Towards Diverse Manipulation Sampling

Marc Toussaint[1] Shiping Ma[1] Cornelius V. Braun[1]

***Abstract*— Generating diverse samples under hard constraints is a core challenge in many areas. In this work we discuss *NLP Sampling* as an integrative view and framework to combine methods from the fields of MCMC, constrained optimization, as well as robotics. While NLP Sampling is a general method for constrained sampling, we demonstrate it here for sampling manipulation strategies. We discuss such model-based diverse manipulation sampling as an alternative to data collection via human demonstration.**

I. Introduction

How could robotic systems acquire diverse manipulation and physical reasoning skills, eventually covering any manipulation that is physically possible? Current research builds to a large degree on learning from data collected from or by humans. While recent large-scale dataset [3] indeed cover a wide diversity of manipulations, which may be sufficient to train application-relevant robotic skills, from a scientific perspective the idea that general purpose manipulation skills originate purely from imitating human demonstrations seems limited.

We consider model-based data generation as a promising novel approach towards diverse manipulation and physical reasoning skills. By model-based data generation we mean a *privileged* setting, where the method has full information on the state and environment, including all shapes, articulations, and physical parameters. The aim then is to generate diverse demonstrations of what is possible in such an environment – which may be the basis for training sensor-based student policies to imitate such skills.

In this abstract we discuss a novel problem formulation, *NLP sampling* [4], as a framework for model-based diverse manipulation sampling. We briefly describe a family of NLP samplers that combine existing building blocks from the fields of nonlinear programming, MCMC and manifold-RRTs for effective NLP sampling, and then demonstrate it on manipulation problems. We discuss related work and outlook at the end.

II. NLP Sampling and a Family of Basic Samplers

We summarize details provided in [4]. NLP Sampling is defined as the problem of sampling

$$x \sim \exp(-f(x)) \quad \text{s.t.} \quad g(x) \le 0,\ h(x) = 0 \ , \qquad (1)$$

for a given energy $f : \mathbb{R}^n \to \mathbb{R}$ and constraint functions $g : \mathbb{R}^n \to \mathbb{R}^m$, $h : \mathbb{R}^n \to \mathbb{R}^{m'}$, which are assumed smoothly differentiable. To clarify the notation, we sample

*This work was supported by Amazon Research Germany GmbH

[1]Learning & Intelligent Systems Lab, TU Berlin toussaint@tu-berlin.de

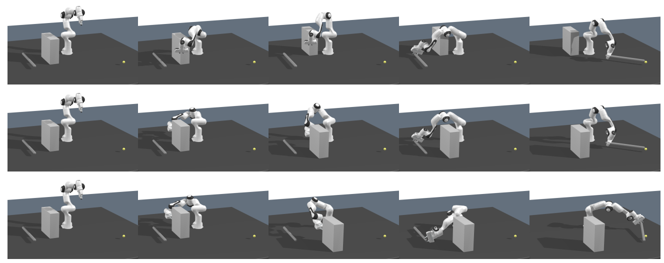

Fig. 1: Three samples from an NLP that formalizes a 4-step sequential manipulation problem where a box first has to be pushed aside to be able to reach for a stick with which then a point can be reached. This problem has a large number of local optima – ordinary optimizers get stuck in infeasible local optima with very high chance and fail to systematically explore the feasible space.

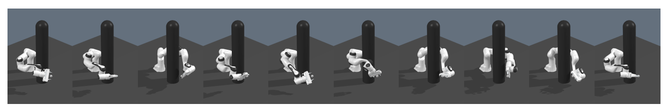

Fig. 2: Ten samples from an NLP that encodes a standard Inverse Kinematics problem for reaching a point around an obstacle. Note the diversity not only in left/right, but also the grasp poses.

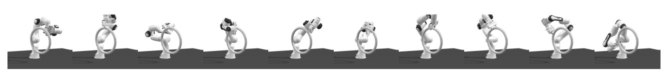

Fig. 3: Ten samples from an NLP that encodes a standard Inverse Kinematics problem for grasping a torus.

from $p(x) \propto \exp(-f(x))\ \mathbb{I}_{g(x)\le 0}\ \mathbb{I}_{h(x)=0}$, with indicator functions $\mathbb{I}$ and unknown partition function. Therefore, our problem specification is a standard nonlinear program (NLP), as for optimization, but instead of generating a single (optimal) solution we aim to sample diversely from the feasible set, potentially biased by costs.

Note that with only equality constraints, the problem becomes sampling on a non-linear (differentiable) constrained manifold, and with only linear inequalities, the problem becomes sampling from a polytope – efficient methods should therefore incorporate ideas from existing specialized samplers for these cases. As is evident from these special cases, efficient NLP Sampling necessitates leveraging evaluations of the constraint functions $g(x), h(g)$ along with their gradients. This differentiates it from naively applying MCMC methods directly on $p(x)$, which has zero probability and score function in the infeasible space.

The NLP sampling problem can be decomposed in two aspects: First finding an interior point (aka. "Phase I optimization" in the context of nonlinear programming), and then sampling correctly from $\propto \exp(-f(x))$ in the feasible space. We propose a family of samplers that combines well-established building blocks from nonlinear programming, MCMC, and sample-based manifold exploration into a two-phase approach. Specifically, a *Restarting Two-Phase NLP Sampler* iterates the following steps until enough samples are collected:

1) Sample a new seed x, potentially conditional to previously found samples D.
2) A **slack downhill** method tries to find a feasible point x within K_{down} steps.
3) If x is feasible, an **interior sampling** method, potentially requiring K_{burn} burn-in steps, tries to collect K_{sam} samples $\sim e^{-f(x)}$ within the feasible space. If the method does not respect feasibility exactly, it is combined with a direct slack reduction (manifold projection) step.

While straight-forward, we are not aware of sampling techniques to adopt this basic two-phase approach. We propose this family as a generic framework to integrate methods from across the fields of MCMC sampling, constrained optimization, and robotics.

More specifically (details described in [4]) we evaluated a range of options for the slack downhill phase, namely:

- Plain gradient descent or Gauss-Newton on the slack
- The above but with noise (which is Langevin, or Riemannian Langevin [2])
- The above, with options for step rejection (1st Wolfe condition, or Metropolis-Hasting)

And we also evaluated a range of options for interior sampling:

- Metropolis-adjusted Langevin (MALA) and Riemannian Langevin
- Metropolis-adjusted nonlinear Hit-And-Run (a novel method)
- Manifold RRT (for equalities only)

Finally, in [4] we also considered several options for informed restarting, namely restarting at points remote from previous samples. But surprisingly, these had, in our evaluations, little effect on the diversity of the resulting samples.

III. Minimum Spanning Tree Score (MSTS) as novel Diversity Metric

All the proposed samplers use Metropolis-adjustment and are therefore "correct" by construction – but only within a connected feasible region! Recall that Metropolis-Hastings ensures that MCMC samplers (after a mixing time) correctly sample from an underlying energy, but only under the assumption of irreducibility (essentially that any transition has non-zero probability in the proposal) [1]. However, our interior samplers are by construction confined to the feasible region that was found in phase one, and mixing times may become long when this feasible region is highly non-linear (as is the case with complex manifolds). When evaluating samplers we are therefore primarily interested in the ability to discover diverse feasible regions, as well as short mixing times. In other terms we are interested in the *diversity per compute effort*.

We found the size of the minimum spanning tree of all points in the generated samples D an interesting quantity that does capture also distances between covered feasible regions, as at least one edge needs to connected distant covered regions. Specifically, we define the Minimum Spanning Tree Score $\text{MSTS}_p(D)$ as the total cost of the tree when using edge costs $|x - x'|^p$. When edge costs are plain Euclidean distances, $p = 1$, it is guaranteed that $\text{MSTS}_1(D)$ strictly increases with $n = |D|$ when adding points. With $p > 1$, the $\text{MSTS}_p(D)$ does not monotonously increase with $|D|$ when adding points, but rather converge to an indicator of separateness of modes.

IV. Results

Figures 1-3 illustrate samples from NLPs that encode manipulation problems. Esp. the first problem is an instance where ordinary optimization methods (which we used in previous work on force-based Task-and-Motion Planning [5] get stuck in local optima with high frequency, thereby compromising the probabilistic completeness of overall planners. In contrast, NLP sampling can systematically sample diversely from the constraint manifold, exploring all possible pushes and placements of the box and grasp points on the stick in this case.

In [4] we provide an extensive quantitative study on the *diversity per compute effort* that a Two-Phase NLP Sampler with the alternative building blocks can achieve.

V. Conclusions

With this abstract we proposed using diverse sampling methods for model-based data generation of robotic manipulation. NLP sampling is a first methdological approach we have developed, which uses formulates manipulation problems as a nonlinear program (assuming full model knowledge), and allows us to use a combination of optimization, restarting, and MCMC methods to generate solutions. Our future research will aim to more broadly apply this to increasingly more complex and realistic robotic manipulation scenarios, as well as evaluate distilling such data into sensor-based reactive policies.

References

[1] C. Andrieu, N. De Freitas, A. Doucet, and M. I. Jordan. An introduction to MCMC for machine learning. *Machine Learning*, 50(1/2):5–43, 2003.

[2] M. Girolami and B. Calderhead. Riemann manifold Langevin and Hamiltonian Monte Carlo methods. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 73(2):123–214, 2011.

[3] Open X.-Embodiment Collaboration. Open X-Embodiment: Robotic Learning Datasets and RT-X Models, June 2024.

[4] M. Toussaint, C. V. Braun, and J. Ortiz-Haro. NLP Sampling: Combining MCMC and NLP Methods for Diverse Constrained Sampling, 2024.

[5] M. Toussaint, J.-S. Ha, and D. Driess. Describing physics for physical reasoning: Force-based sequential manipulation planning. *IEEE Robotics and Automation Letters*, 2020.