

# Efficient Gradient-Based Inference for Manipulation Planning in Contact Factor Graphs

Jeongmin Lee, Sunkyung Park, Minji Lee, and Dongjun Lee

**Abstract**—This paper presents a framework designed to tackle a range of planning problems arise in manipulation, which typically involve complex geometric-physical reasoning related to contact and dynamic constraints. We introduce the Contact Factor Graph (CFG) to graphically model these diverse factors, enabling us to perform inference on the graphs to approximate the distribution and sample appropriate solutions. We propose a novel approach that can incorporate various phenomena of contact manipulation as differentiable factors, and develop an efficient inference algorithm for CFG that leverages this differentiability along with the conditional probabilities arising from the structured nature of contact. Our results demonstrate the capability of our framework in generating viable samples and approximating posterior distributions for various manipulation scenarios.

## I. INTRODUCTION

Manipulation planning is a crucial and essential problem for robots to achieve their task goals. Given that typical manipulation tasks are executed through contact, the problem often arises various range of inverse problems over contact mechanics and structural dynamics, mostly requiring geometric-physical reasoning.

This reasoning in manipulation planning typically involves considering a variety of factors, resulting in a non-convex, multi-modal search space. One natural perspective for addressing these problems is to perform inference on the posterior distribution of the variables [1], [2]. By framing the problem this way, one can apply diverse probabilistic inference methods, such as optimization, Markov Chain Monte Carlo (MCMC) or variational inference, to explore the posterior distribution and identify the most probable solutions. This approach has shown successful demonstrations in recent work on motion planning [3]–[5]. However, in manipulation planning, the inherent complexity of contact interactions presents various challenges, such as creating a tractable formulation, developing scalable inference algorithms, and generating data using sampling methods. Consequently, finding an efficient way to seek a diverse set of solutions within the distribution remains elusive.

Given this context, in this work, we introduce a new modeling and inference algorithm to address diverse reasoning arise in manipulation planning, leveraging the differentiable

representation and inherent structure of the contact mechanics. We first propose a formulation of the problem through a graphical model, termed Contact Factor Graphs (CFG). This model effectively organizes the relationships between object poses and contact forces, incorporating a variety of contact and dynamic factors. We describe how each factor can be formulated in *differentiable* manner and demonstrate that various reasoning problems for manipulation can be represented within this framework. Then, we introduce a novel gradient-based inference within the CFG. By capitalizing on the structured nature of the contact factors, we explain the advantages that can be obtained when using conditional probabilities in the inference process. We demonstrate the derivation of an efficient score function by exploiting the differentiability of the factors and the envelope theorem. The proposed framework is demonstrated to be capable of effective sample generation and approximation of the posterior probability distribution through diverse examples.

## II. RELATED WORKS

### A. Planning as Inference

Formulating and solving problems using graphical models and inference provides a structured approach with a stochastic perspective. For instance, [2] offers an overview of the inference perspective in control problems, while [6] elaborates on the utility of factor graphs in various robotic applications. Addressing the continuous motion planning challenge, [3] advocates for leveraging this perspective through factor graph modeling, accompanied by maximum-a-posteriori (MAP) inference via optimization techniques. Uraín et al. [7] construct a reactive policy by sampling from a multimodal posterior distribution using the cross-entropy method. The effectiveness of Bayesian inference in trajectory optimization is demonstrated in [8], [9]. In [10], geometric and physical constraint satisfaction problems are modeled as graphs and addressed through diffusion-based inference techniques. One advantage of such graphical models is their compositional nature. Our graphical model also employs contact-related constraints in a compositional manner but places greater emphasis on a model-based, differentiable energy function.

### B. Problems with Contact

For problems involving contact, the inference approach introduces challenges as they involve constraints not only on the planning state but also on the contact impulse as non-linear complementarity [11]. This may necessitate searching over hybrid modes [12], such as mixed-integer programming

This research was supported by the National Research Foundation (NRF) grant funded by the Ministry of Science and ICT (MSIT) of Korea (RS-2022-00144468), and the Ministry of Trade, Industry & Energy (MOTIE) of Korea (RS-2024-00419641). Corresponding author: Dongjun Lee.

The authors are with the Department of Mechanical Engineering, IAMD and IOER, Seoul National University, Seoul, Republic of Korea. {ljmlgh,sunk1136,mingg8,djlee}@snu.ac.kr.

[13], which is typically computationally expensive. From an optimization perspective, contact-implicit constraints can make the problem mode-invariant, as demonstrated in [14]–[16]. Recent works have focused more on improving the tractability and utility of the formulations. Some reflections on a “good” model for physical reasoning are discussed in [17]. This is also related to the recent emergence of differentiable simulators [18]–[21], which can provide gradients to help solve inverse problems involving contact physics. Several studies have demonstrated their efficacy compared to black-box sampling and inference methods commonly used in data generation processes and deep reinforcement learning [22]–[24]. Our work aligns with this trend, yet, instead of directly differentiating the simulation, we propose to form a factor graph with differentiable contact factors, which offers a broader perspective on modeling and facilitates tailored gradient-based inference.

### III. MODELING CONTACT FACTOR GRAPHS

#### A. Problem Formulation

In this work, we primarily address the challenge of generating solution sets for geometric and physical constraints satisfaction encountered in contact-based tasks, assuming that the task skeleton is provided by high-level discrete search. Also we base our analysis on given geometric and physical parameters, focusing on the variable  $X = \{q, u, \lambda\}$ , where  $q$ ,  $u$ , and  $\lambda$  stand for the configuration space, external input, and contact force, respectively. The constraints among these variables are modeled as factors, which can be visualized through a graphical model. In the following section, we detail the types of factors we primarily consider.

#### B. Contact Mechanics

Incorporating contact constraints into the factor graph is essential and has significant implications. Here, we present a range of constraints involved in contact mechanics. One fundamental property of contact is its unilateral nature, resulting in the following constraints:

$$\begin{aligned} g(q) &\geq 0 \quad (\text{non-penetration}) \\ g(q)\lambda &= 0 \quad (\text{complementarity}) \end{aligned} \quad (1)$$

where  $g(q) \in \mathbb{R}$  is the gap function between objects. Another crucial aspect of contact mechanics is the Coulomb friction law, which we precisely define as follows:

$$\mathcal{C}_{n(q)} \ni \mu\lambda_n + \lambda_t \perp \beta n(q) + v_t \in \mathcal{C}_{n(q)} \quad (\text{Coulomb}) \quad (2)$$

Here, the subscript  $n, t$  denote normal/tangential,  $\mu \in \mathbb{R}^+$  is the friction coefficient,  $\beta \in \mathbb{R}$  is an auxiliary variable,  $n(q)$  is the contact normal,  $\mathcal{C}_{n(q)}$  is a second-order cone with  $n(q)$  as its axis. The normal and tangential components of  $\lambda$  are expressed as:

$$\lambda_n = n(q)n(q)^T \lambda \quad \lambda_t = (I - n(q)n(q)^T) \lambda$$

Also the contact Jacobian  $J_c$  and the tangential velocity  $v_t$  are defined as<sup>1</sup>

$$\begin{aligned} J_c &= [I \quad -[c(q)]_\times] \\ v_t &= (I - n(q)n(q)^T)J_c \delta q \end{aligned}$$

where  $c(q)$  is the contact point and  $[\cdot]_\times$  denotes the skew matrix. In the proposition below, we provide a rationale of the condition specified (2):

*Proposition 1:* The condition (2) is equivalent to the Karush-Kuhn-Tucker conditions derived from the principle of maximal dissipation [25]:

$$\begin{aligned} 0 &\leq \|v_t\| \perp \mu\lambda_n - \|\lambda_t\| \geq 0 \\ \|v_t\|\lambda_t + \mu\lambda_n v_t &= 0. \end{aligned} \quad (3)$$

while we slightly abuse notation here by denoting  $\lambda_n$  as  $n(q)^T \lambda$ .

*Proof:* Consider the case where  $\mu\lambda_n - \|\lambda_t\| > 0$ , which corresponds to the stick condition in (3), and  $\|v_t\| = 0$  holds. In (2), this case implies  $\beta = 0$  and consequently  $\|v_t\| = 0$ , establishing equivalence. In the case of  $\mu\lambda_n - \|\lambda_t\| = 0$ , which aligns with the slip condition in (3), the condition  $\|v_t\|\lambda_t + \mu\lambda_n v_t = 0$  holds. In (2), this situation indicates that  $\mu\lambda_n + \lambda_t$  resides on the boundary of  $\mathcal{C}_{n(q)}$ , and accordingly,  $\beta n(q) + v_t$  must also lie on the boundary, where  $\beta = \|v_t\|$ . Consequently,  $\lambda_t^T v_t + \|\lambda_t\|\|v_t\| = 0$ , thus confirming their equivalence. ■

Despite the equivalence of (2) and (3), we opt to utilize the proposed from (2) in our framework to avoid explicit use of non-differentiable 2-norm  $\|\cdot\|$  terms with a combination of conic constraints.

#### C. Quasi-Dynamics

To make planning results dynamically feasible, employing dynamics factor is also essential. In general, the relation can be formalized as follows:

$$A(q)\delta q - b(q) - J_u(q)^T u - J_c(q)^T \lambda = 0 \quad (4)$$

where  $A(q) \in \mathbb{R}^{d \times d}$ ,  $b(q) \in \mathbb{R}^d$  are the dynamics matrix/vector typically related to inertia and control gain,  $J_u(q)$  is the input mapping matrix. As we focus on manipulation problem in this work, we primarily adopt the quasi-static or quasi-dynamic assumption [23] in (4). While this may overlook certain dynamic effects, such as Coriolis forces, it generally leads to more well-defined inference process.

#### D. Differentiable Contact Features

The aforementioned equations on contact constraints are composed of contact features such as gap  $g(q)$ , point  $p(q)$ , and normal  $n(q)$ . As our objective is to develop an efficient *gradient-based* inference scheme, it is essential to ensure the differentiability of factors. To achieve this, we adopt a differentiable support function [26] as the geometry representation in our framework. This approach theoretically

<sup>1</sup>In this work, we express  $\lambda$  and  $J_c$  with respect to global coordinates. This obviates the need to explicitly specify the contact tangential coordinates, making it easier in computation and differentiation.

guarantee to provide differentiable contact features (DCF) and can represent arbitrary compositions of convex shapes. It is important to note that such differentiability of contact features between geometries is under-explored and has not been utilized enough even in recent literature [23], [27], where shapes are often simplified to spheres or planes.

#### E. Composition of Factors

In manipulation planning, the graphical model should integrate the various constraint factors outlined earlier in a compositional manner. Often, we need to utilize a specific set of factors tailored to the task at hand, with each factor corresponding to a subset of variables. Below we present some examples of characterized sets of constraints for specific situations: For static contact manipulation tasks like grasping or placement, the necessary factors may include:

$$g(q) \geq 0 \quad g(q)\lambda = 0 \quad \mu\lambda_n + \lambda_t \in \mathcal{C}_{n(q)} \quad (5)$$

with static equilibrium from (4). These functional constraints (5) indicate the set of variables that satisfy contact complementarity and maintain stability with respect to rational contact forces. Alternatively, if the objective is to maintain contact without slipping, as in pivoting scenarios, the following set of factors can be employed:

$$g(q) = 0 \quad v_t = 0 \quad \mu\lambda_n + \lambda_t \in \mathcal{C}_{n(q)} \quad (6)$$

with quasi-dynamics equation from (4). The above (5) and (6) can be interpreted as distinct sets of contact factors. Similarly, various types of interactions can be modeled using a compositional arrangement of factors while the joint distribution on the variable nodes can be expressed as follows:

$$p(X) = \prod_{i=1}^M p_i(X_i) \text{ where } p_i(X_i) \propto \exp(-f_i) \quad (7)$$

where  $M$  is the number of factor node,  $X_i \subseteq X$  is the set of variable adjacent to  $i$ -th factor,  $f_i$  is the energy function that encode the constraints. The form of the energy function depends on the type of constraints such as equality ( $r_i = 0$ ), inequality ( $r_i \geq 0$ ), and cone constraints ( $r_i \in \mathcal{C}$ ). In this work, we assume the following forms of energy function  $f_i$  for each type of constraint:

$$\frac{1}{2}\|r_i\|^2, \quad \frac{1}{2}\|\min(r_i, 0)\|^2, \quad \frac{1}{2}\|d_{\mathcal{C}}(r_i)\|^2 \quad (8)$$

which are for equality, inequality, and cone, respectively. Here,  $d_{\mathcal{C}}$  is the L2 distance to the cone  $\mathcal{C}$ . As mentioned earlier, each  $r_i$  is readily differentiable. We also remark that our CFG formulation does not simply embed a forward simulator; instead, it provides the flexibility to model various types of interactions within physical reasoning. Fig. 1 illustrates an example of CFG modeling.

### IV. INFERENCE IN CONTACT FACTOR GRAPHS

Given the modeling described in Sec. III, we need to perform inference on the joint distribution (7). Approximating such complex multimodal distributions presents a significant challenge in machine learning. In this section, we propose

an efficient Bayesian inference algorithm that exploits on the structure of the inner factorized likelihood function.

#### A. MAP Inference on the Conditional Distribution

Joint distribution on  $X$  is typically high-dimensional, especially when the number of contact candidate increases. To alleviate this issue, let us first consider the distribution on  $u, \lambda$  conditioned by  $q$ . Then we can perform MAP inference on the conditional distribution  $p(u, \lambda | q)$  through following optimization:

$$(\lambda^*(q), u^*(q)) = \arg \min_{\lambda, u} \sum_{i=1}^M f_i(\lambda, u, q). \quad (9)$$

Our key insight is that for the factors (1), (2) and (4), and also their task-specific versions (e.g., (5)),  $u, \lambda$  exhibit linearity in  $r_i$  given  $q$ . Then following lemma gives the convexity of the energy function form (8):

*Lemma 1:* The energy functions defined in (8) are convex and have Lipschitz continuous gradient with respect to  $r_i$ .

*Proof:* For inequality and equality types, the properties are trivial. For cone types, we refer to Lemma 2 of [28] for proof. ■

Consequently, the optimization problem described in (9) is convex, enabling us to perform MAP inference on the conditional distribution  $p(u, \lambda | q)$  in an efficient manner, on a global scale. As shown in Lemma 1, the gradient of this function possesses Lipschitz continuity, from which we can derive the Hessian matrix. For the Newton step, we can employ the semi-analytic primal (SAP) solver presented in [28], which utilizes exact linesearch and is proven to achieve superlinear convergence. Although this algorithm was originally developed for physics simulations, its theoretical properties are all preserved for (8), allows it to be seamlessly adapted to our process. We observe that this certifiable convex optimization provides significant convergence advantages in the inference process, as shown in Sec. V. Additionally, our scheme can dramatically reduce the number of expensive contact feature computation function calls, which are usually numerical processes themselves.

#### B. Conditional Independence and Factorization

Aforementioned Newton steps for solving (9) directly requires the factorization of the Hessian matrix. Here, the structure of the matrix corresponds to the configuration of  $p(u, \lambda | q)$ , which we can exploit during the factorization process. For instance, independence arises between  $u$  and  $\lambda$  acting at different time intervals due to the Markov property of physics (i.e.,  $q$  at any point in time only affects  $q, u, \lambda$  at the immediately preceding point in time). Generally, the dependency between variables in conditional distribution is determined by the quasi-dynamics factors (4), particularly through the structure of the Jacobians. In our framework, we effectively leverage this sparsity pattern during the factorization of the Hessian, which can be equivalently interpreted as an elimination process on the factor graph [6].

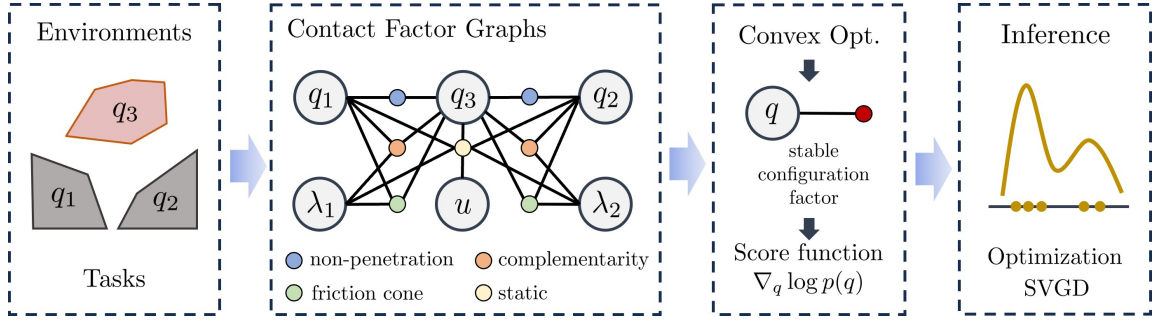


Fig. 1: Overview example of the proposed framework. Given the information on environment, contact factor graphs is composed using dynamics and contact factors relevant to the tasks. During the inference phase, convex optimization is employed to compress the distribution with respect to  $q$ . Score function is computed and applied within optimization or SVGD for inference.

### C. Inference on $q$

Preceding MAP inference on the conditional distribution identifies the most plausible estimates  $\lambda$  and  $u$  for a given  $q$ . Building on this, we redefine the inference problem over the distribution of  $q$  as follows:

$$p(q) \propto \prod_{i=1}^M \exp(-f_i(\lambda^*(q), u^*(q), q)). \quad (10)$$

This distribution (10) is not exactly a marginalization of the original distribution; rather, it represents the conditionally optimal distribution of  $q$  from the MAP estimates  $\lambda^*(q)$  and  $u^*(q)$ . Yet, this conditioned distribution remains multimodal and is intractable to compute directly. We should not merely assessing  $q$  in isolation but considering how  $q$  behaves when optimized estimates  $\lambda^*(q)$  and  $u^*(q)$  are factored into the evaluation.

1) *Score Function via Envelope Theorem:* To perform inference on the distribution given in (10), we can adopt various optimization, MCMC or variational inference algorithms. Here, the score function can play an essential role in the process from high-dimensional or constrained distributions. Within our framework, given that the MAP inference (e.g., (9)) represents a parameterized convex optimization with respect to  $q$ , we propose to derive the score function for (10) by employing the envelope theorem [29]. Accordingly, we get

$$\nabla_q \log p(q) = \sum_{i=1}^M \frac{\partial f_i(\lambda^*, u^*, q)}{\partial q}. \quad (11)$$

Note that the right hand side is composed of *partial* derivative of  $f_i$  with respect to  $q$ , therefore does not require any derivatives of  $\lambda^*$  and  $u^*$  with respect to  $q$ . It is worth noting that computing this derivative using implicit differentiation typically necessitates matrix inversion of problem-size matrices or a tailored design of the solver [30], which can degrade the efficiency of the entire framework. In the supplementary material, we summarize the results of specific derivation of partial derivatives for diverse contact factors.

2) *MAP Inference:* Following (10), conducting MAP inference in a CFG essentially entails solving the following

---

#### Algorithm 1: MAP Inference in CFG

---

```

1 Input:  $\mathcal{G}$ 
2 Initialize  $l = 0, q^0$ , inverse Hessian  $B^0$ 
3 while not converged do
4   Compute DCF for  $q^l$ 
5   Solve (9) under  $q^l$  using SAP solver
6   Compute score function  $\nabla \log p(q^l)$  (11)
7   Compute  $\Delta q^l = -B^l \nabla \log p(q^l)$ 
8   Compute  $\alpha \leftarrow \text{Line-Search}(q^l, \Delta q^l)$ 
9   Update  $q^{l+1} \leftarrow q^l + \alpha \Delta q^l$ 
10  Update inverse Hessian  $B^{l+1}$  via BFGS
11   $l \leftarrow l + 1$ 
12 end
13 Output:  $X$ 

```

---

optimization problem:

$$\min_q \sum_{i=1}^M f_i(\lambda^*(q), u^*(q), q) \quad (12)$$

Given that (11) provides the gradient efficiently, using specific gradient-descent methods can facilitate the solution of (12). Among the various algorithms available, we have chosen to use the Broyden–Fletcher–Goldfarb–Shanno (BFGS) algorithm [31]. This quasi-Newton method is particularly effective because it reliably converges by estimating the inverse Hessian matrix at each step, using only the gradient information provided. Alg. 1 summarizes the complete MAP inference process in CFG.

3) *Variational Inference:* The MAP inference in Alg. 1 typically yields only a single solution and thus cannot capture the multi-modality of the probabilistic distribution (10). Additionally, the solution obtained may be a local optimum and may not be sufficiently reasonable. Many problems involving contacts in robotics have multi-modal solutions, making it necessary to capture this variability. In this context, variational inference on CFG can provide an overall approximation of the distribution, which can be formulated as

$$\hat{p}^*(q) = \arg \min_{\hat{p}} D_{\text{KL}}(\hat{p}(q) \parallel p(q)) \quad (13)$$

---

**Algorithm 2:** Variational Inference in CFG

---

```
1 Input:  $\mathcal{G}$ , kernel  $\mathcal{K}$ 
2 Initialize  $l = 0$ , learning rate  $\epsilon$ , particle  $\{q^j\}_{j=1}^S$ 
3 while not converged do
4   for  $j = 1 : S$  do
5     Compute DCF for  $q^{j,l}$ 
6     Solve (9) under  $q^{j,l}$  using SAP solver
7     Compute kernel function  $\mathcal{K}(q^{j,l}, q)$ 
8     Compute score function  $\nabla \log p(q^{j,l})$  (11)
9     Update particles (14)
10  end
11   $l \leftarrow l + 1$ 
12 end
13 Output:  $X$ 
```

---

where  $D_{\text{KL}}$  represents the Kullback-Leibler divergence.

To address (13), we utilize Stein Variational Gradient Descent (SVGD [32]). In SVGD, the distribution  $\hat{p}(q)$  is modeled non-parametrically using a set of particles. At each iteration, these particles are updated to minimize the divergence from the true distribution, utilizing techniques from reproducing kernel Hilbert space. The update process for each particle  $q$  is as follows:

$$q^j \leftarrow q^j + \epsilon \eta(q^j)$$
$$\eta(q) = \sum_{j=1}^S [\mathcal{K}(q^j, q) \nabla_{q^j} \log p(q^j) + \nabla_{q^j} \mathcal{K}(q^j, q)] \quad (14)$$

where  $S$  is the number of particles,  $\epsilon \in \mathbb{R}^+$  is a learning rate,  $\mathcal{K}$  is a positive definite kernel function. Essentially, the process (14) guides the particles to follow the score function, while a repulsive force prevents them collapse into a single mode. As (14) requires score function, this mechanism can be seamlessly integrated with the computation of (11).

Integrating with SVGD offers significant advantages, including the ability to perform operations in parallel across all particles and generally superior particle efficiency compared to MCMC-based sampling algorithms. The detailed variational inference process is presented in Alg. 2. Additionally, the overall framework is illustrated in Fig. 1.

## V. ILLUSTRATIVE EXAMPLES

In this section, we present various illustrative examples of manipulation planning using CFG. For the implementation, we primarily use MATLAB and C++, with `fminunc` and custom BFGS module. The planning results are validated through physics simulation [33].

### A. Stable Placement

In our first experimental scenario, we tackle the problem of inferring stable placement poses for a given object within a 2D environment. To address this problem, we apply the principles specified in (5) within the CFG with equilibrium constraints under various external inputs  $u$ . Note that we

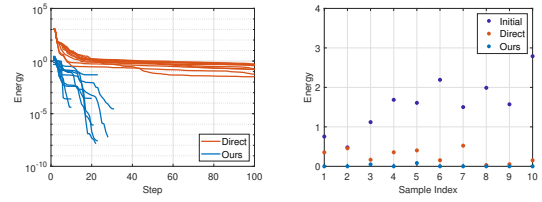


Fig. 2: Comparison our MAP inference algorithm and direct inference on the joint distribution. Left: Convergence over iterations. Right: Quality of the final results.

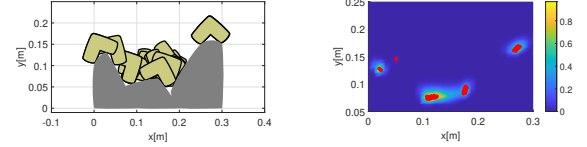


Fig. 3: Examples of variational inference results in CFG. Left: Stable object poses sampled from the approximated distribution. Right: Visualization of the SVGD results under a fixed rotation.

do not pre-specify contact pairs; instead, our CFG search navigates the space of making and breaking contacts.

We compare our inference scheme (Alg. 1) with the baseline approach that performs inference directly on the joint distribution (7) by optimization (we refer to this as `direct` hereafter). As shown in Fig. 2, our method exhibits significantly faster convergence. For 10 randomly initialized samples, all of them successfully converge: 2 to (undesirable) local minima and 8 to proper stable poses. In the `direct` method, most of the samples fail to converge properly, and only 1 of them succeed in finding a stable pose.

We also apply variational inference (Alg. 2) to approximate distribution of stable pose. For the kernel function, we use a common radial basis function. The results depicted in Fig. 3, demonstrate that CFG modeling combined with gradient-based SVGD successfully generates particles that approximate the distribution, enabling us to sample diverse and high-quality poses.

### B. Pivoting

Next, we consider the pivoting task, which involves rotating an object while maintaining contact with both the floor and the manipulator. In CFG, we model the problem that reasons on the contact points (where the manipulator will push the object during the task) and the appropriate contact forces that achieve quasi-dynamics with the stick state (6) at each time step.

Our results are depicted in Fig. 4. Given the clear multimodality of the possible pushing point, we test variational inference in CFG. As shown, the resulting particles from Alg. 2 successfully capture the distribution. We also compared these results with ensemble MCMC [34], a widely-used Bayesian inference method. Although the ensemble scheme offers advantages in exploration, the resulting samples fail to properly capture the distribution, possibly due to the complex contact landscape and non-reliance on gradient information.

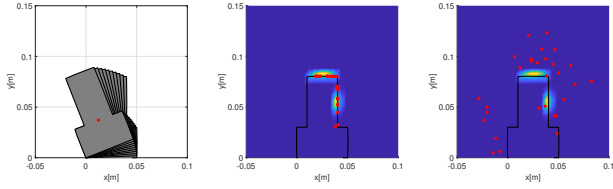


Fig. 4: Left: Motion of pivoting manipulation. Middle: Visualization of SVGD results within our framework. Right: Visualization of ensemble MCMC results.

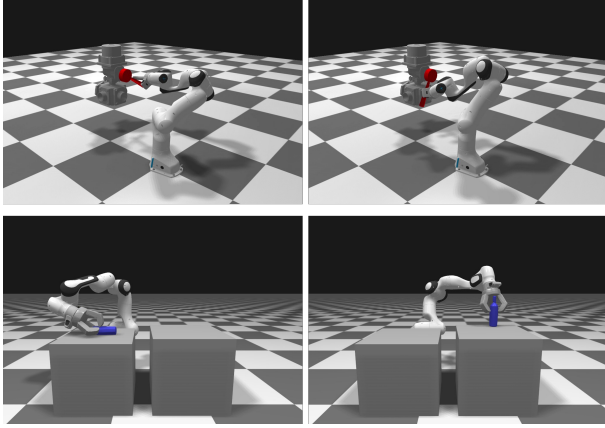


Fig. 5: Snapshots of planning results generated through inferences in CFG. Top: valve turning with slide maneuvers. Bottom: multifinger grasping and placing.

### C. Valve Turning with Slide

Subsequently, we test a scenario in which the robot uses its end effector to turn a lever on a valve. Here, we give the critical constraint that the robot must maintain contact with the lever and turn it while avoiding reaching its joint limit. Therefore, it should effectively exploit sliding on the contact surface. As illustrated in the compressed graph in Fig. 6, we model the planning problem as a sequence of touch and turn actions. The touch factor is represented by  $g(q) = 0$ , while the turn factor incorporates  $g(q) = 0$ , the desired lever angle, joint limits, and the friction law (2).

Then as in integrated task and motion planning [35], we can sequentially apply the CFG inference process in a factored manner to derive the complete solution. Initially, we sample the touching pose  $q_0$ , followed by the turning trajectory  $q_1, \dots, q_N$ . Across 10 different lever geometries, we successfully generate planning results in an average of 74 ms (see Fig. 5 for a snapshot) using Alg. 1, whereas the *Direct* method failed to produce any successful samples.

### D. Multifinger Grasp and Place

Finally, we tackle the multifinger grasp and place problem, which involves inferring proper grasping poses for a gripper and a given object, and then place it in a stable pose. Similar to the valve turning example, we can represent and solve the problem sequentially: first, find the gripper pose  $q_e$  with manipulator pose  $q_1$ , then determine the placement pose  $q_2$  as depicted in Fig. 6. We include kinematic feasibility, collision

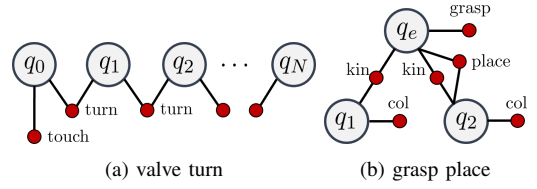


Fig. 6: Compressed form of CFG with respect to  $q$  (as in Fig. 1) for manipulation planning problems. For better efficiency, inference can be performed sequentially on each subgraph.

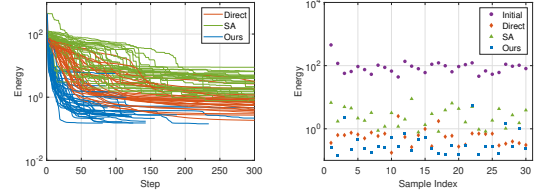


Fig. 7: Comparison our MAP inference algorithm and direct inference on the joint distribution/simulation annealing. Left: Convergence over iterations. Right: Quality of the final results.

avoidance, and the set of contact factors (5) to naturally handle complementarity and stability, while eliminating the need to pre-specify the contact interface.

By aforementioned strategy, we successfully generate planning results in an average of 384 ms (see Fig. 5 for a snapshot). To validate the sample generation performance, we compare  $(q_e, q_1)$  sampling using Alg. 1, with the *direct* method and also simulated annealing (SA), which performs global optimization through sampling without relying on gradient information. The results, depicted in Fig. 7, demonstrate that our methods achieve significantly faster convergence and yield higher-quality solutions. As other methods exhibit very low sampling success rates, the overall planning time slowed down by more than  $10\times$ .

## VI. DISCUSSION AND CONCLUSION

In this paper, we introduce the Contact Factor Graph (CFG) framework, which serves as a versatile tool for addressing a wide geometric-physical reasoning problems arise in manipulation planning. CFG facilitates the reasoning by modeling a differentiable, factorized probabilistic distribution aligned with contact mechanics and dynamics. We further present an inference algorithm for CFG, employing parameterized convex optimization techniques that leverage efficient gradient computation through the envelope theorem.

One limitation of the current framework is our assumption that the skeleton of manipulation planning is predefined. As partially demonstrated in Sec. V, we believe that various reasoning in CFG can be extended by sequential sampling on subgraphs, therefore can be combined to a range of works on task-level planning [35]–[37] or subgoal sampling [23], to significantly improve the tractability of the diverse array of complex manipulation. Also, our efficient and scalable solution sample generation scheme can also be used for data generation to build learning-based models, particularly for learning implicit distributions for diffusion processes [10].



## REFERENCES

- [1] M. Botvinick and M. Toussaint. Planning as inference. *Trends in cognitive sciences*, 16(10):485–488, 2012.
- [2] S. Levine. Reinforcement learning and control as probabilistic inference: Tutorial and review. *arXiv preprint arXiv:1805.00909*, 2018.
- [3] M. Mukadam, J. Dong, X. Yan, F. Dellaert, and B. Boots. Continuous-time gaussian process motion planning via probabilistic inference. *The International Journal of Robotics Research*, 37(11):1319–1340, 2018.
- [4] J. Urain, A. T. Le, A. Lambert, G. Chalvatzaki, B. Boots, and J. Peters. Learning implicit priors for motion optimization. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 7672–7679, 2022.
- [5] H. Yu and Y. Chen. A gaussian variational inference approach to motion planning. *IEEE Robotics and Automation Letters*, 8(5):2518–2525, 2023.
- [6] F. Dellaert. Factor graphs: Exploiting structure in robotics. *Annual Review of Control, Robotics, and Autonomous Systems*, 4:141–166, 2021.
- [7] J. Urain, A. Li, P. Liu, C. D’Eramo, and J. Peters. Composable energy policies for reactive motion generation and reinforcement learning. *The International Journal of Robotics Research*, 42(10):827–858, 2023.
- [8] A. Lambert, A. Fishman, D. Fox, B. Boots, and F. Ramos. Stein variational model predictive control. In *Conference on Robot Learning*, pages 1278–1297, 2020.
- [9] T. Power and D. Berenson. Constrained stein variational trajectory optimization. *arXiv preprint arXiv:2308.12110*, 2023.
- [10] Z. Yang, J. Mao, Y. Du, J. Wu, J. B. Tenenbaum, T. Lozano-Pérez, and L. P. Kaelbling. Compositional diffusion-based continuous constraint solvers. In *Conference on Robot Learning*, pages 1563–1574, 2023.
- [11] J. Lee, M. Lee, and D. Lee. Large-dimensional multibody dynamics simulation using contact nodalization and diagonalization. *IEEE Transactions on Robotics*, 39(2):1419–1438, 2022.
- [12] J. C. Trinkle and J. J. Hunter. A framework for planning dexterous manipulation. In *IEEE International Conference on Robotics and Automation*, pages 1245–1251, 1991.
- [13] R. Deits, T. Koolen, and R. Tedrake. Lvis: Learning from value function intervals for contact-aware robot controllers. In *IEEE International Conference on Robotics and Automation*, pages 7762–7768, 2019.
- [14] I. Mordatch, Z. Popović, and E. Todorov. Contact-invariant optimization for hand manipulation. In *Proceedings of the ACM SIGGRAPH/Eurographics symposium on computer animation*, pages 137–144, 2012.
- [15] M. Posa, C. Cantu, and R. Tedrake. A direct method for trajectory optimization of rigid bodies through contact. *The International Journal of Robotics Research*, 33(1):69–81, 2014.
- [16] T. A. Howell, K. Tracy, S. Le Cleac’h, and Z. Manchester. Calipso: A differentiable solver for trajectory optimization with conic and complementarity constraints. In *The International Symposium of Robotics Research*, pages 504–521, 2022.
- [17] M. Toussaint, J. Ha, and D. Driess. Describing physics for physical reasoning: Force-based sequential manipulation planning. *IEEE Robotics and Automation Letters*, 5(4):6209–6216, 2020.
- [18] M. Geilinger, D. Hahn, J. Zehnder, M. Bäcker, B. Thomaszewski, and S. Coros. Add: Analytically differentiable dynamics for multibody systems with frictional contact. *ACM Transactions on Graphics (TOG)*, 39(6):1–15, 2020.
- [19] K. Werling, D. Omens, J. Lee, I. Exarchos, and C. K. Liu. Fast and feature-complete differentiable physics for articulated rigid bodies with contact. In *Robotics: Science and Systems*, 2021.
- [20] T. A. Howell, S. Le Cleac’h, J. Z. Kolter, M. Schwager, and Z. Manchester. Dojo: A differentiable simulator for robotics. *arXiv preprint arXiv:2203.00806*, 9.
- [21] M. Lee, J. Lee, and D. Lee. Differentiable dynamics simulation using invariant contact mapping and damped contact force. In *IEEE International Conference on Robotics and Automation*, pages 11683–11689, 2023.
- [22] D. Turpin, L. Wang, E. Heiden, Y. Chen, M. Macklin, S. Tsogkas, S. Dickinson, and A. Garg. Grasp’d: Differentiable contact-rich grasp synthesis for multi-fingered hands. In *European Conference on Computer Vision*, pages 201–221, 2022.
- [23] T. Pang, H. Suh, L. Yang, and R. Tedrake. Global planning for contact-rich manipulation via local smoothing of quasi-dynamic contact models. *IEEE Transactions on Robotics*, 2023.
- [24] R. Antonova, J. Yang, K. M. Jatavallabhula, and J. Bohg. Rethinking optimization with differentiable simulation from a global perspective. In *Conference on Robot Learning*, pages 276–286, 2022.
- [25] M. Macklin, K. Erleben, M. Müller, N. Chentanez, S. Jeschke, and V. Makovychuk. Non-smooth newton methods for deformable multibody dynamics. *ACM Transactions on Graphics*, 38(5):1–20, 2019.
- [26] J. Lee, M. Lee, and D. Lee. Uncertain pose estimation during contact tasks using differentiable contact features. In *Robotics: Science and Systems*, 2023.
- [27] S. Le Cleac’h, T. A. Howell, S. Yang, C.-Y. Lee, J. Zhang, A. Bishop, M. Schwager, and Z. Manchester. Fast contact-implicit model predictive control. *IEEE Transactions on Robotics*, 2024.
- [28] A. M. Castro, F. N. Permenter, and X. Han. An unconstrained convex formulation of compliant contact. *IEEE Transactions on Robotics*, 39(2):1301–1320, 2022.
- [29] M. Carter. *Foundations of mathematical economics*. MIT press, 2001.
- [30] A. Agrawal, S. Barratt, S. Boyd, E. Busseti, and W. M. Moursi. Differentiating through a cone program. *arXiv preprint arXiv:1904.09043*, 2019.
- [31] R. Fletcher. *Practical methods of optimization*. John Wiley & Sons, 2000.
- [32] Q. Liu and D. Wang. Stein variational gradient descent: A general purpose bayesian inference algorithm. *Advances in neural information processing systems*, 29, 2016.
- [33] J. Lee, M. Lee, and D. Lee. Modular and parallelizable multibody physics simulation via subsystem-based admm. In *IEEE International Conference on Robotics and Automation*, pages 10132–10138, 2023.
- [34] D. Foreman-Mackey, D. W. Hogg, D. Lang, and J. Goodman. emcee: the mcmc hammer. *Publications of the Astronomical Society of the Pacific*, 125(925):306, 2013.
- [35] C. R. Garrett, R. Chitnis, R. Holladay, B. Kim, T. Silver, L. P. Kaelbling, and T. Lozano-Pérez. Integrated task and motion planning. *Annual review of control, robotics, and autonomous systems*, 4:265–293, 2021.
- [36] M. Toussaint. Logic-geometric programming: An optimization-based approach to combined task and motion planning. In *IJCAI*, pages 1930–1936, 2015.
- [37] T. Migimatsu and J. Bohg. Object-centric task and motion planning in dynamic environments. *IEEE Robotics and Automation Letters*, 5(2):844–851, 2020.