this method "hallucinates" an environment that is more familiar to the normalizing flow so that the flow produces reliable results. However, the key insight behind our projection method is that the "hallucinated" environment cannot be arbitrary, it should be constrained to preserve important features of the true environment for the MPC problem at hand.

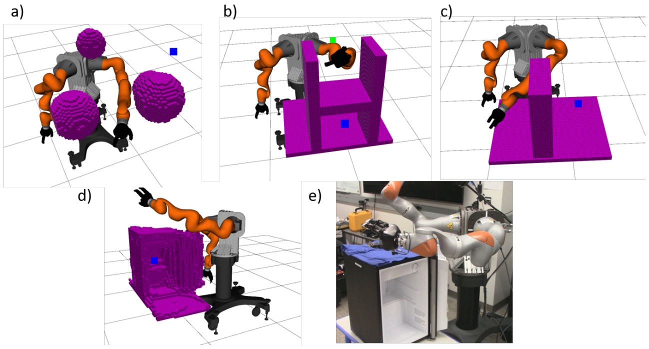

Fig. 1. We evaluate our approach on control of a kinematic 7DoF manipulator on four environments in simulation (a-d). Tasks consist of a) Navigating around spherical obstacles b) Reaching into a shelf c) Going from one side of a wall to another d) Reaching inside a fridge e) Real world setup for the reaching into a fridge task. The voxel grid in d) was generated from the fridge in e) using multiple views of a Kinect v2.

II. PROBLEM STATEMENT

We consider a discrete-time system with state $x \in \mathbb{R}^{d_x}$ and control $u \in \mathbb{R}^{d_u}$ and known transition probability $p(x_{t+1}|x_t, u_t)$. We define trajectories with horizon T as $\tau = (X, U)$, where $X = \{x_0, x_1, ...x_T\}$ and $U = \{u_0, u_1, ...u_{T-1}\}$.

Given an initial state x_0, a goal state x_G, and a signed-distance field (SDF) of the the environment E, our objective is to find U which minimizes the expected cost $E_{p(X|U)}[J(\tau)]$ for a given cost function J.

We reformulate SOC as an inference problem (as in [7], [8], [4], [5]). First, we introduce a binary 'optimality' random variable o for a trajectory such that $p(o=1|\tau) \propto \exp(-J(\tau))$ and aim to compute the posterior $p(\tau|o=1) \propto p(o=1|\tau)p(\tau)$, where $p(\tau)$ is a prior on trajectories induced by a prior on controls.

In general, this posterior is intractable, so we use variational inference to approximate it with tractable $q(\tau)$ which minimizes the KL-divergence $KL(q(\tau)||p(\tau|o=1))$ [9]. The variational posterior factorizes as $p(X|U)q(U)$, and thus our goal is to compute the approximate posterior over control sequences $q(U)$.

[1] Authors are with the University of Michigan, Ann Arbor, MI, USA. {tpower, dmitryb}@umich.edu

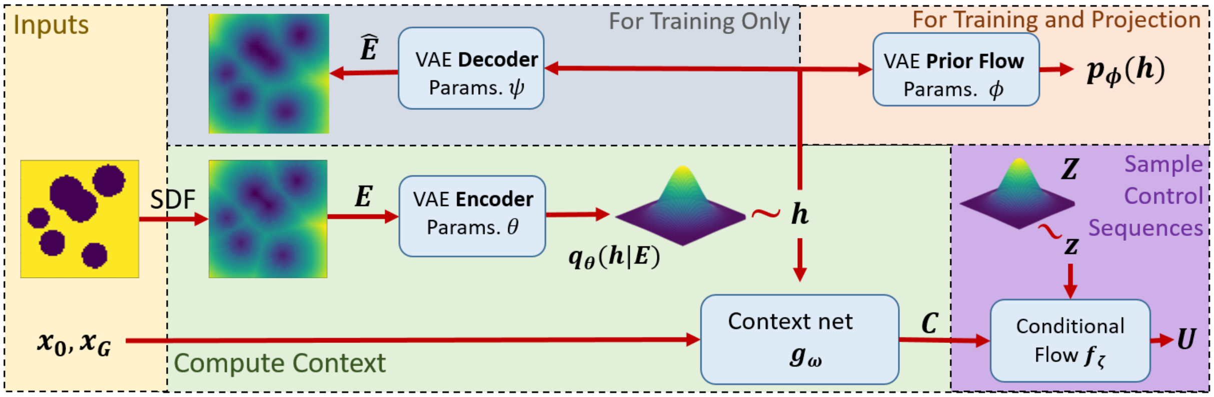

Fig. 2. The architecture of our method for sampling control sequences. We take as input initial and goal states x_0, x_G, and the environment, converted to a signed distance field E. E is input into a VAE to produce a latent distribution $q_\theta(h|E)$, which we sample to get the environment embedding h. This h is used, along with x_0 and x_G as input to the network g_ω to produce a context vector C. C, along with a sample from a Gaussian distribution Z, is input into the conditional normalizing flow f_ζ to produce a control sequence U. During training only, we use a decoder to reconstruct the SDF from h as part of the loss. We also use a normalizing flow prior for the VAE to compute an OOD score for a given h, which is necessary to perform projection.

III. METHOD

Our method uses amortized variational inference to learn a conditional control sequence posterior $q(U|x_0, x_g, E)$ given a dataset $\mathscr{D} = \{E, x_0, x_G\}^N$. Our proposed architecture for $q(U|x_0, x_g, E)$ is shown in Figure 2. We represent and learn $q(U|x_0, x_g, E)$ as a conditional normalizing flow [10], and use the resulting distribution as a sampling distribution for control sequences. Our method uses a VAE to encode an environment SDF into an environment embedding h. This environment embedding is used along with the start and goal as the input to a conditional normalizing flow, which samples control sequences that are goal-directed and avoid collision. The latent space prior of the VAE is used as a differentiable OOD score, which we use to adapt the environment embedding of novel environments online to *project* novel environments 'in-distribution' via:

$$\hat{h} = \arg\min_{h} \mathscr{L}_{trajectory} - \log p_\phi(h) \tag{1}$$

Where $p_\phi(h)$ is the latent prior of the VAE, and $\mathscr{L}_{trajectory}$ is a cost on the quality of trajectories sampled using a given environment embedding, when evaluated on the true environment SDF. Adding this trajectory cost regularizes the OOD projection, so that the projected environment must preserve features of the true environment which are relevant for the purpose of sampling low-cost trajectories. We use the $q(U|x_0, x_g, E)$ as a sampling distribution to improve two sample-based MPC controllers, MPPI [3] and iCEM [2]. Both of these algorithms use iteratively improved Normal distributions as the sampling distribution, we augment both by additionally sampling actions from the normalizing flow. When using projection, we iteratively perform projection with one gradient step per MPC timestep.

IV. EXPERIMENTS

We perform an experiment for controlling a kinematic 7DoF manipulator shown for different environments in Figure 1. We train the entire pipeline on a dataset generated from the environment with spherical obstacles, shown in Figure 1 a). The number and size of the spherical obstacles is randomized during training.

We evaluate on several novel environments shown in Figure 1 (b-d). We evaluate on one environment generated from real-world data, shown in 1 (d,e). The results across these environments for our proposed methods are shown in table I. Our results demonstrate that our incorporating the normalizing flow learned sampling distribution improves both MPPI and iCEM, and that by performing the online projection we are able to generalize to novel environments, including real-world environments. In particular, we see that the best performing algorithms across all evaluated environments are methods which use our proposed normalizing flow and OOD projection. A video demonstrating the proposed methods applied to a real robot are shown in https://youtu.be/owVIj6rWLLM.

For future work, we plan to extend this framework beyond collision-free navigation to tasks which necessitate interacting with the environment, such as object manipulation.

	In-Distribution		Out-of Distribution					
	Spheres Environment		**Shelf Environment**		**Wall Environment**		**Fridge Environment**	
Method	Success	Cost	Success	Cost	Success	Cost	Success	Cost
MPPI [3]	0.83	836	0.24	1900	0.12	1938	0.16	1944
SVMPC [5]	0.82	737	0.08	2132	0.42	1628	0.44	1946
iCEM [2]	0.85	694	0.66	1302	0.36	1768	0.89	898
FlowMPPI	0.85	698	0.65	1355	0.62	1280	0.74	1080
FlowiCEM	0.86	628	0.62	1339	0.44	1573	0.94	850
FlowMPPIProject	**0.87**	**582**	**0.75**	**1127**	0.64	1178	0.83	819
FlowiCEMProject	0.86	612	0.66	1268	**0.7**	**1109**	**0.97**	**798**

TABLE I

RESULTS FOR ATTEMPTING TASK 100 TIMES FOR EACH ENVIRONMENT IN SIMULATION. THE ENVIRONMENTS ARE SHOWN IN FIGURE 1. THE FRIDGE ENVIRONMENT IS GENERATED FROM REAL-WORLD DATA FROM THE FRIDGE SHOWN IN FIGURE 1 (E)

.

References

[1] M. Kobilarov, "Cross-entropy motion planning," *IJRR*, 2012.

[2] C. Pinneri, S. Sawant, S. Blaes, J. Achterhold, J. Stueckler, M. Rolinek, and G. Martius, "Sample-efficient cross-entropy method for real-time planning," in *CoRL*, 2020.

[3] G. Williams, P. Drews, B. Goldfain, J. M. Rehg, and E. A. Theodorou, "Information-Theoretic Model Predictive Control: Theory and Applications to Autonomous Driving," *IEEE Trans. Robot.*, 2018.

[4] M. Okada and T. Taniguchi, "Variational Inference MPC for Bayesian Model-based Reinforcement Learning," in *CoRL*, 2020.

[5] A. Lambert, A. Fishman, D. Fox, B. Boots, and F. Ramos, "Stein Variational Model Predictive Control," in *CoRL*, 2020.

[6] L. Barcelos, A. Lambert, R. Oliveira, P. Borges, B. Boots, and F. Ramos, "Dual Online Stein Variational Inference for Control and Dynamics," in *RSS*, 2021.

[7] K. Rawlik, M. Toussaint, and S. Vijayakumar, "On stochastic optimal control and reinforcement learning by approximate inference," in *IJCAI*, 2013.

[8] M. Toussaint, "Robot trajectory optimization using approximate inference," in *ICML*, 2009.

[9] D. M. Blei, A. Kucukelbir, and J. D. McAuliffe, "Variational inference: A review for statisticians," *Journal of the American Statistical Association*, 2017.

[10] C. Winkler, D. Worrall, E. Hoogeboom, and M. Welling, "Learning likelihoods with conditional normalizing flows," 2019.