Learning Implicit Priors for Motion Optimization

Alexander Lambert*[1], An T. Le*[2], Julen Urain*[2], Georgia Chalvatzaki[2], Byron Boots[1] and, Jan Peters[2]

***Abstract*— Motion optimization is an effective framework for generating smooth and safe trajectories for but it suffers from local optima that hinder its applicability, especially for multi-objective tasks. In this paper, we study the integration of Energy-Based Models (EBM) as implicit priors for guiding optimizers. This work presents a set of necessary modeling and algorithmic choices to effectively learn and integrate EBMs into motion optimization. We investigate the benefit of smoothness regularization in the learning process that benefits gradient-based optimizers. Moreover, we present a set of EBM architectures for learning generalizable distributions over trajectories that are important for the subsequent deployment of EBMs. Videos and additional details are available at https://sites.google.com/view/implicit-priors.**

I. Introduction

Motion planning is a fundamental property for autonomous robots to achieve task-specific goals. In the context of autonomous robot manipulation, the trajectories, that a robot should execute, should satisfy several constraints, e.g., approaching the goal while avoiding collisions and joint limits. Naturally, a complex motion plan can be viewed as a multi-objective optimization problem along a specific time horizon. In this work, we study motion planning in light of *motion optimization* methods [1]–[4].

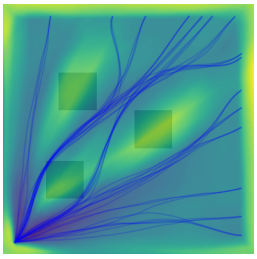
Motion optimization is an inherently local optimization method that relies on the initialization, iteratively making local updates at every optimization step. Hence, due to the possible non-convexity of the cost function, these optimization methods suffer from local minima. Additionally, if the cost function is sparse, it might be hard to get the proper information for reaching low-cost regions, and therefore, the initial proposal may barely improve. A way to avoid the local minima traps in trajectory optimization is to include a set of handcrafted priors in the trajectory optimization [1], [2]. In a different vein, to capture the inherent multimodality of multi-objective motion planning tasks, a line of work proposes to learn trajectory distributions from data to guide the optimization process away from local minima [5], [6].

In this work, we study the prior modeling using Energy Based Models (EBM) [7] as *implicit models* [8] for motion optimization. EBMs can be represented in arbitrary latent spaces (e.g. task spaces), without the requirement of representing them in the configuration space of the robot, as opposed to most explicit learning from demonstration methods [5]. Additionally, due to their exponential nature, EBMs can be easily combined, allowing the composition of multiple priors, representing sub-tasks of the manipulation task, into a single structured prior. We propose a motion optimization framework using implicit prior functions that are modular, learnable, differentiable, and composable. Using our learned EBMs as priors, we can integrate multimodal information that can bias and guide the optimization process towards finding a feasible and smooth solution in complex tasks.

∗ All authors contributed equally. Ordering is alphabetical.

[1] Paul G. Allen School of Computer Science and Engineering (CSE), University of Washington (USA)

[2] Computer Science Department, Technische Universität Darmstadt (Germany)



(a) EBM for trajectories

(b) Pouring Task

Fig. 1: Implicit trajectory distributions are learned from demonstrations using EBMs. These guide motion optimization to produce feasible trajectories for new problems. (a) An obstacle avoidance energy function, with generated optimal trajectories towards different goal locations. (b) A robot manipulator using learned EBMs to pour a cup within a cluttered scene.

II. Background

Trajectory optimization. Denoting the system state at time t to be $\boldsymbol{x}_t \in \mathbb{R}^d$, we can define a discrete-time trajectory as the sequence $\boldsymbol{\tau} \triangleq (\boldsymbol{x}_0, \boldsymbol{x}_1, ..., \boldsymbol{x}_{T-1}, \boldsymbol{x}_T)$ over a planning horizon T. For a given start-state $\boldsymbol{x}_0$, trajectory optimization aims to find the optimal trajectory $\boldsymbol{\tau}^*$ which minimizes an objective function $c(\boldsymbol{\tau}, \boldsymbol{x}_0)$:

$$\boldsymbol{\tau}^* = \arg\min_{\boldsymbol{\tau}} c(\boldsymbol{\tau}, \boldsymbol{x}_0, \boldsymbol{x}_g) \tag{1}$$

Summarizing the *context* parameters of the planning problem as $\mathcal{E} = [\boldsymbol{x}_0, \boldsymbol{x}_g, ...]^\top$, the objective function can also be written more generally to include any number of cost terms: $c(\boldsymbol{\tau}, \mathcal{E}) = \sum_i c_i(\boldsymbol{\tau}, \mathcal{E})$. Gradient-based strategies for trajectory optimization typically resort to second-order iterative methods similar to Gauss-Newton [4], [9], or use pre-conditioned gradient-descent [1] to find a locally optimal solution to the objective. On the other hand, sampling-based approaches resort to stochastics generation of candidate trajectories using a proposal distribution. These samples are then evaluated on the objective and weighted according to their relative performance [2], [10].

Planning as inference. The duality between probabilistic inference and optimization for planning and control has been widely explored [4], [11], [12]. With open-loop trajectory optimization, in particular, we view the trajectory τ as a random variable and first consider the target distribution:

$$p(\boldsymbol{\tau}; \mathcal{E}) = \frac{1}{Z} \prod_i p_i(\boldsymbol{\tau}; \mathcal{E}) \tag{2}$$

where each p_i term consists of an individual probability factor. Optimization can then be formulated as a *maximum-likelihood estimation* (MLE) problem, where we seek to find $\boldsymbol{\tau}^* = \arg\max_{\boldsymbol{\tau}} p(\boldsymbol{\tau}; \mathcal{E})$. This can be done by minimizing the negative-log of the distribution:

$$\boldsymbol{\tau}^* = \boldsymbol{\tau}_{\text{MLE}} = \arg\min_{\boldsymbol{\tau}} -\log \prod_i p_i(\boldsymbol{\tau}; \mathcal{E}) \tag{3}$$

Assuming that these probability densities belong to the exponential family, we can relate them to the previous cost terms: $p_i(\boldsymbol{\tau}; \mathcal{E}) \propto \exp(-c_i(\boldsymbol{\tau}; \mathcal{E}))$. Substituting these into Eq. (3) recovers Eq. (1).

Fig. 2: Learned EBM (blue) and its gradient (orange) in a 1D dataset. (Left) EBM trained with vanilla CD. (Right) EBM trained with CD loss + denoising regularizer (6).

III. Method

Given a new context $\mathcal{E}$, performing inference over the posterior distribution in Eq. (2) and Eq. (3) requires that we define a prior distribution of trajectories, $p(\boldsymbol{\tau};\mathcal{E})$. Given a dataset $\mathcal{D} = \{\boldsymbol{\tau}_j, \mathcal{E}_j\}_{j=1:N}$, we propose to model and *learn* such prior trajectory distributions from collected data. We define this distribution as an EBM:

$$p_{\boldsymbol{\theta}}(\boldsymbol{\tau}|\mathcal{D};\mathcal{E}) = \frac{1}{Z}\exp(-E_{\boldsymbol{\theta}}(\boldsymbol{\tau},\mathcal{E})). \tag{4}$$

with model parameters $\boldsymbol{\theta}$. In practice, $\mathcal{E}$ represents the planning contexts, e.g., goal targets, obstacles positions, trajectory phase, etc. The dataset $\mathcal{D}$ may consist of a collection of expert demonstrations on different environments, and we aim to fit a density function representing the data distribution with Contrastive Divergence (CD) objective [13]. While the learned prior distribution is based on a set of demonstrations, we desire to adapt to novel scenarios beyond the demonstrated examples. Notably, instead of learning a monolithic EBM-based prior, we can factor this prior distribution depending on various aspects of the problem:

$$p_{\boldsymbol{\theta}}(\boldsymbol{\tau}|\mathcal{D};\mathcal{E}) \propto \prod_i \exp(-E_{\boldsymbol{\theta}_i}(\boldsymbol{\tau},\mathcal{E}_i)). \tag{5}$$

Such a factored distribution allows us to leverage composability, learn modular EBM factors independently, and combine them as needed for novel scenarios and planning problems [14]. Furthermore, to properly deploy the CD loss to learn trajectory distributions for motion optimization, we need to make multiple algorithmic and modeling choices. In the following, we introduce a set of proposed choices to properly learn and represent high-dimensional, long-horizon multimodal trajectory distributions via EBMs that can be beneficial for their deployment in motion optimization.

Smoothing EBMs for gradient-based optimization. To deploy EBMs in motion optimization, we need a smooth energy landscape. However, the CD objective generates an energy landscape with multiple plateaus, with high energy values in regions where there are no data and a plateau of low-energy in the regions of the data points. While the energy landscape may capture well the distribution of the demonstrations, it might not be helpful for gradient-based sampling or optimization, with gradients close to zero in the plateau and high gradients in the cliffs (Fig. 2). We propose adding a denoising score matching loss [15], [16] as regularization to smoothen the energy landscape and improve gradient information. Denoising score matching first generates a noisy sample given a data sample $\tilde{\boldsymbol{x}} \sim p(\tilde{\boldsymbol{x}}|\boldsymbol{x}) = \mathcal{N}(\boldsymbol{x}|\boldsymbol{\sigma}^2\boldsymbol{I})$, as $\tilde{\boldsymbol{x}} = \boldsymbol{x} + \boldsymbol{\sigma}\boldsymbol{\epsilon}$, with $\boldsymbol{\epsilon} \sim \mathcal{N}(0, \boldsymbol{I})$, and, then, matches the score function, $\nabla_{\boldsymbol{x}} E_{\boldsymbol{\theta}}$ to denoise the sample $\tilde{\boldsymbol{x}}$ back to $\boldsymbol{x}$

$$\mathcal{L}_{\text{DSM}} = \mathbb{E}_{p_{\mathcal{D}}(\boldsymbol{x},\mathcal{E})}\mathbb{E}_{p(\tilde{\boldsymbol{x}}|\boldsymbol{x})}\left[||\boldsymbol{\epsilon} - \nabla_{\boldsymbol{x}} E_{\boldsymbol{\theta}}(\tilde{\boldsymbol{x}},\mathcal{E})||_2^2\right]. \tag{6}$$

The loss (6) encourages the gradient of the EBM to point towards the data distribution contrarily to the CD loss.

Task-specific EBMs for Motion Optimization Here, we introduce a set of model choices to represent EBMs for motion optimization, as making proper choices on the EBM architecture improves the data representation capacity and the generalization of the learned models.

Object-centric EBMs. Learning task-conditioned motion models is a vital tool for representing task-adaptive motion behaviors. In our work, we propose to learn object-centric EBM that are useful for representing desired movements in manipulation tasks that involve objects, conditioning the learned EBM on the objects' poses.

Phase-conditioned EBM. The usability of phase-conditioned priors for motion optimization is necessary for long-horizon tasks, as with long-horizon trajectories, the dimension of the input space increases, and the learning of an EBM in that space might be challenging. Therefore, we propose the phase-conditioned EBMs:

$$p(\boldsymbol{x}|\alpha) \propto \exp(-E_{\boldsymbol{\theta}}(\boldsymbol{x},\alpha)), \tag{7}$$

with $\boldsymbol{x}$ the state and α being the phase. The phase represents a continuous variable moving from 0 to 1, encoding the temporal evolution of the manipulation task. The phase-conditioned EBM represents the state-occupancy distribution for different instances of the manipulation task. Nevertheless, the phase-conditioned EBM lacks any temporal relation between temporally adjacent points, generating non-smooth trajectories. To confront this effect, we propose combining phase-conditioned EBMs with trajectory smoothing costs to represent smooth trajectory distributions:

$$p(\boldsymbol{\tau}) \propto \exp\left(-\sum_k E_{\boldsymbol{\theta}}(\boldsymbol{x}_k,\alpha_k) + (\boldsymbol{x}_k - \boldsymbol{x}_{k+1})^2\right). \tag{8}$$

Although we may be able to learn data-driven prior distributions, as described in the previous section, we still require reliable inference and optimization methods to derive optimal trajectories $\boldsymbol{\tau}^*$ given a new planning problem expressed by $\mathcal{E}$. In the following, we present the methods for evaluation and inference on learned EBMs for trajectory distributions. This includes techniques for sampling and optimization which account for the composability of our EBM functions, as well as a stochastic trajectory optimization method suited for the planning tasks considered here.

Structured Planning Priors. Since we cannot sample from EBMs directly, we need to initialize samples by first drawing from an initial distribution, and then perform sequential updates to approximate the relevant modes. Further, learning task-specific EBM model-components is useful for portions of the objective function which are hard to define. However, we may insist on biasing our sampling given the structure of the planning problem, and incorporate known, well-defined requirements such as goal-seeking behavior and smoothness. This can be addressed by incorporating relevant distributions which are known *a-priori*, to the contextual-prior in Eq. (4)

$$p_{\boldsymbol{\theta}}(\boldsymbol{\tau}|\mathcal{D};\mathcal{E}) \propto p_0(\boldsymbol{\tau};\boldsymbol{\mu}_s,\boldsymbol{\mu}_g)\prod_i \exp(-E_{\boldsymbol{\theta}_i}(\boldsymbol{\tau},\mathcal{E}_i)), \tag{9}$$

where $p_0(\boldsymbol{\tau};\boldsymbol{\mu}_s,\boldsymbol{\mu}_g)$ is a general trajectory-based prior, which is typically a Gaussian Process (GP) prior representing the space of smooth and continuous-time trajectories [4], [17]. Similarly to [4], [18], we can directly integrate goal-reaching and smoothness into this distribution, which can then be directly sampled to initialize optimization. In practice, we can efficiently generate large quantities of these time-correlated trajectories due to our parallelized GPU implementation. Note, however, it is intractable to incorporate explicit priors on behaviors such as obstacle avoidance, for example. Hence, in practice we must resort to a *combination* of implicit and explicit priors to generate feasible trajectories from Eq. (9).

Stochastic Trajectory Optimization with GP-Priors. We can iteratively update the time-correlated sampling prior, described in the previous section, to fit the modes of a learned EBM and

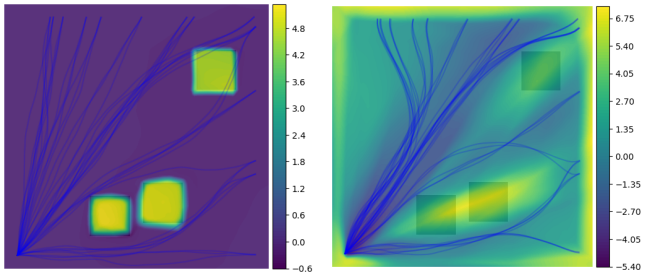


Fig. 3: Learned obstacle-EBMs conditioned on novel obstacle locations. (Left) free-space point sampling and (right) expert trajectory distributions, with multi-goal planning solutions depicted by blue trajectories. Discontinuities and implicit obstacle surfaces are well captured using sparse free-space point-samples during training, whereas distributions of trajectory-based demonstrations can be captured neatly by the EBMs. The latter provides a convenient "guiding" energy function for a new context, improving samples derived from multi-modal stochastic trajectory optimization.

allow us to sample optimal trajectories in a new planning context. The optimizer, which we call Stochastic Gaussian Process Motion Planning (StochGPMP), is closely related to the importance sampling scheme used by CEM and MPPI but uses goal-directed GP distributions. We refer the reader to link stated in the abstract for details on the derivation and specific update procedure. When used in our experiments, we select to update the mean of each component GP in the distribution. This can be easily performed in configuration space, although it requires an initial goal configuration to be approximated using Inverse-Kinematics.

IV. Experimental Evaluation

Experiment I: Simulated Planar Navigation. In this setting, a holonomic robot must reach a goal location while avoiding obstacles in a planar environment. We assume that the start, goal, and obstacle locations are known for a given planning problem, but the obstacle *geometries* (ex. size, shape) are unknown. We want to learn an implicit distribution that captures the collision-free trajectories which lead to a particular goal. Here, we investigate two possible sources of empirical data: (1) sparsely populated point-distributions in free-space and (2) a set of expert trajectory distributions. Here, the learned EBM can be expressed as $E_{\boldsymbol{\theta}}(\boldsymbol{x}, \{\boldsymbol{x}^i_{obs}\}^N_{i=1})$, where $\boldsymbol{x}$ is a particular 2D-state, and $\boldsymbol{x}_{obs}$ the position of an obstacle (here, $N = 3$). The model is a simple 2-hidden layer MLP (width=512), with concatenated inputs.

Examples of the learned EBM for both cases are shown in Fig. 3. The resulting energy functions, in either case, manage to effectively capture the demonstration distributions, conditioned on new obstacle locations. We compare this method of implicit trajectory generation to a standard Behavioral Cloning (BC) baseline, where the learned policy outputs the current velocity, $\dot{\boldsymbol{q}} = \boldsymbol{f}(\boldsymbol{q}; \mathcal{X}_o, \boldsymbol{x}_g)$ which is conditioned on the set of obstacle poses $\mathcal{X}_o = \{\boldsymbol{x}_o\}$ and the goal location $\boldsymbol{x}_g$. We perform a quantitative analysis on the EBM methods by measuring the success rate on the validation set as a function of optimization iterations needed by the planner (Table I).

Opt. iters.	0	5	10	25	50
EBM-Free-space	0.556	0.470	0.643	0.747	0.852
EBM-Expert Traj.	0.556	0.690	0.791	0.847	0.877
Behavioral cloning	0.04	–	–	–	–

TABLE I: Planar navigation: Average success rate per environment, as a function of optimization iterations. A planning trajectory is deemed successful if it ends within a radius of 1.5 from the goal, without hitting the underlying obstacles.

Experiment II: Robot pouring amid obstacles. We investigate the integration of EBMs in trajectory optimization for a pouring task in the presence of obstacles with a 7dof LWR-Kuka robot arm, showing its effectiveness in complex manipulation tasks. This experiment investigates (i) the benefit of including smoothness regularization in the EBM training, (ii) the advantages of phase-conditioned EBM w.r.t. learning the EBM in trajectory space, and (iii) the generalization of our EBMs in the context of the pouring task regarding arbitrary pouring places, and in the presence of obstacles. Instances of our method's performance are available in Fig. 4 for the simulated task, and Fig. 1 for our zero-transfer to the actual robotic setup. To learn the pouring EBM, we recorded 500 trajectory demonstrations of the pouring task. The demonstrations were generated using a set of handcrafted policies and were initialized in arbitrary initial configurations. To properly encode the temporal information in the data, we learn a time conditioned EBM, $E_{\boldsymbol{\theta}}(\boldsymbol{x}|\alpha)$. In our problem, $\boldsymbol{x}$ is a 6-dimensional state, representing the 3D position of the bottom and tip of the glass w.r.t. the pouring pot frame. Centering the EBM to the pouring pot's frame allows us to generalize the EBM to arbitrary pot poses. Additionally, we include the denoising regularization and compare its performance and compare to a baseline without the proposed regularization. We compare against three baselines. First, a solver without any prior, to appropriately evaluate the benefits of adding guiding priors. Second, a phase-conditioned EBM without smoothness regularization combined with the optimizer. Third, an EBM that is directly learned in the trajectory space to investigate the benefit of phase-conditioned EBMs in trajectory optimization. The objective function is defined by the composition of a set of cost functions–fixed initial configuration, fixed target configuration, trajectory smoothness, obstacle avoidance, keep the glass pointing up to avoid spilling and pour inside the pot. The learned EBM is added as an additional factor in the optimization problem. We optimize using a tempering scheme, giving more importance to the prior at the beginning and reducing its influence at the end of the optimization process. We report performance both in obstructed and obstacle-free environments. We run 50 episodes for each case. we randomize the position of the pouring pot and the obstacles on each episode. The obtained results are presented in Fig. 5.

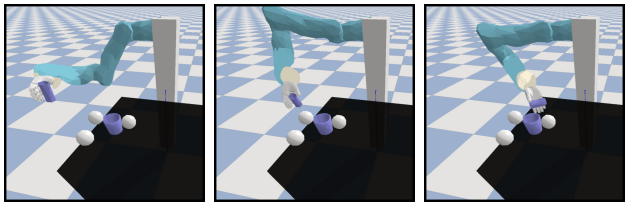


Fig. 4: A visual representation of the pouring in cluttered task.

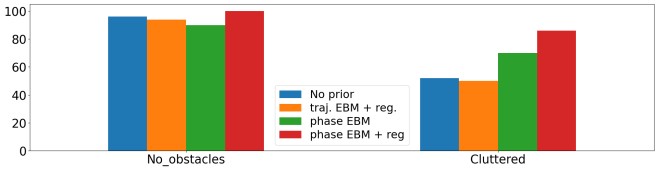


Fig. 5: Success rate (%) for the pouring task. (Left): Experiment without collision obstacles. We observe that the phase EBM with regularization can improve the performance slightly. (Right): Experiment with obstacles in the environment. We observe a clear benefit of using phase-based EBM in contrast with trajectory-based EBM. Training the EBM in high dimensional space requires too many samples, and it is difficult to get smooth EBMs representing the demonstrations. The EBM trained with regularization improves the obtained results with respect to non-regularized EBM due to informative gradient towards the demonstration region.

REFERENCES

[1] N. Ratliff, M. Zucker, J. A. Bagnell, and S. Srinivasa, "Chomp: Gradient optimization techniques for efficient motion planning," in *IEEE ICRA*, 2009.

[2] M. Kalakrishnan, S. Chitta, E. Theodorou, P. Pastor, and S. Schaal, "Stomp: Stochastic trajectory optimization for motion planning," in *IEEE ICRA*, 2011.

[3] J. Schulman *et al.*, "Motion planning with sequential convex optimization and convex collision checking," *IJRR*, 2014.

[4] M. Mukadam, J. Dong, X. Yan, F. Dellaert, and B. Boots, "Continuous-time gaussian process motion planning via probabilistic inference," *IJRR*, 2018.

[5] D. Koert, G. Maeda, R. Lioutikov, G. Neumann, and J. Peters, "Demonstration based trajectory optimization for generalizable robot motions," in *IEEE-RAS 16th Humanoids*, 2016.

[6] M. A. Rana *et al.*, "Towards robust skill generalization: Unifying learning from demonstration and motion planning," in *CoRL*, PMLR, 2017.

[7] Y. LeCun, S. Chopra, R. Hadsell, M. Ranzato, and F. Huang, "A tutorial on energy-based learning," *Predicting structured data*, 2006.

[8] Y. Du and I. Mordatch, "Implicit generation and generalization in energy-based models," *arXiv preprint arXiv:1903.08689*, 2019.

[9] M. Toussaint, "Newton methods for k-order markov constrained motion problems," *arXiv preprint arXiv:1407.0414*, 2014.

[10] Z. I. Botev *et al.*, "The cross-entropy method for optimization," in *Handbook of statistics*, Elsevier, 2013.

[11] M. Toussaint, "Robot trajectory optimization using approximate inference," in *Proceedings of the 26th annual international conference on machine learning*, pp. 1049–1056, 2009.

[12] S. Levine, "Reinforcement learning and control as probabilistic inference: Tutorial and review," *arXiv preprint arXiv:1805.00909*, 2018.

[13] G. E. Hinton, "Training products of experts by minimizing contrastive divergence," *Neural computation*, 2002.

[14] J. Urain, A. Li, P. Liu, C. D'Eramo, and J. Peters, "Composable energy policies for reactive motion generation and reinforcement learning," in *R:SS*, 2021.

[15] P. Vincent, "A connection between score matching and denoising autoencoders," *Neural computation*, 2011.

[16] Y. Song *et al.*, "Score-based generative modeling through stochastic differential equations," in *ICLR*, 2021.

[17] T. D. Barfoot, C. H. Tong, and S. Särkkä, "Batch continuous-time trajectory estimation as exactly sparse gaussian process regression.," in *Robotics: Science and Systems*, vol. 10, pp. 1–10, Citeseer, 2014.

[18] A. Lambert and B. Boots, "Entropy regularized motion planning via stein variational inference," *RSS Workshop on Integrating Planning and Learning*, 2021.