Fighting Uncertainty with Gradients: Offline Reinforcement Learning via Diffusion Score Matching

Anonymous Author(s)
Affiliation
Address
email

Abstract: Offline optimization paradigms such as offline Reinforcement Learning (RL) or Imitation Learning (IL) allow policy search algorithms to make use of offline data, but require careful incorporation of uncertainty in order to circumvent the challenges of distribution shift. Gradient-based policy search methods are a promising direction due to their effectiveness in high dimensions; however, we require a more careful consideration of how these methods interplay with uncertainty estimation. We claim that in order for an uncertainty metric to be amenable for gradient-based optimization, it must be (i) stably convergent to data when uncertainty is minimized with gradients, and (ii) not prone to underestimation of true uncertainty. We investigate smoothed distance to data as a metric, and show that it not only stably converges to data, but also allows us to analyze model bias with Lipschitz constants. Moreover, we establish an equivalence between smoothed distance to data and data likelihood, which allows us to use score-matching techniques to learn gradients of distance to data. Importantly, we show that offline model-based policy search problems that maximize data likelihood do not require values of likelihood; but rather only the gradient of the log likelihood (the score function). Using this insight, we propose Score-Guided Planning (SGP), a planning algorithm for offline RL that utilizes score-matching to enable first-order planning in high-dimensional problems, where zeroth-order methods were unable to scale, and ensembles were unable to overcome local minima.

Keywords: Diffusion, Score-Matching, Offline, Model-Based Reinforcement Learning, Imitation Learning, Planning under Uncertainty

1 Introduction

Uncertainty minimization is a central problem in offline optimization, which manifests as many different paradigms in robot learning. In offline model-based RL (MBRL [1, 2, 3, 4]), penalization of uncertainty acts as a regularizer against model bias [5, 6, 7] and prevents the optimizer from exploiting model error [8, 6, 9, 10]. In offline model-free RL, it regularizes against overestimation of the Q function [11]. In addition, many imitation learning (IL) algorithms can be viewed as minimizing distribution shift from the demonstrator distribution [12] .

Despite the importance of uncertainty, statistical uncertainty quantification remains a difficult problem [13, 14, 8, 6, 15]; Gaussian processes (GPs) [6, 16, 17] rarely scale to high dimensions, and ensembles [8, 4, 18] are prone to underestimating the true uncertainty [14]. As such, previous works have often taken the more direct approach of staying near the data by maximizing the data likelihood. These methods either minimize distribution shift between the optimized and data distribution for behavior regularization [19, 20, 21, 22], or the occupation measure of the optimized policy and the data distribution [11, 23, 24, 25, 26, 27, 28, 29]. Both of these directions would require the likelihood of the data distribution to be estimated.

A promising direction for estimating the data likelihood is to leverage techniques from likelihood-based generative modeling, such as variational autoencoders (VAE) [30, 31, 32], generative adversarial networks (GAN) [33, 34, 35, 12], and flow-based models [24, 36]. Yet, these prior works have shown

Submitted to the 7th Conference on Robot Learning (CoRL 2023). Do not distribute.

that training density models to generate accurate likelihoods can be challenging, especially for high-dimensional data. While gradient-based methods have good scalability properties that make them desirable for tackling offline optimization problems in this high-dimensional regime, the effect of incorrect likelihoods are further exacerbated in this setting as they lead gradient-based methods into spurious local minima. Thus, we ask in this work: can we design gradient-based offline optimization methods that encourage data likelihood, without explicit generative modeling of likelihoods?

Our key insight is that in order to maximize the data likelihood with gradient-based methods, we do not need access to the likelihood itself. Rather, having access to a first-order oracle (gradients), known as the *score function*, is sufficient. We claim that directly utilizing the score function has two benefits compared to likelihood-based modeling. i) First, recent breakthroughs in score-based modeling [37, 38, 39] show that the score function is considerably easier to estimate with score-matching techniques [39, 38], as it bypasses estimation of the partition function that is required for computation of exact likelihoods [39, 40]. ii) In addition, we show that score matching with annealed perturbations [39] gives gradients that stably drive decision variables to land exactly on data when uncertainty is minimized with gradient-based optimization, a property we term *data stability*. We demonstrate this by showing that the negative log likelihood of the perturbed empirical distribution, whose gradients score-matching estimates, is equivalent to a softened distance to data [41].

Furthermore, we ask: when, and why, would approaches that penalize distance to data surpass the ensemble method of statistical uncertainty quantification [4, 9]? We show that unlike empirical variance among ensembles, we can relate how much smoothed distance to data underestimates true uncertainty with the Lipschitz constant, for which we can use statistical estimation to put confidence bounds [17], or utilize structured domain knowledge [42]. Moreover, we show that ensembles do not necessarily have the data stability property due to statistical noise; therefore, optimizing for ensemble variance can easily lead to local minima away from data in gradient-based optimization.

To put our theory into a practical algorithm, we propose Score-Guided Planning (SGP), a gradient-based algorithm that estimates gradients of the log likelihood with score matching, and solves uncertainty-penalized offline optimization problems that additively combine the cumulative reward and the log likelihood of data *without* any explicit modeling of likelihood. SGP enables stable uncertainty minimization in high-dimensional problems, enabling offline MBRL to scale even to pixel action-spaces. We validate our theory on empirical examples such as the cart-pole system, the D4RL benchmark [43], a pixel-space single integrator, and a box-pushing task [44] on hardware.

2 Preliminaries

Offline Model-Based Optimization. We first introduce a setting of *offline model-based optimization* [45]. In this setting, we aim to find x that minimizes an objective function $f : \mathbb{R}^n \to \mathbb{R}$, but are not directly given access to f; instead, we have access to $x_i \sim p(x)$, and their corresponding values $f(x_i)$, such that the dataset consists of $\mathcal{D} = \{(x_i, f(x_i))\}$. Denoting $\hat{p}(x; \mathcal{D})$ as the empirical distribution corresponding to dataset $\mathcal{D}$, offline model-based optimization solves

$$\min_x f_{\theta^*}(x) \quad \text{s.t.} \quad \theta^* = \arg\min_\theta \mathbb{E}_{x \sim \hat{p}(x;\mathcal{D})} \left[\| f_\theta(x) - f(x) \|^2 \right]. \tag{1}$$

In words, we minimize a surrogate loss $f_\theta(x)$, where we choose θ as the solution to empirical risk minimization of matching f given the data. Denoting $x^* = \arg\min_x f_{\theta^*}(x)$, one measure of performance of this procedure is error at optimality, $\| f(x^*) - f_\theta(x^*) \|$.

Uncertainty Penalization. The gap $\| f(x^*) - f_\theta(x^*) \|$ potentially can be large if f_θ fails to approximate f correctly at x^*, which is likely if x^* is out-of-distribution (o.o.d.). To remedy this, previous works [8, 45, 46] have proposed adding a loss term that penalizes o.o.d. regions. We denote this penalized objective as

$$\bar{f}_\theta(x) \coloneqq f_\theta(x) + \beta \mu^2(x), \tag{2}$$

where $\beta \in \mathbb{R}_{\geq 0}$ is some weighting parameter, and $\mu(x)$ is some notion of uncertainty. Intuitively this restricts the choice of optimal x to the training distribution used to find θ^*, since these are the values

that can be trusted. If the uncertainty metric overestimates the true uncertainty, $\|f(x) - f_\theta(x)\| \leq \mu(x)$, it is possible to bound the error at optimality directly using $\mu(x^*)$.

Offline Model-Based RL. While the offline model-based optimization problem described is a one-step problem, the problem of offline model-based RL (MBRL) involves sequential decision making using an offline dataset. In offline MBRL, we are given a dataset $\mathcal{D} = \{(x_t, u_t, x_{t+1})_i\}$ where $x \in \mathbb{R}^n$ denotes the state, $u \in \mathbb{R}^m$ denotes action, t is the time index and i is the sample index. Again denoting $\hat{p}(x_t, u_t, x_{t+1}; \mathcal{D})$ as the empirical distribution corresponding to $\mathcal{D}$, and introducing $\mu(x_t, u_t)$ as a state-action uncertainty metric [8], uncertainty-penalized offline MBRL solves

$$\begin{aligned} \max_{x_{1:T}, u_{1:T}} \quad & \textstyle\sum_{t=1}^{T} r_t(x_t, u_t) - \beta\mu^2(x_t, u_t) \\ \text{s.t.} \quad & x_{t+1} = f_{\theta^*}(x_t, u_t)\ \forall t, \\ & \theta^* = \arg\min_\theta \mathbb{E}_{(x_t, u_t, x_{t+1}) \sim \hat{p}} \left[\|x_{t+1} - f_\theta(x_t, u_t)\|^2 \right]. \end{aligned} \tag{3}$$

In words, we first approximate the transition dynamics of the system from data, then solve the uncertainty-penalized optimal control problem assuming the learned dynamics. Previous works have used ensembles [8, 9], or likelihood-based generative models of data [24] to estimate this uncertainty.

The resulting open-loop planning problem can either be solved with first-order methods [47, 48], or zeroth-order sampling-based methods [44, 9] such as CEM [49] or MPPI [50]. While insights from stochastic optimization [51] tell us that first-order methods are more favorable in high dimensions and longer horizons as the variance of zeroth-order methods scale with dimension of decision variables [52], first-order methods require careful consideration of how amenable the uncertainty metric is to gradient-based optimization.

3 Distance to Data as a Metric of Uncertainty

In order to find an metric of uncertainty that is amenable for gradient-based offline optimization, we investigate smoothed distance to data as a candidate. We show that this metric has favorable properties that allows us to stably minimize uncertainty back to data using gradients. In addition, it allows us to quantify how much we underestimate true uncertainty by using the Lipschitz constant of error. All proofs for theorems in this section are included in Appendix A.

3.1 Properties of Distance to Data

We first formally define our proposed metric of uncertainty for offline model-based optimization.

Definition 3.1 (Distance to Data)**.** Consider a dataset $\mathcal{D} = \{x_i\}$ and an arbitrary point $x \in \mathbb{R}^n$. The standard squared distance from x to the set $\mathcal{D}$ can be written as $d(x; \mathcal{D})^2 = \min_{x_i \in \mathcal{D}} \frac{1}{2}\|x - x_i\|^2$. We define smoothed distance to data using a smoothed version of this standard squared distance,

$$d_\sigma(x; \mathcal{D})^2 := \mathsf{Softmin}_\sigma \tfrac{1}{2}\|x - x_i\|^2 + C = -\sigma^2 \log \left[\textstyle\sum_i \exp\left[-\frac{1}{2\sigma^2}\|x - x_i\|^2 \right] \right] + C, \tag{4}$$

where C is some constant to ensure positiveness of $d_\sigma(x; \mathcal{D})^2$, and $\sigma > 0$, also known as the *temperature parameter*, controls the degree of smoothing with $\sigma \to 0$ converging to the true min.

The motivation for introducing smoothing is to make the original non-smooth distance metric more amenable for gradient-based optimization [53, 54]. We now consider benefits of using $\mu(x) = d_\sigma(x; \mathcal{D})$ as our uncertainty metric, and show that as the smoothing level is annealed down, minimizing this distance allows us to converge to the points in the dataset.

Proposition 1 (Data Stability)**.** Consider a monotonically decreasing sequence σ_k such that $\sigma_k \to 0$, and denote x_k^n as the nth gradient descent iteration of $\min_x d_{\sigma_k}(x; \mathcal{D})^2$. Then, almost surely with random initialization and appropriate step size, we have

$$\lim_{k \to \infty} \lim_{n \to \infty} x_k^n \in \mathcal{D}. \tag{5}$$

We note that GPs show similar characteristics empirically [6]; however, empirical variance among ensembles [8] is prone to having local minima away from data due to statistical variations, especially

with small number of ensembles M, which makes them an unreliable metric to use for gradient-based optimization. Next, we show that it is possible to analyze how much softened distance to data underestimates true uncertainty using the Lipschitz constant of the model bias L_e.

Proposition 2 (Lipschitz Bounds)**.** Let L_e be the local Lipschitz constant of the true error (bias) $e(x) \coloneqq \|f(x) - f_\theta(x)\|_2$ valid over $\mathcal{Z} \subseteq \mathcal{X}$, where $\mathcal{X}$ is the domain of the input x. Then, $e(x)$ is bounded by

$$e(x) \leq e(x_c) + \sqrt{2} L_e \sqrt{d_\sigma(x; \mathcal{D})^2 + C_2}, \tag{6}$$

for all $x \in \mathcal{Z}$, where $x_c \coloneqq \arg\min_{x_i \in \mathcal{D}} \frac{1}{2}\|x - x_i\|_2^2$, i.e., the closest data-point, and $C_2 = \sigma^2 \log N - C$, where C is defined in (4).

In general, it is difficult to obtain L_e in the absence of more structured knowledge of f. However, it is possible to obtain confidence bounds on L_e using statistical estimation with pairwise finite slopes $\|e(x_i) - e(x_j)\| / \|x_i - x_j\|$ within the dataset [55, Ch. 3] [17] . We believe this offers benefits over ensembles as characterizing the convergence of neural network weights with randomly initialized points is far more complex to analyze. We compare distance to data to other uncertainty metrics in Figure 1 in simple 1D offline model-based optimization, where we show that ensembles [8, 9] have local minima outside of data, and unpredictably underestimates uncertainty due to model bias.

3.2 Estimating Gradients of Distance to Data with Score Matching

Although we have shown benefits of smoothed distance to data as an uncertainty metric amenable for gradient-based optimization, it is costly to compute at inference time as we need to iterate through the entire dataset. At the cost of losing guarantees of exact computation in Section 3.1, we consider ways to amortize this computation with function approximation. We first show the equivalence of the smoothed distance-to-data to the negative log likelihood of the *perturbed empirical distribution* [39], which applies randomized smoothing [53, 54] to the empirical distribution $\hat{p}(x; \mathcal{D})$.

Definition 3.2 (Perturbed Empirical Distribution)**.** Consider a dataset $\mathcal{D} = \{x_i\}$ and its corresponding empirical distribution $\hat{p}(x; \mathcal{D})$. We define $p_\sigma(x; \mathcal{D})$ as the noise-perturbed empirical distribution,

$$p_\sigma(x'; \mathcal{D}) \coloneqq \int \hat{p}(x; \mathcal{D}) \mathcal{N}(x'; x, \sigma^2 \mathbf{I}) dx = \frac{1}{N} \sum_{x_i \in \mathcal{D}} \mathcal{N}(x'; x_i, \sigma^2 \mathbf{I}). \tag{7}$$

Proposition 3. The negative log-likelihood of the perturbed empirical distribution $p_\sigma(x)$ is equivalent to smoothed distance to data by a factor of σ^2, up to some constant that does not depend on x,

$$-\sigma^2 \log p_\sigma(x; \mathcal{D}) = d_\sigma(x; \mathcal{D})^2 + C(N, n, \sigma). \tag{8}$$

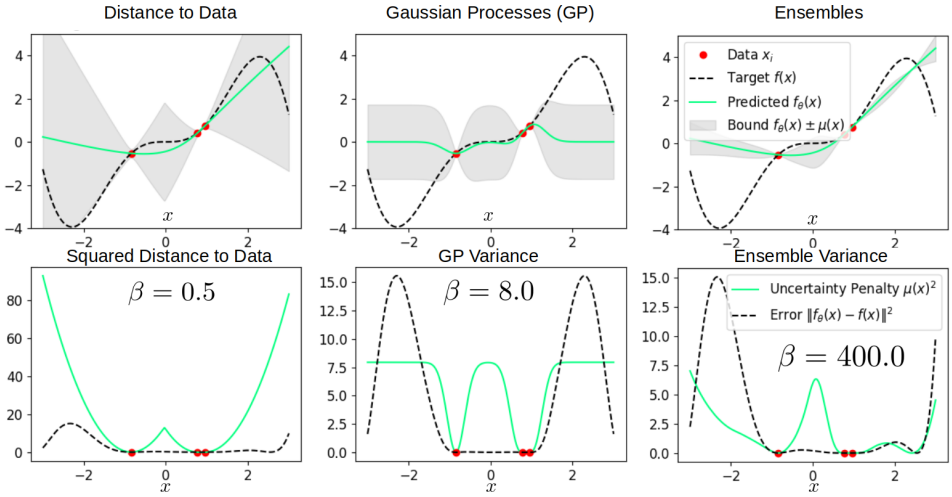

Figure 1: Comparison of different uncertainty metrics. Top row: Visualization of distance to data against GPs and ensembles with $M = 2$. Bottom row: Visualization of the penalty $\mu(x)^2$. All the metrics underestimate uncertainty to varying degrees, but distance to data can be more amenable for analysis; as we increase samples, distance to data will be able to bound the true uncertainty more closely by estimating Lipschitz constants using pairwise slopes in the dataset. In addition, distance to data shows more stability to data while ensembles have local minima outside of data.

This connection to the perturbed empirical distribution allows us to use generative modeling tools that utilize random perturbations of data [56, 39, 38], such as denoising autoencoders. However, computing likelihoods directly have proven to be difficult for high-dimensional data [39]. Additionally, even if such a likelihood-based generative model can give good likelihoods, there are no guarantees on the quality of its gradients, which defeats our purpose of finding a metric that is amenable to gradient-based optimization. Thus, we propose to use approaches that estimate the gradients of the perturbed empirical distribution (score function) directly [39, 38], which have shown promising performance in generative modeling [37] as it bypasses the estimation of the partition function.

Estimating the score function is a process known as *score-matching*. Following [39], we introduce *noise-conditioned score function* $s(x, \sigma) \coloneqq \nabla_x \log p_\sigma(x; \mathcal{D})$ [39] and aim to optimize the following objective given some sequence of annealed smoothing parameters σ_k,

$$\min_\theta \textstyle\sum_k \sigma_k^2 \mathbb{E}_{x \sim p_{\sigma_k}(x;\mathcal{D})} \left[\| s_\theta(x, \sigma_k) - \nabla_x \log p_{\sigma_k}(x; \mathcal{D}) \|^2 \right], \tag{9}$$

which has been shown to be equivalent to the denoising-score-matching loss [56]. Compared to explicitly computing $\nabla_x \log p_{\sigma_k}(x; \mathcal{D})$ which would require iterating through the entire dataset, the denoising loss allows us to learn the score function using batches of data.

$$\min_\theta \textstyle\sum_k \sigma_k^2 \mathbb{E}_{\substack{x \sim \hat{p}(x;\mathcal{D}) \\ x' \sim \mathcal{N}(x'; x, \sigma_k^2 \mathbf{I})}} \left[\| s_\theta(x'; \sigma_k) + \sigma_k^{-2}(x' - x) \|^2 \right]. \tag{10}$$

4 Planning with Gradients of Data Likelihood

Having illustrated benefits of using smoothed distance to data as an uncertainty metric, we now turn to the sequential decision-making setting of offline MBRL, where we use notation from Section 2.

Offline MBRL with Data Likelihood. Consider a learned dynamics model f_θ from the dataset $\mathcal{D} = \{(x_t, u_t, x_{t+1})_i\}$, as well as the perturbed empirical distribution $p_\sigma(x_t, u_t; \mathcal{D})$ of the (x_t, u_t) pairs in the dataset $\mathcal{D}$. Then, given some sequence of rewards r_t, we consider the following planning problem of maximizing both the reward and the likelihood of data,

$$\begin{aligned} \max_{x_{1:T}, u_{1:T}} \quad & \sum_{t=1}^{T} r_t(x_t, u_t) + \beta \sigma^2 \sum_{t=1}^{T} \log p_\sigma(x_t, u_t; \mathcal{D}) \\ \text{s.t.} \quad & x_{t+1} = f_\theta(x_t, u_t) \quad \forall t \in [1, T]. \end{aligned} \tag{11}$$

Note that $-\sigma^2 \log p_\sigma(x_t, u_t)$ is equivalent to smoothed distance to data $d_\sigma(x_t, u_t; \mathcal{D})^2$ as an objective, and acts as an uncertainty penalty, preventing the optimizer from exploiting o.o.d. solutions.

Score-Guided Planning (SGP). Given a differentiable r_t, we propose a single-shooting algorithm that rolls out f_θ iteratively and computes the gradient of the cumulative reward with respect to the input trajectory parameters $u_{1:T}$. We modify this gradient by adding the learned score function projected into the space of decision variables. In particular, the derivative of $\log p(x_i, u_i)$ at time i with respect to decision variable u_j at time j can be evaluated through the chain rule,

$$\frac{\partial}{\partial u_j} \log p_\sigma(x_i, u_i) = \frac{\partial}{\partial x_i} \log p_\sigma(x_i, u_i) \frac{\partial x_i}{\partial u_j} + \frac{\partial}{\partial u_i} \log p_\sigma(x_i, u_i) \frac{\partial u_i}{\partial u_j}. \tag{12}$$

Note that $\nabla_{x_i} \log p_\sigma(x_i, u_i)$ and $\nabla_{u_i} \log p_\sigma(x_i, u_i)$ can be obtained using the noise-conditioned score estimator $s_\theta(x, u, \sigma)$ in Section 3.2, where we extend the domain to include both x and u. After computing this gradient with reverse-mode autodiff (Appendix B.1), we call optimizers that accept gradient oracles, such as Adam [48]. We also note that this gradient computation can be extended to feedback policy gradients in Appendix B.4. Finally, we train a noise-conditioned score function for some sequence σ_k, and anneal the noise-level during optimization in order to ensure accurate score estimation [39], and leverage the result of Proposition 1.

Related Methods. LDM [24] is a closely-related MBRL method that solves a likelihood-constrained version of our problem, and uses flow models [36] to approximate the data likelihood. Many IL

methods are also related, as SGP with zero rewards can be used to imitate demonstration data by maximizing data likelihood [12] (Appendix B.5). In particular, AIRL [34] maximizes the rewardless version of our objective using GANs[33]. However, these methods primarily rely on likelihood-based generative models, and do not consider the interplay of generative modeling with gradient-based optimization. Diffuser [25, 57] shares similar methodologies with SGP, and solves a variant of Equation (11) with a quadratic-penalty-based approximation of direct transcription (Appendix B.3), which comes with benefits of numerical stability for long horizons [58, exercise 10.1], and robustness to sparse rewards [59]. Finally, Diffusion Policies [40] attempt to maximize the state-conditioned action sequence distribution, but is a behavior cloning method that does not account for rewards.

5 Empirical Results

We now test our proposed algorithm (SGP) empirically and show that it is an effective method for offline optimization that has scalable properties in high-dimensional problems by leveraging gradient-based optimization. All details for the included environments are presented in Appendix C.

Cart-pole System with Learned Dynamics. We apply our method to swing up a cart-pole model, which undergoes a long duration ($T = 60$) dynamic motion, which directly translates to high number of decision variables in single shooting. We first show the effects of model bias in in Fig. 2.C, where vanilla MBRL plans a trajectory that leaves the region (red box) of training data. In contrast, SGP keeps the trajectory within the training distribution (Fig.2.A). We further compare our method against the baselines of ensembles (Fig.2.B,E) and CEM [49] (Fig.2.D,E). To fairly compare CEM against SGP, we additionally train a data distance estimator that predicts the likelihoods directly based on explicit computation of the softmin distance in Equation (4) (Appendix C).

We observe that the convergence of CEM is much slower than first-order methods (Fig.2.F). However, we surprisingly obtain more asymptotic performance with CEM rather than using Adam for ensembles - we believe this signifies presence of local minima in gradient-based minimization of ensemble variance, as opposed to CEM which has some stochastic smoothing that allows it to escape local minima [52]. Finally, we note that unlike score functions, distance to data is considerably more difficult to train as we need to loop through the entire dataset to compute one sample. As a result, it is costly to train and we were unable to train it to good performance in a reasonable amount of time.

D4RL Benchmark. To evaluate our method against other methods on a standard benchmark, we use the MuJoCo [60] tasks in the D4RL [43] dataset with three different environments and sources of data. To turn our planner into a controller, we solve the planning problem with some finite horizon, rollout

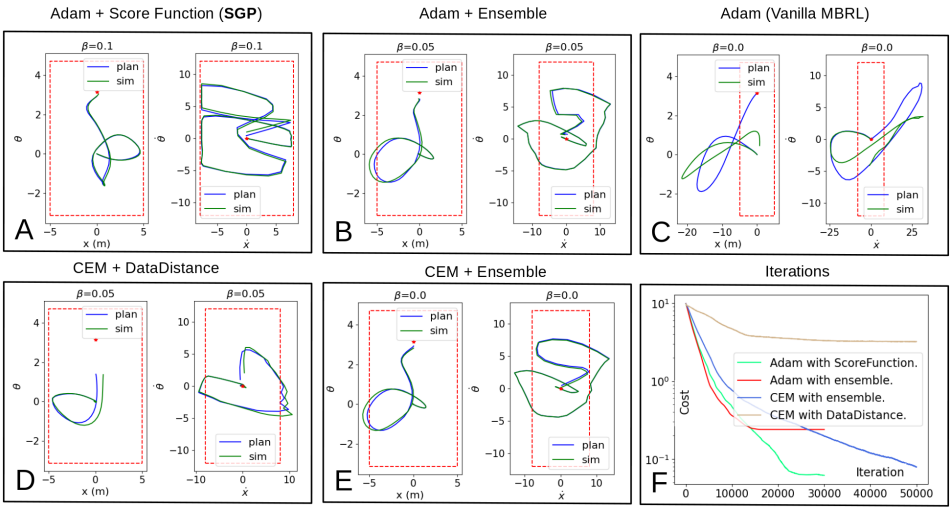

Figure 2: Experiments for the cart-pole system, where blue is the hallucinated plan and green is the actual rollout, and the red box is the regime of collected data. Note that SGP (**A**) results in a successful swingup while MBRL (**C**) fails. Compared to other baselines such as **B**,**D**,**E**, our method achieves much lower true cost with faster convergence (**F**).

Dataset	Environment	BC	MOPO	MBOP	SAC	CQL	Diffuser	Ours
Random	halfcheetah	2.1	**35.4**	30.7	30.5	**35.4**	-	31.3 $\pm$ 0.5
Random	hopper	1.6	**11.7**	4.5	11.3	10.8	-	**12.3** $\pm$ 1.3
Random	walker2d	9.8	13.6	8.6	4.1	7.0	-	**15.6** $\pm$ 7.1
Medium	halfcheetah	42.6	42.3	28.3	-4.3	44.0	42.8	**46.6** $\pm$ 2.8
Medium	hopper	52.9	28.0	4.9	48.8	58.5	**74.3**	60.6$\pm$ 5.8
Medium	walker2d	75.3	17.8	12.7	0.9	72.5	**79.6**	36.2$\pm$ 6.8
Med-Expert	halfcheetah	55.2	63.3	**105.9**	1.8	91.6	88.9	61.6 $\pm$ 2.4
Med-Expert	hopper	52.5	23.7	55.1	1.6	**105.4**	103.3	**109.2**$\pm$ 0.3
Med-Expert	walker2d	**107.5**	44.6	70.2	-0.1	**108.8**	**106.9**	103.0$\pm$ 2.8

Table 1: Comparing performance of our algorithm on the D4RL [43] dataset. Similar to [25, 63], we bold-face 95% of max performance. Ours is averaged from 5 trials. We emphasize that we outperform ensembles (MOPO) in many of the tasks.

the first optimal action u_1^*, and recompute the plan in a standard model-predictive control (MPC) [61] fashion. We compare against methods such as Behavior Cloning (BC) [21, 20], MOPO [8], MBOP [62], CQL [22], and Diffuser [25]. Our results demonstrate that our algorithm performs comparably to other state-of-the-art methods. On many of the tasks, we demonstrate better performance compared to MOPO [8] which uses ensembles for uncertainty estimation, while requiring less memory. This empirically supports our proposed benefits of using score matching for gradient-based offline MBRL. In addition, we outperform BC in many tasks, illustrating that we achieve better performance than pure imitation learning by incorporating rewards.

Pixel-Based Single Integrator. Pixel spaces have long been a challenge for MBRL due to the challenges of reliably controlling from images [64], learning pixel-space dynamics, and combating the resulting model-bias [2, 29, 65, 66]. To show scalability of our method, we present a 32×32 pixel-based single integrator environment where the action space is defined in pixel space, similar to a spatial action map [67]; see Fig. 3 for visuals. Our planning problem requires proposing a sequence of 32×32 control images, which optimizes reward (goal-reaching while maximizing running rewards) when rolled out through the dynamics (see Appendix C for details). We demonstrate that SGP capably mitigates model bias, generating plans consisting of plausible observation/controls near the data; thus, these plans are closely followed when executed open-loop under the true dynamics (Fig.3.A). When ignoring the effect of model bias and planning naïvely with MBRL (Fig.3.B), i.e., setting $\beta = 0$, the resulting trajectory exploits the subleties of the chosen reward function, planning an unrealistic trajectory which leads to poor rewards at runtime. Also, we show that ensembles are unable to stably converge back to the data manifold of images seen during training, instead getting trapped in local minima, planning unrealistic trajectories which translate to poor runtime rollouts on the true system (Fig.3.C). Moreover, due to the high dimensionality of this action space ($u_t \in \mathbb{R}^{32^2}$), we see that zeroth-order methods, such as CEM [49], lead to very low convergence rates (Fig.3.D) due to the large number of decision variables [51, 52]; again, this leads to poor reward seen at runtime.

Box-Pushing with Learned Marker Dynamics. To validate our method on the data-scarce regime of hardware, we prepare a box-pushing task from [44] where we leverage the quasistatic assumption [68] and treat the positions of the motion capture markers as if they are states. An interesting feature

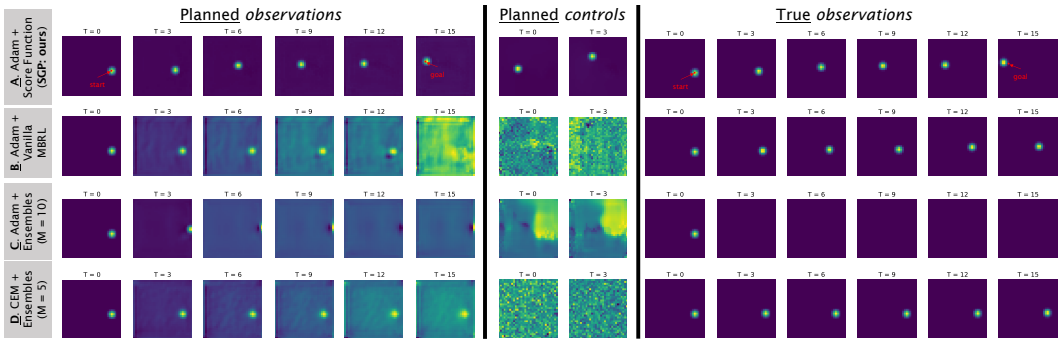

Figure 3: Planned (left) and true (right) rollouts of observations for the pixel single-integrator with pixel action space (center). Note that SGP (**A**) significantly outperforms Vanilla MBRL (**B**), Adam + Ensembles (**C**), and CEM + Ensembles (**D**), which are unable to combat model bias during rollout.

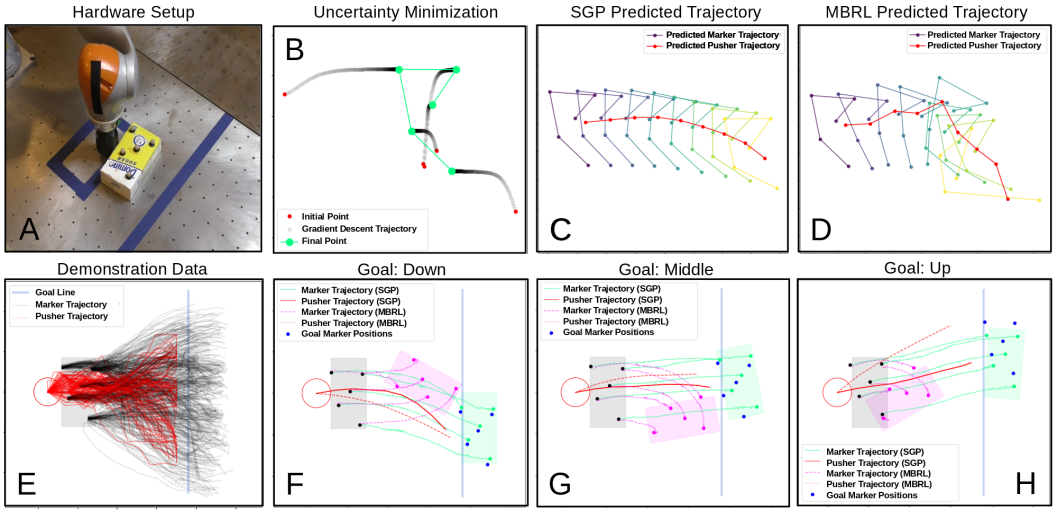

Figure 4: Visualization of the box-pushing experiment in **A**, where data was collected from 100 demonstrations illustrated in **E**. We show that SGP successfully stabilizes markers back to their implicit constraints (**B**), which allows better predictions (**C**) than vanilla MBRL (**D**). We additionally use the reward to change the goal into three different regions of down (**F**), middle (**G**), and up (**H**), where vanilla MBRL fails, and behavior cloning is not capable of.

of this setup is that the markers implicitly live on a constraint manifold where the distance between each marker is fixed; we ask if using score matching can stabilize the rollouts of prediction to obey this implicit constraint, similar to how diffusion stabilizes back to the data manifold [41]. To test this, we collect about 100 demonstrations of the box being pushed to different positions, which lead to 750 samples of data tuples (x_t, u_t, x_{t+1}). We aim to show that i) SGP can enforce implicit constraints within the data, ii) distance to data acts as a successful uncertainty metric in data-scarce offline MBRL, and iii) we can use the reward to change the task from imitation data.

Our results in Fig.4 demonstrate that SGP successfully imposes implicit constraints on the data. Not only does minimization of uncertainty in the absence of rewards result in stabilization to the marker position constraints (Fig.4.B), but the rollouts also become considerably more stable when we use the distance to data penalty (Fig.4.C). In contrast, MBRL with $\beta = 0$ destroys the keypoint structure as dynamics are rolled out, resulting in suboptimal performance (Fig.4.D). Finally, we demonstrate through Fig.4.F,G,H that we can use rewards to show goal-driven behavior to various goals from a single set of demonstration data, which behavior cloning is not capable of.

6 Conclusion and Discussion of Limitations

We proposed SGP, which is a *first-order* offline MBRL planning algorithm that learns gradients of distance to data with score-matching techniques, and solves planning problems that jointly maximize reward and data likelihood. Through empirical experiments, we showed that SGP beats baselines of zeroth-order methods and ensembles, has comparable performance with state-of-art offline RL algorithms, and scales to pixel-space action spaces with up to $15,360$ decision variables for planning.

We conclude with listing some limitations of our approach. Unlike ensembles, our method by construction discourages extrapolation, which can be a limitation when the networks jointly recover meaningful inductive bias. We also believe that computation is a current bottleneck for realtime application of our MPC, which can take between 1-2 seconds per iteration due to gradient computations. We believe formulating our problem as policy search (Appendix B.4) might lead to considerable amortization of computation at inference time. Our algorithm is additionally subject to the shortcomings of single shooting, such as gradient instability for long horizons [52], or not being able to handle sparse rewards due to myopicness. We plan to further explore a direct transcription version of our problem by leveraging our connection to existing work [25]. Finally, we have not yet investigated the performance of our method in the presence of aleatoric uncertainty.

References

[1] C. Atkeson and J. Santamaria. A comparison of direct and model-based reinforcement learning. In Proceedings of International Conference on Robotics and Automation, volume 4, pages 3557–3564 vol.4, 1997. doi:10.1109/ROBOT.1997.606886.

[2] H. J. T. Suh and R. Tedrake. The surprising effectiveness of linear models for visual foresight in object pile manipulation. In S. M. LaValle, M. Lin, T. Ojala, D. Shell, and J. Yu, editors, Algorithmic Foundations of Robotics XIV, pages 347–363, Cham, 2021. Springer International Publishing. ISBN 978-3-030-66723-8.

[3] M. Watter, J. T. Springenberg, J. Boedecker, and M. Riedmiller. Embed to control: A locally linear latent dynamics model for control from raw images, 2015.

[4] K. Chua, R. Calandra, R. McAllister, and S. Levine. Deep reinforcement learning in a handful of trials using probabilistic dynamics models, 2018.

[5] J. Schneider. Exploiting model uncertainty estimates for safe dynamic control learning. In M. Mozer, M. Jordan, and T. Petsche, editors, Advances in Neural Information Processing Systems, volume 9. MIT Press, 1996. URL https://proceedings.neurips.cc/paper_files/paper/1996/file/93fb9d4b16aa750c7475b6d601c35c2c-Paper.pdf.

[6] M. Deisenroth and C. Rasmussen. Pilco: A model-based and data-efficient approach to policy search. pages 465–472, 01 2011.

[7] P. Abbeel, M. Quigley, and A. Y. Ng. Using inaccurate models in reinforcement learning. In Proceedings of the 23rd International Conference on Machine Learning, ICML '06, page 1–8, New York, NY, USA, 2006. Association for Computing Machinery. ISBN 1595933832. doi:10.1145/1143844.1143845. URL https://doi.org/10.1145/1143844.1143845.

[8] T. Yu, G. Thomas, L. Yu, S. Ermon, J. Zou, S. Levine, C. Finn, and T. Ma. Mopo: Model-based offline policy optimization, 2020.

[9] A. Nagabandi, K. Konolige, S. Levine, and V. Kumar. Deep dynamics models for learning dexterous manipulation. In L. P. Kaelbling, D. Kragic, and K. Sugiura, editors, 3rd Annual Conference on Robot Learning, CoRL 2019, Osaka, Japan, October 30 - November 1, 2019, Proceedings, volume 100 of Proceedings of Machine Learning Research, pages 1101–1112. PMLR, 2019. URL http://proceedings.mlr.press/v100/nagabandi20a.html.

[10] R. Kidambi, A. Rajeswaran, P. Netrapalli, and T. Joachims. Morel : Model-based offline reinforcement learning, 2021.

[11] S. Fujimoto, D. Meger, and D. Precup. Off-policy deep reinforcement learning without exploration, 2019.

[12] S. K. S. Ghasemipour, R. Zemel, and S. Gu. A divergence minimization perspective on imitation learning methods, 2019.

[13] A. Malik, V. Kuleshov, J. Song, D. Nemer, H. Seymour, and S. Ermon. Calibrated model-based deep reinforcement learning. In K. Chaudhuri and R. Salakhutdinov, editors, Proceedings of the 36th International Conference on Machine Learning, volume 97 of Proceedings of Machine Learning Research, pages 4314–4323. PMLR, 09–15 Jun 2019. URL https://proceedings.mlr.press/v97/malik19a.html.

[14] V. Kuleshov, N. Fenner, and S. Ermon. Accurate uncertainties for deep learning using calibrated regression, 2018.

[15] Y. Gal and Z. Ghahramani. A theoretically grounded application of dropout in recurrent neural networks, 2016.

[16] J. Kocijan, R. Murray-Smith, C. Rasmussen, and A. Girard. Gaussian process model based predictive control. In Proceedings of the 2004 American Control Conference, volume 3, pages 2214–2219 vol.3, 2004. doi:10.23919/ACC.2004.1383790.

[17] C. Knuth, G. Chou, N. Ozay, and D. Berenson. Planning with learned dynamics: Probabilistic guarantees on safety and reachability via lipschitz constants. IEEE Robotics Autom. Lett., 6(3): 5129–5136, 2021.

[18] T. Kurutach, I. Clavera, Y. Duan, A. Tamar, and P. Abbeel. Model-ensemble trust-region policy optimization, 2018.

[19] Y. Wu, G. Tucker, and O. Nachum. Behavior regularized offline reinforcement learning, 2019.

[20] S. Ross, G. J. Gordon, and J. A. Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning, 2011.

[21] D. Pomerleau. Alvinn: An autonomous land vehicle in a neural network. In D. Touretzky, editor, Proceedings of (NeurIPS) Neural Information Processing Systems, pages 305 – 313. Morgan Kaufmann, December 1989.

[22] A. Kumar, A. Zhou, G. Tucker, and S. Levine. Conservative q-learning for offline reinforcement learning, 2020.

[23] N. Jaques, A. Ghandeharioun, J. H. Shen, C. Ferguson, A. Lapedriza, N. Jones, S. Gu, and R. Picard. Way off-policy batch deep reinforcement learning of implicit human preferences in dialog, 2019.

[24] K. Kang, P. Gradu, J. J. Choi, M. Janner, C. Tomlin, and S. Levine. Lyapunov density models: Constraining distribution shift in learning-based control. In K. Chaudhuri, S. Jegelka, L. Song, C. Szepesvari, G. Niu, and S. Sabato, editors, Proceedings of the 39th International Conference on Machine Learning, volume 162 of Proceedings of Machine Learning Research, pages 10708–10733. PMLR, 17–23 Jul 2022. URL https://proceedings.mlr.press/v162/kang22a.html.

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

[26] J. Peters, K. Muelling, and Y. Altun. Relative entropy policy search. In Proceedings of 24th AAAI Conference on Artificial Intelligence (AAAI '10), pages 1607 – 1612, July 2010.

[27] G. Chou, N. Ozay, and D. Berenson. Model error propagation via learned contraction metrics for safe feedback motion planning of unknown systems. In 2021 60th IEEE Conference on Decision and Control (CDC), Austin, TX, USA, December 14-17, 2021, pages 3576–3583. IEEE, 2021.

[28] C. Knuth, G. Chou, J. Reese, and J. Moore. Statistical safety and robustness guarantees for feedback motion planning of unknown underactuated stochastic systems. ICRA, 2023.

[29] G. Chou, N. Ozay, and D. Berenson. Safe output feedback motion planning from images via learned perception modules and contraction theory. In Algorithmic Foundations of Robotics XV - Proceedings of the Fifteenth Workshop on the Algorithmic Foundations of Robotics, WAFR 2022, College Park, MD, USA, 22-24 June, 2022, volume 25 of Springer Proceedings in Advanced Robotics, pages 349–367. Springer, 2022.

[30] D. P. Kingma and M. Welling. Auto-encoding variational bayes, 2022.

[31] D. Ha and J. Schmidhuber. Recurrent world models facilitate policy evolution, 2018.

[32] J. D. Co-Reyes, Y. Liu, A. Gupta, B. Eysenbach, P. Abbeel, and S. Levine. Self-consistent trajectory autoencoder: Hierarchical reinforcement learning with trajectory embeddings, 2018.

[33] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial networks, 2014.

[34] J. Fu, K. Luo, and S. Levine. Learning robust rewards with adversarial inverse reinforcement learning, 2018.

[35] J. Ho and S. Ermon. Generative adversarial imitation learning, 2016.

[36] D. Rezende and S. Mohamed. Variational inference with normalizing flows. In F. Bach and D. Blei, editors, Proceedings of the 32nd International Conference on Machine Learning, volume 37 of Proceedings of Machine Learning Research, pages 1530–1538, Lille, France, 07–09 Jul 2015. PMLR. URL https://proceedings.mlr.press/v37/rezende15.html.

[37] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer. High-resolution image synthesis with latent diffusion models, 2022.

[38] J. Sohl-Dickstein, E. A. Weiss, N. Maheswaranathan, and S. Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics, 2015.

[39] Y. Song and S. Ermon. Generative modeling by estimating gradients of the data distribution, 2020.

[40] C. Chi, S. Feng, Y. Du, Z. Xu, E. Cousineau, B. Burchfiel, and S. Song. Diffusion policy: Visuomotor policy learning via action diffusion, 2023.

[41] F. Permenter and C. Yuan. Interpreting and improving diffusion models using the euclidean distance function, 2023.

[42] F. Latorre, P. Rolland, and V. Cevher. Lipschitz constant estimation of neural networks via sparse polynomial optimization, 2020.

[43] J. Fu, A. Kumar, O. Nachum, G. Tucker, and S. Levine. D4rl: Datasets for deep data-driven reinforcement learning, 2021.

[44] L. Manuelli, Y. Li, P. Florence, and R. Tedrake. Keypoints into the future: Self-supervised correspondence in model-based reinforcement learning, 2020.

[45] B. Trabucco, X. Geng, A. Kumar, and S. Levine. Design-bench: Benchmarks for data-driven offline model-based optimization, 2022.

[46] B. Trabucco, A. Kumar, X. Geng, and S. Levine. Conservative objective models for effective offline model-based optimization, 2021.

[47] Y. Li, J. Wu, R. Tedrake, J. B. Tenenbaum, and A. Torralba. Learning particle dynamics for manipulating rigid bodies, deformable objects, and fluids, 2019.

[48] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization, 2017.

[49] P.-T. de Boer, D. P. Kroese, S. Mannor, and R. Y. Rubinstein. A tutorial on the cross-entropy method. Annals of Operations Research, 134(1):19–67, Feb 2005. ISSN 1572-9338. doi: 10.1007/s10479-005-5724-z. URL https://doi.org/10.1007/s10479-005-5724-z.

[50] G. Williams, A. Aldrich, and E. Theodorou. Model predictive path integral control using covariance variable importance sampling, 2015.

[51] S. Ghadimi and G. Lan. Stochastic first- and zeroth-order methods for nonconvex stochastic programming, 2013.

[52] H. J. T. Suh, M. Simchowitz, K. Zhang, and R. Tedrake. Do differentiable simulators give better policy gradients?, 2022.

[53] J. C. Duchi, P. L. Bartlett, and M. J. Wainwright. Randomized smoothing for stochastic optimization, 2012.

[54] H. J. T. Suh, T. Pang, and R. Tedrake. Bundled gradients through contact via randomized smoothing, 2022.

[55] S. Coles. An introduction to statistical modeling of extreme values. Springer Series in Statistics. Springer-Verlag, London, 2001. ISBN 1-85233-459-2.

[56] P. Vincent. A connection between score matching and denoising autoencoders. Neural Computation, 23(7):1661–1674, 2011. doi:10.1162/NECO_a_00142.

[57] A. Ajay, Y. Du, A. Gupta, J. Tenenbaum, T. Jaakkola, and P. Agrawal. Is conditional generative modeling all you need for decision-making?, 2022.

[58] R. Tedrake. Underactuated Robotics. 2023. URL https://underactuated.csail.mit.edu.

[59] O. Rybkin, C. Zhu, A. Nagabandi, K. Daniilidis, I. Mordatch, and S. Levine. Model-based reinforcement learning via latent-space collocation, 2021.

[60] E. Todorov, T. Erez, and Y. Tassa. Mujoco: A physics engine for model-based control. In 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems, pages 5026–5033, 2012. doi:10.1109/IROS.2012.6386109.

[61] J. Richalet, A. Rault, J. Testud, and J. Papon. Model predictive heuristic control: Applications to industrial processes. Automatica, 14(5):413–428, 1978. ISSN 0005-1098. doi:https://doi.org/10.1016/0005-1098(78)90001-8. URL https://www.sciencedirect.com/science/article/pii/0005109878900018.

[62] A. Argenson and G. Dulac-Arnold. Model-based offline planning, 2021.

[63] I. Kostrikov, A. Nair, and S. Levine. Offline reinforcement learning with implicit q-learning, 2021.

[64] G. Chou and R. Tedrake. Synthesizing stable reduced-order visuomotor policies for nonlinear systems via sums-of-squares optimization. CoRR, abs/2304.12405, 2023.

[65] C. Finn and S. Levine. Deep visual foresight for planning robot motion, 2017.

[66] D. Hafner, J. Pasukonis, J. Ba, and T. Lillicrap. Mastering diverse domains through world models, 2023.

[67] A. Zeng, P. Florence, J. Tompson, S. Welker, J. Chien, M. Attarian, T. Armstrong, I. Krasin, D. Duong, A. Wahid, V. Sindhwani, and J. Lee. Transporter networks: Rearranging the visual world for robotic manipulation, 2022.

[68] T. Pang, H. J. T. Suh, L. Yang, and R. Tedrake. Global planning for contact-rich manipulation via local smoothing of quasi-dynamic contact models, 2023.

[69] J. D. Lee, M. Simchowitz, M. I. Jordan, and B. Recht. Gradient descent only converges to minimizers. In Conference on learning theory, pages 1246–1257. PMLR, 2016.

[70] A. Braides. Gamma-Convergence for Beginners. Oxford University Press, 07 2002. ISBN 9780198507840. doi:10.1093/acprof:oso/9780198507840.001.0001. URL https://doi.org/10.1093/acprof:oso/9780198507840.001.0001.

[71] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Köpf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala. Pytorch: An imperative style, high-performance deep learning library, 2019.

[72] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In Medical Image Computing and Computer-Assisted Intervention - MICCAI 2015 - 18th International Conference Munich, Germany, October 5 - 9, 2015, Proceedings, Part III, volume 9351 of Lecture Notes in Computer Science, pages 234–241. Springer, 2015.

[73] Y. Song and S. Ermon. Improved techniques for training score-based generative models. In NeurIPS 2020, 2020.