Sampling Constrained Trajectories Using Composable Diffusion Models

Thomas Power[1], Rana Soltani-Zarrin[2], Soshi Iba[2] and Dmitry Berenson[1]

***Abstract*— Trajectory optimization and optimal control are powerful tools for synthesizing complex robot behavior using appropriate cost functions and constraints. However, methods for solving the optimization problem are often prone to local minima and sensitive to initialization. Casting trajectory optimization as an inference problem can alleviate some of these issues by generating distributions over solutions. However, the resulting inference problem can be costly. In this work, we present an approach for using diffusion models to learn a distribution over constraint-satisfying low-cost trajectories. This learned distribution is then used as the initialization for an inference-based trajectory optimization algorithm. We exploit the composability of diffusion models to generalize the learned generative model to out-of-distribution constraints which consist of the composition of multiple in-distribution constraints. We demonstrate the benefit of our approach by showing improvement over baselines on a constrained 12DoF Quadrotor task and a 7DoF robot manipulator task.**

I. Introduction

Trajectory optimization and optimal control are important tools for generating complex robot behavior [1]–[5]. When performing trajectory optimization, ensuring constraint satisfaction is crucial to ensure trajectories are safe. Satisfying these constraints can be very difficult as constraint-satisfying trajectories may lie on lower-dimensional manifolds that have zero measure, presenting difficulties for sample-based methods. In addition, many useful tasks entail constrained optimization problems that are non-convex and exhibit multiple local minima. This makes trajectory optimization difficult for gradient-based methods, as poor initialization may lead to poor local minima or infeasible solutions.

In this paper, we formulate the constrained trajectory optimization problem as a Bayesian inference problem. This view has advantages as it aims to find a distribution over trajectories rather than a single trajectory alone, which can improve exploration of the search space and give greater robustness to initialization. Previous methods taking the inference view of trajectory optimization have only been able to incorporate constraints via penalties in the cost [6]–[9]. Tuning the weights of the penalties is challenging due to possible conflicts with the objective function.

Recently, Power and Berenson [10] proposed Constrained Stein Variational Trajectory Optimization (CSVTO), an algorithm that uses a non-parametric approximation of the posterior over low-cost constraint-satisfying trajectories. By generating diverse sets of trajectories, this method is more robust to initialization than baselines that rely on a single trajectory. However, the method can still fail if all of the trajectory initializations are poor. In addition, the computational time increases as the number of constraints and time horizon increases. When running with a limited computational budget we generally do not have time to run until convergence. In this paper, we introduce a method for learning a generative model of trajectories which is used as an initialization for CSVTO. Crucially, we propose using *composable* diffusion models to generalize the learned generative model to out-of-distribution constraints which consist of the compositions of constraints seen in training. This composition ability is important for tasks where the task-specific constraints are not known at training-time and training on all possible constraints the robot might encounter is intractable. Our results show improved performance with a finite computational budget for two experiments; a 12DoF quadrotor and a 7DoF manipulator.

This work was supported by Honda Research Institute. [1]Robotics Department, University of Michigan, Ann Arbor, MI, USA. {tpower, dmitryb}@umich.edu [2]Honda Research Institute, San Jose, CA, USA {rana_soltanizarrin, siba}@honda-ri.com

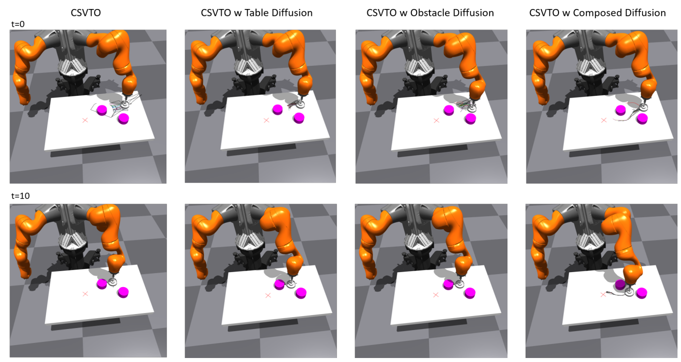

Fig. 1. Example trajectories for the 7DoF manipulator on a table experiment. At the first timestep, the initial trajectories from CSVTO are quite poor, and CSVTO is unable to pass through the narrow passage. CSVTO with single-constraint diffusion models generates initial trajectories towards the goal but fails to make progress past the initial passage. CSVTO with the composed diffusion generates trajectories that immediately pass through the narrow passage and satisfy the table constraint.

II. Related Work

a) Learning-based Constrained Planning: Learning-based approaches have previously been used to improve planning in constrained domains. Qureshi et al. proposed Constrained Motion Planning Networks (CoMPNetX) [11], a learning-based method for constrained sample-based motion planning. Generative Adversarial Networks (GANs) have been used to learn distributions of configurations satisfying constraints, by Lembono et al [12] for use with constrained sample-based motion planning and by [13] for generating initializations for a trajectory optimization problem. Kicki et al. proposed an approach for generating constraint-satisfying trajectories with a neural network that outputs a B-spline parameterization of the trajectories [14].

b) Diffusion Models in Robotics: Diffusion probabilistic models [15]–[17] are a class of generative models that have been recently applied to generating high-quality images [15], trajectories [18], [19] learning multi-modal policies [20] and learning costs for grasp optimization [21]. Two mechanisms have been proposed for incorporating conditioning information, classifier-guidance [22], which uses the gradients of an additionally learned classifier with the unconditional diffusion model, and classifier-free guidance [23] which instead learns a conditional diffusion model which takes the context information as input to the diffusion model. One interesting feature of diffusion models that has been recently explored is the composition of context information at test time, generalizing to novel combinations of context [19], [24], [25]. Recent work by Carvalho et al. [26] has applied diffusion models to motion planning. Their approach is similar to ours but does not consider problems with constraints other than obstacle avoidance.

III. Preliminaries

A. Trajectory Optimization

Trajectory optimization is commonly modeled as an Optimal Control Problem (OCP). We consider a discrete-time system with state $\mathbf{x} \in \mathbb{R}^{d_x}$ and control $\mathbf{u} \in \mathbb{R}^{d_u}$ and dynamics $\mathbf{x}_t = f(\mathbf{x}_{t-1}, \mathbf{u}_{t-1})$. We define finite horizon trajectories with horizon T as $\tau = (\mathbf{X}, \mathbf{U})$, where $\mathbf{X} = \{\mathbf{x}_1, ...\mathbf{x}_T\}$ and $\mathbf{U} = \{\mathbf{u}_0, ...\mathbf{u}_{T-1}\}$. Given an initial state $\mathbf{x}_0$, the aim when solving an OCP is to find a trajectory τ that minimizes a given cost function C subject to equality and inequality constraints:

$$\begin{aligned}
\min_{\tau} \quad & C(\tau) \\
\text{s.t.} \quad & \\
& h(\tau) = 0 \\
& g(\tau) \leq 0 \\
& \forall t \in \{1, \ldots, T\} \\
& f(\mathbf{x}_{t-1}, \mathbf{u}_{t-1}) = \mathbf{x}_t \\
& \mathbf{u}_{\text{min}} \leq \mathbf{u}_{t-1} \leq \mathbf{u}_{\text{max}} \\
& \mathbf{x}_{\text{min}} \leq \mathbf{x}_{t-1} \leq \mathbf{x}_{\text{max}}.
\end{aligned} \tag{1}$$

B. Variational Inference for Trajectory Optimization

In this section, we will demonstrate how unconstrained trajectory optimization can be framed as an inference problem, as in [9], [27]–[29]. By using this framing we leverage approximate inference tools, in particular, Variational Inference [30]. In this section, we will show how this framing leads to an entropy-regularized objective [8] which aims to find a high-entropy distribution over low-cost trajectories. This results in improved exploration of the search space and greater robustness to initialization.

To reframe trajectory optimization as probabilistic inference, we first introduce an auxiliary binary random variable o for a trajectory such that

$$p(o = 1|\tau) = \exp\left(-C(\tau)\right). \tag{2}$$

We can see that maximizing the log-likelihood of $p(o = 1|\tau)$ is equivalent to minimizing the cost. We aim to find the posterior distribution $p(\tau|o = 1) \propto p(o = 1|\tau)p(\tau)$, where τ, $p(\tau) = p(\mathbf{X}, \mathbf{U})$ is a prior on trajectories. For deterministic dynamics, this prior is determined by placing a prior on controls $\mathbf{U}$. Choosing a Gaussian prior results in a squared control cost. Alternatively the prior could be learned from data [31]. The trajectory prior is

$$p(\tau) = p(\mathbf{U}) \prod_{t=1}^{T} \delta(\mathbf{x}_t - \hat{\mathbf{x}}_t) \tag{3}$$

where $\hat{\mathbf{x}}_t = f(\mathbf{x}_{t-1}, \mathbf{u}_{t-1})$, and δ is the Dirac delta function. This inference problem can be performed exactly for the case of linear dynamics and quadratic costs [32], [33]. However, in general, this problem is intractable and approximate inference techniques must be used. We use variational inference to approximate $p(\tau|o = 1)$ with distribution $q(\tau)$ which minimizes the divergence $\mathcal{KL}(q(\tau)||p(\tau|o = 1))$ [30]. This results in minimizing the *variational free energy* $\mathcal{F}$ (see [10] for a detailed derivation)

$$\mathcal{F}(q) = \mathbb{E}_{q(\tau)}[C(\tau)] - \mathbb{E}_{q(\tau)}[\log p(\tau)] - \mathcal{H}(q(\tau)), \tag{4}$$

where $\mathcal{H}(q(\tau))$ is the entropy of $q(\tau)$. Intuitively, we can understand that the first term promotes low-cost trajectories, the second is a regularization on the trajectory, and the entropy term prevents the variational posterior from collapsing to a *maximum a posteriori* (MAP) solution.

IV. Problem Statement

We frame the constrained optimal control problem introduced in Section III-A as a probabilistic inference problem, using ideas developed in Section III-B. We consider the constrained optimization problem as an unconstrained optimization problem with infinite cost assigned to constraint violations. This results in $p(o = 1|\tau) = 0 \implies p(\tau|o = 1) = 0$, hence constraint violating trajectories are zero probability. We can convert the unconstrained optimization problem to to the following constrained optimization problem on the space of probability distributions:

$$\begin{aligned}
q^* = \min_{q} \quad & \tilde{\mathcal{F}}(q) \\
\text{s.t.} \quad & \\
& P_q(h(\tau) = 0) = 1 \\
& P_q(g(\tau) \leq 0) = 1 \\
& \forall t \in \{1, \ldots, T\} \\
& P_q(f(\mathbf{x}_{t-1}, \mathbf{u}_{t-1}) = \mathbf{x}_t) = 1 \\
& P_q(\mathbf{u}_{\text{min}} \leq \mathbf{u}_{t-1} \leq \mathbf{u}_{\text{max}}) = 1 \\
& P_q(\mathbf{x}_{\text{min}} \leq \mathbf{x}_t \leq \mathbf{x}_{\text{max}}) = 1.
\end{aligned} \tag{5}$$

CSVTO solves this functional optimization problem with a non-parametric approximation of q^*, and solves a single planning query for a specified cost C, constraints g, h, and dynamics f. We can then say that $q^* = q^*(C, g, h, f)$. We will assume f is fixed, and that the cost C is parameterized by a start x_0 and a goal x_g. We assume that the constraints g, h are parameterized by $\{\theta, y\}$, where $\theta \in \mathbb{R}^n$ are continuous parameterizations of the constraint and $y \in [1, ..., M]$ is an indicator variable for the constraint type. For instance, different types of constraints could be obstacle avoidance constraints vs. end-effector pose constraints. Thus $q^* = q^*(x_0, x_g, \theta)$. We aim to learn a generative model which approximates this q^*. For a given x_0, x_g, θ we use CSVTO to generate sampled trajectories $(\mathbf{X}, \mathbf{U})$ from q^*. The data from which we will learn our generative model is $\{\{\mathbf{X}_i, \mathbf{U}_i\}_{i=1}^K, x_0, x_g, \theta, y\}^N$. By using this generative model as an initialization, our goal is to achieve better performance and lower constraint violation within a limited computational budget. In addition, we seek to generalize to unseen combinations of constraints, i.e. for constraints h_i, h_j seen individually during training we aim to generalize to the case where it is necessary to satisfy both h_i and h_j.

V. Method

Given trajectory samples $(\mathbf{X}, \mathbf{U}) \sim q^*(x_0, x_g, \theta)$, we use a conditional diffusion model $p_\psi(X, U | x_0, x_g, \theta)$ to learn a generative model of the data. We will first give an overview of diffusion models.

A. Diffusion Models

Diffusion probabilistic models [15]–[17] are a class of generative models that have been shown to be highly effective for learning distributions of trajectories [18], [19]. Given a dataset $\mathcal{D} = \{\tau\}^N$, the data samples are τ_0 and a predefined forward noising process $q(\tau_{k+1}|\tau_k) = \mathcal{N}(\sqrt{\alpha_k}\tau_k, (1-\alpha_k)\mathbf{I})$ is used to progressively add noise to the data for K steps, resulting in $\tau_1, ..., \tau_K$ increasingly noisy latent vectors. K and α_k are chosen such that $\tau_K \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. To sample from the model, we use a trainable reverse process $p_\psi(\tau_{k-1}|\tau_k) = \mathcal{N}(\mu(\tau_k, k), \Sigma_k)$, where μ is parameterized by a neural network, and Σ_k is typically fixed, but can in principle be learned. Diffusion models are learned with the loss

$$\mathcal{L}(\psi) = \mathbb{E}_{k\sim[1,K], \tau_0\sim\mathcal{D}, \tau_k\sim q(\tau_k|\tau_0), \epsilon\sim\mathcal{N}(\mathbf{0},\mathbf{I})}[||\epsilon - \epsilon_\psi(\tau_k, k)||^2] \tag{6}$$

where ϵ_ψ is a neural network. The mean of the reverse process μ is then calculated from this ϵ_ψ.

a) Classifier-free Guidance for Conditional Diffusion Models: In the previous section, we described an unconditional diffusion model. However, we would like to generate trajectories conditioned on the start, goal, and constraints. For convenience, we label all contextual information as c, the dataset is then $\mathcal{D} = \{\tau, c\}^N$. We use a technique known as classifier-free guidance [23]. The diffusion model is modified to also take the context as an input $\epsilon(\tau_k, c, k)$. The loss then becomes

$$\mathcal{L}(\psi) = \mathbb{E}_{k\sim[1,K], \tau_0, c\sim\mathcal{D}, \tau_k\sim q(\tau_k|\tau_0), \epsilon\sim\mathcal{N}(\mathbf{0},\mathbf{I})}[||\epsilon - \epsilon_\psi(\tau_k, c, k)||^2]. \tag{7}$$

During training, with some dropout probability $p_{unconditional}$ we replace c with $\emptyset$, this effectively trains a conditional generative model and an unconditional generative model together. When sampling, the conditional and unconditional models are combined via

$$\hat{\epsilon}(\tau_k, c, k) = \epsilon_\psi(\tau_k, \emptyset, k) + \omega(\epsilon_\psi(\tau_k, c, k) - \epsilon_\psi(\tau_k, \emptyset, k)), \tag{8}$$

where ω controls the influence of the conditioning information. We can see that $\omega = 1$ corresponds to simply using $\hat{\epsilon} = \epsilon_\psi(\tau_k, c, k)$, ω is typically chosen to be larger than 1 to more strongly incorporate the conditioning information.

B. Composing Constraints

As introduced in the previous section, the conditional diffusion model is $\epsilon_\psi(\tau_k, c, k)$, where c is the contextual information. The contextual information is $\{x_0, x_g, \theta, \hat{y}\}$, where $\hat{y}$ is a one-hot encoding of y. We are interested in composing constraints, such that we can generalize to novel combinations of constraints that have not been seen together during training. Suppose we have L constraints, then the contextual information is $c = \{x_0, x_g, \theta_1, \hat{y}_1, ..., \theta_L, \hat{y}_L\}$. These are composed at test time via

$$\hat{\epsilon}(\tau_k, c, k) = \epsilon_\psi(\tau_k, \emptyset, k) + \sum_{i=1}^{L} \omega_i(\epsilon_\psi(\tau_k, \{x_0, x_g, \theta_i, \hat{y}_i\}, k) - \epsilon_\psi(\tau_k, \emptyset, k)), \tag{9}$$

where ω_i is a hyperparameter that controls the relative influence of each constraint.

C. Architecture

For the neural network architecture we use the 1-D convolutional U-Net described in [18]. We encode the start, goal, and constraint information with a multi-layer perceptron (MLP) to a $\mathbb{R}^{256}$ vector which is used to condition the network via Feature-wise Linear Modulation (FiLM) [34]. To query and train the unconditional diffusion model, we replace this vector with the zero vector. We train with Adam and a learning rate of 1×10^{-4}.

D. Using the learned model for planning with CSVTO

In principle, a perfect generative model planning would simply consist of sampling from the diffusion model, as in [18], [19]. However, to ensure that trajectories satisfy the constraints we use the samples as the initialization for CSVTO. CSVTO starts from an initial set of particles and drives the particles towards constraint satisfaction and low cost while also promoting diversity. Suppose we are running CSVTO with N particles. In the first time-step, we sample N trajectories from the generative model to get a set of initial trajectory samples. At subsequent timesteps, CSVTO gives N initial trajectories which are the shifted result of the previous time-steps optimization. We sample an additional N trajectories from the generative model and choose the best N of the $2N$ trajectories to serve as the initialization for the optimization.

VI. Evaluation

We evaluate our approach in three experiments. The first is a constrained 12DoF quadrotor task which has nonlinear underactuated dynamics. The second experiment is a 7DoF robot manipulator task where the aim is to move the robot end-effector to a goal location while being constrained to move along the surface of a table.

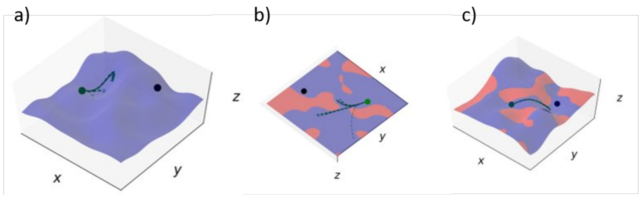

Fig. 2. Experimental setup for the training and evaluation of the quadrotor tasks. The quadrotor must travel to the goal location. a) The quadrotor is constrained to travel along a non-linear surface shown in purple. b) The quadrotor must avoid the infeasible regions in the x-y plane shown in red. c) The quadrotor must satisfy both the previous constraints. The combination of these two constraints is not seen during training

A. Ablations

We compare using our proposed composable learned generative model for trajectory optimization with ablations. The first is CSVTO without a learned generative model. For the second ablation we compare against using the learned diffusion model without the composability, e.g. for a task that consists of satisfying two constraints, we use a diffusion model which only takes into account one constraint.

B. 12DoF Quadrotor

For this task, there are two types of constraints; a constraint that the quadrotor must travel along a nonlinear surface $z = f_{surf}(x, y)$, and that it must avoid obstacles in the x-y plane, with in-collision configurations described by $f_{obs}(x, y) < 0$. To generate different versions of each of these two types of constraints, we sample f_{obs} and f_{surf} from a Gaussian Process prior with an RBF kernel and zero mean function. To do this, we sample 10×10 function evaluations on an x-y grid. We then fit a GP to these and use the posterior mean as the constraint function. The constraint is parameterized by the 10×10 function values. We collect a dataset that consists of trajectories that satisfy either the surface constraint or the obstacle avoidance constraint. To collect the dataset, we generate 10000 parameterizations for each constraint type. For each constraint we run CSVTO with 16 particles and generate trajectories for 10 different starts and goals. We evaluate with an unseen setup in which the quadrotor must satisfy both the obstacle constraint and the surface constraint at the same time. Examples are shown in Figure 2.

We run this experiment for 20 trials with randomly sampled starts, goals, surface, and obstacle constraints. The results are shown in Figure 3. CSVTO with the composed diffusion outperforms the ablations, achieving succeeding 13/20 times at a goal threshold of 0.6m, compared with 9/20 for CSVTO with no diffusion, the next best baseline. We see that CSVTO with a diffusion model that only takes into account the surface constraint performs similarly to CSVTO with no diffusion, whereas CSVTO which only takes into account the obstacle constraint performs significantly worse.

C. Robot Manipulator on Surface

For this experiment, we use the same experimental setup as that described in VI-C. In this task, there are again two types of constraints, a surface constraint that the end-effector must be constrained to along the surface of the table,

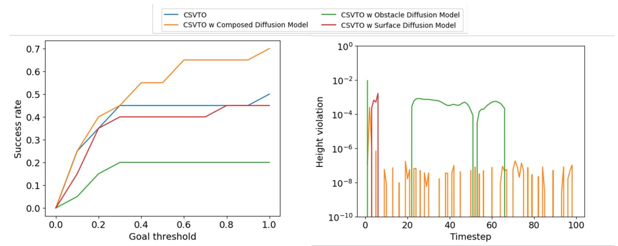

Fig. 3. Results for quadrotor experiments. The left row shows the success rate vs. goal region size. The right shows the average constraint violation as a function of time

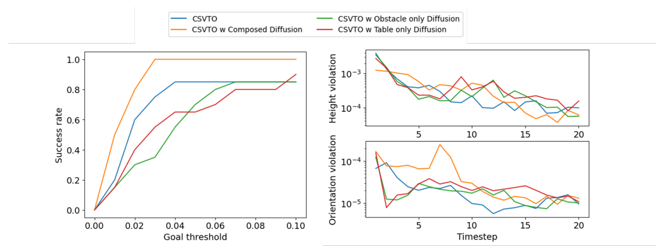

Fig. 4. Results for manipulator table experiments. The left row shows the success rate as we increase the size of the goal region. The right shows the average constraint violation as a function of time

and an obstacle constraint that the end-effector must avoid two cylindrical obstacles in the x-y plane. For the obstacle constraint, $\theta \in \mathbb{R}^4$ is the x-y positions of the center of both obstacles, while for the surface constraint, $\theta = [h, 0, 0, 0]$, where h is the table height. We collect a dataset by generating 10000 surface constraints and 10000 obstacle constraints. For each constraint, we run CSVTO with 16 particles and generate trajectories for 10 different starts and goals. We evaluate our approach in a scenario in which the robot must satisfy both constraints at the same time.

We run this experiment for 20 trials with randomly sampled starts, goals, surface, and obstacle constraints. The results are shown in Figure 4. CSVTO with the composed diffusion outperforms the ablations, achieving succeeding 20/20 times at a goal threshold of 0.04m, compared with 17/20 for CSVTO with no diffusion, the next best baseline. Examples of all methods are shown in Figure 1.

VII. Conclusion

In this paper, we presented a method for learning a generative model to initialize CSVTO using composable diffusion models. We demonstrate that by incorporating the learned generative model we can outperform CSVTO, with a success rate of 13/20 vs 9/20 for a 12DoF quadrotor task and 20/20 vs 17/20 on a 7DoF manipulator. In both of these experiments, we apply the generative model to novel combinations of constraints. In future work, we would like to extend this approach to more manipulation tasks with more diverse combinations of constraints.

References

[1] R. Bonalli, A. Cauligi, A. Bylard, and M. Pavone, "Gusto: Guaranteed sequential trajectory optimization via sequential convex programming," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2019, pp. 6741–6747.

[2] J. Schulman, Y. Duan, J. Ho, A. Lee, I. Awwal, H. Bradlow, J. Pan, S. Patil, K. Goldberg, and P. Abbeel, "Motion planning with sequential convex optimization and convex collision checking," *Int. J. Rob. Res.*, vol. 33, no. 9, pp. 1251–1270, 2014.

[3] T. A. Howell, B. E. Jackson, and Z. Manchester, "Altro: A fast solver for constrained trajectory optimization," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2019, pp. 7674–7679.

[4] M. S. Phoon, P. S. Schmitt, and G. V. Wichert, "Constraint-based task specification and trajectory optimization for sequential manipulation," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2022, pp. 197–202.

[5] C. Brasseur, A. Sherikov, C. Collette, D. Dimitrov, and P.-B. Wieber, "A robust linear mpc approach to online generation of 3d biped walking motion," in *Proc. 15th IEEE-RAS Int. Conf. Humanoid Robots*, 2015, p. 595–601.

[6] H. Yu and Y. Chen, "A gaussian variational inference approach to motion planning," *IEEE Robot. Autom. Lett.*, vol. 8, no. 5, pp. 2518–2525, 2023.

[7] M. Mukadam, X. Yan, and B. Boots, "Gaussian process motion planning," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2016, pp. 9–15.

[8] A. Lambert and B. Boots, "Entropy regularized motion planning via stein variational inference," 2021, arxiv.2107.05146.

[9] A. Lambert, A. Fishman, D. Fox, B. Boots, and F. Ramos, "Stein variational model predictive control," in *Proc. Conf. Robot Learn.*, 2020.

[10] T. Power and D. Berenson, "Constrained stein variational trajectory optimization," 2023, arxiv.2308.12110.

[11] A. H. Qureshi, J. Dong, A. Baig, and M. C. Yip, "Constrained motion planning networks x," *IEEE Transactions on Robotics*, vol. 38, no. 2, pp. 868–886, 2022.

[12] T. S. Lembono, E. Pignat, J. Jankowski, and S. Calinon, "Learning constrained distributions of robot configurations with generative adversarial network," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 4233–4240, 2021.

[13] J. Ortiz-Haro, J.-S. Ha, D. Driess, and M. Toussaint, "Structured deep generative models for sampling on constraint manifolds in sequential manipulation," in *Conference on Robot Learning*, 2021.

[14] P. Kicki, P. Liu, D. Tateo, H. Bou-Ammar, K. Walas, P. Skrzypczyński, and J. Peters, "Fast kinodynamic planning on the constraint manifold with deep neural networks," 2023, arxiv.2301.04330.

[15] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," in *Proceedings of the 34th International Conference on Neural Information Processing Systems*, 2020.

[16] J. Sohl-Dickstein, E. Weiss, N. Maheswaranathan, and S. Ganguli, "Deep unsupervised learning using nonequilibrium thermodynamics," in *Proceedings of the 32nd International Conference on Machine Learning*, 2015, pp. 2256–2265.

[17] Y. Song and S. Ermon, "Generative modeling by estimating gradients of the data distribution," in *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, 2019.

[18] M. Janner, Y. Du, J. Tenenbaum, and S. Levine, "Planning with diffusion for flexible behavior synthesis," in *International Conference on Machine Learning*, 2022.

[19] A. Ajay, Y. Du, A. Gupta, J. B. Tenenbaum, T. S. Jaakkola, and P. Agrawal, "Is conditional generative modeling all you need for decision making?" in *The Eleventh International Conference on Learning Representations*, 2023.

[20] C. Chi, S. Feng, Y. Du, Z. Xu, E. Cousineau, B. Burchfiel, and S. Song, "Diffusion policy: Visuomotor policy learning via action diffusion," in *Proceedings of Robotics: Science and Systems (RSS)*, 2023.

[21] J. Urain, N. Funk, J. Peters, and G. Chalvatzaki, "Se(3)-diffusionfields: Learning smooth cost functions for joint grasp and motion optimization through diffusion," *IEEE International Conference on Robotics and Automation (ICRA)*, 2023.

[22] P. Dhariwal and A. Q. Nichol, "Diffusion models beat GANs on image synthesis," in *Advances in Neural Information Processing Systems*, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, Eds., 2021. [Online]. Available: https://openreview.net/forum?id=AAWuCvzaVt

[23] J. Ho and T. Salimans, "Classifier-free diffusion guidance," 2022.

[24] N. Liu, S. Li, Y. Du, A. Torralba, and J. B. Tenenbaum, "Compositional visual generation with composable diffusion models," in *ECCV 2022*, S. Avidan, G. Brostow, M. Cissé, G. M. Farinella, and T. Hassner, Eds., 2022, pp. 423–439.

[25] Y. Du, C. Durkan, R. Strudel, J. B. Tenenbaum, S. Dieleman, R. Fergus, J. Sohl-Dickstein, A. Doucet, and W. S. Grathwohl, "Reduce, reuse, recycle: Compositional generation with energy-based diffusion models and MCMC," in *Proceedings of the 40th International Conference on Machine Learning*, 2023, pp. 8489–8510.

[26] J. Carvalho, A. T. Le, M. Baierl, D. Koert, and J. Peters, "Motion planning diffusion: Learning and planning of robot motions with diffusion models," 2023, arxiv.2308.01557.

[27] K. Rawlik, M. Toussaint, and S. Vijayakumar, "On stochastic optimal control and reinforcement learning by approximate inference," in *Robot.: Sci. Syst.*, 2013.

[28] M. Toussaint, "Robot trajectory optimization using approximate inference," in *Proc. Int. Conf. Mach. Learn.*, 2009, p. 1049–1056.

[29] M. Okada and T. Taniguchi, "Variational inference mpc for bayesian model-based reinforcement learning," in *Proc. Conf. Robot Learn.*, 2020, pp. 258–272.

[30] D. M. Blei, A. Kucukelbir, and J. D. McAuliffe, "Variational inference: A review for statisticians," *J. Am. Stat. Assoc.*, vol. 112, no. 518, pp. 859–877, 2017.

[31] J. Urain, A. T. Le, A. Lambert, G. Chalvatzaki, B. Boots, and J. Peters, "Learning implicit priors for motion optimization," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2022, pp. 7672–7679.

[32] H. Attias, "Planning by probabilistic inference," in *Proc. 9th Int. Workshop on Artificial Intelligence and Statistics*, 2003, pp. 9–16.

[33] J. Watson, H. Abdulsamad, and J. Peters, "Stochastic optimal control as approximate input inference," in *Proc. Conf. Robot Learn.*, vol. 100, 2020, pp. 697–716.

[34] E. Perez, F. Strub, H. de Vries, V. Dumoulin, and A. C. Courville, "Film: Visual reasoning with a general conditioning layer," in *AAAI*, 2018.