2024 IEEE International Conference on Robotics and Automation (ICRA 2024)
May 13-17, 2024. Yokohama, Japan

# TPGP: Temporal-Parametric Optimization with Deep Grasp Prior for Dexterous Motion Planning

Haoming Li[1], Qi Ye[1†], Yuchi Huo[2], Qingtao Liu[1], Shijian Jiang[1], Tao Zhou[1], Xiang Li[3], Yang Zhou[3], Jiming Chen[1]

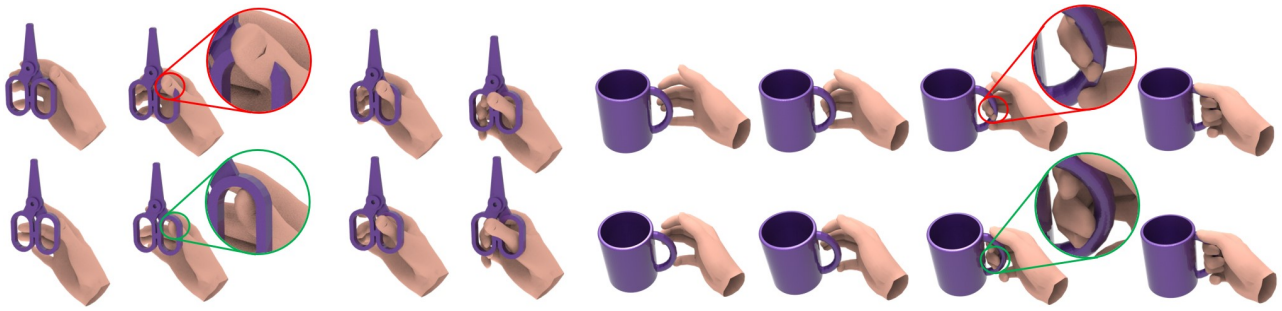

Fig. 1: Two examples of erroneous (top) and optimized (bottom) grasping motion.

***Abstract*— Grasping motion planning aims to find a feasible grasping trajectory in the configuration space given an input target grasp. While optimizing grasp motion with two or three-fingered grippers has been well studied, the study on natural grasp motion planning with a dexterous hand remains a very challenging problem due to the high dimensional working space. In this work, we propose a novel temporal-parametric grasp prior (TPGP) optimization method to simplify the difficulty of grasping trajectory optimization for the dexterous hand while maintaining smooth and natural properties of the grasping motion. Specifically, we formulate the discrete trajectory parameters into a temporal-based parameterization, where the prior constraint provided by a hand poser network, is introduced to ensure that hand pose is natural and reasonable throughout the trajectory. Finally, we present a joint target optimization strategy to enhance the target pose for more feasible trajectories. Extensive validations on two public datasets show that our method outperforms state-of-the-art methods regarding grasp motion on various metrics.**

## I. Introduction

Grasp motion planning aims to generate a collision-free path towards grasping an object as illustrated in Figure 1. The problem has been studied for decades in robotics and most existing grasping motion planning problems mainly focus on robotic arms with grippers. In recent years, humanoid robots have gained increasing interest both in the academic and industrial community, and grasping motion planning for dexterous hands with high degrees of freedom (DOFs) has become an important research problem.

Over the past decade, researchers have adopted various types of methods for the grasp motion planning task, including sampled-based approaches [1]–[5] and trajectory optimization [6]–[8]. Sampling-based planning involves repeatedly sampling configurations from a space to find the optimal path. It could incur very high computational costs for the high DOFs manipulator. On the other hand, optimization-based methods find a collision-free path to the goal by minimizing an objective function according to some criterion, which is relatively more efficient than sampling-based methods. Previous trajectory optimization methods [6], [7], [9] have showcased their effectiveness in grasp motion planning tasks employing two or three-fingered robotic grippers. However, when applied to high DOFs dexterous hands, the complexity of interaction poses increases significantly. These methods struggle with optimizing high-dimensional trajectory search spaces, often getting trapped into local minima. In addition to the high DOFs, the planning for dexterous hand motion involves coordination between fingers. Both sampling-based and optimization-based methods face the challenge of producing natural hand pose sequences to avoid uncanny interactions.

In this work, we propose a novel temporal-parametric optimization method incorporating hand priors for grasp motion planning to tackle these challenges. Instead of optimizing pose parameters at each motion step, we model the whole motion trajectory with a function parameterized by variables controlling the joint moving tempo. The temporal parametric method comes from the observation that despite the high degree of freedom in dexterous hands, during the process of grasping objects, all fingers move inwards to target poses, and different coordination of moving tempo forms different grasp motions. For the second challenge,

[1]College of Control Science and Engineering, Zhejiang University, Hangzhou, 310027, China.
[2]State Key Lab of CAD&CG, Zhejiang University, and Zhejiang Lab.
[3]OPPO US Research Center.
[†]Qi Ye (Corresponding Author, qi.ye@zju.edu.cn) is with the College of Control Science and Engineering, the State Key Laboratory of Industrial Control Technology, Zhejiang University, and the Key Key Lab of CS&AUS of Zhejiang Province.

This work was supported in part by the National Natural Science Foundation of China (Grant Number: 62088101,62103372,62233013), and OPPO Research Fund.

we learn a grasp prior by a network named the hand poser (HPoser) to make the joint movement form a natural grasp motion, which encodes the hand poses into a latent space, and we optimize in the latent space. With the parametric method and the grasp prior, we reduce the parameters to be optimized from $\#steps \times \#pose$ to $(\#latentcode + \#translation + \#globalorientation) \times 2$, i.e. hundreds to dozens in our setting. Finally, as target poses generated from networks may exhibit small errors, we further propose a joint target optimization (JTO) strategy to refine the final grasping pose within the trajectory optimization, resulting in a more reasonable grasping motion.

In summary, our paper has the following key contributions:

- we propose an effective trajectory optimization algorithm for grasp motion planning, which formulates the problem with functions parameterized temporal-based variables.
- we propose an HPoser network to learn a latent distribution for the hand interaction pose and integrate it with our temporal-based optimization to produce natural grasp motions.
- we propose a joint target optimization strategy, which continuously adjusts the target grasp pose during the trajectory optimization process to obtain more feasible grasping trajectories.

## II. Related Works

### A. Sampling-Based Methods

Sampling-based planning involves repeatedly sampling configurations from a space, expanding a search graph to cover this space, and then finding a collision-free path from a start to a goal configuration [1]–[4], [10]. For example, probabilistic road-maps [1] construct a dense graph from random samples in obstacle-free areas of the robot's configuration space. Cross-Entropy Method (CEM) [3] aims to find the optimal solutions for combinatorial and continuous nonconvex optimization problems within convex bounded domains. Rapidly exploring Random Trees methods [2], [4] find trajectories by incrementally building space-filling trees through directed sampling. Although effective, these methods are difficult to use in some applications due to the computational challenges, especially for the high-dimension working space.

### B. Trajectory Optimization

Unlike sampling-based strategies, trajectory optimization methods begin with an initial, potentially unrealistic trajectory and refine it by minimizing a cost function. For example, Covariant Hamiltonian Optimization (CHOMP) [6], [8] optimizes a cost function using covariant gradient descent. TrajOpt [11] solves a sequential quadratic program and performs convex continuous-time collision checking. Some works attempt to integrate deep learning with optimization algorithms to address trajectory optimization problems. In GPMP2 [7], the problem is defined as an inference task on a factor graph, where the objective is to identify the maximum posterior trajectory through the resolution of a nonlinear least squares problem. These methods effectively optimize grasping motions involving two or three-fingered grippers but encounter challenges when dealing with dexterous hands. In contrast, we introduce a temporal-parametric-based approach for optimizing high-DOFs grasping motion trajectories.

### C. Data-Driven-Based Methods

As the field of deep learning continues to evolve, researchers are actively exploring data-driven approaches to generate plausible grasping trajectories [12] or provide rich prior knowledge for the optimization of grasping trajectories [13]. TOCH [13] design a neural network to provide a corrected hand-object distance field for optimizing the grasping pose of the trajectory. GOAL [12] uses the generative model to learn the distribution of the grasping sequence and generate various grasp motions given an object as conditional input. The generated trajectories by these methods are relatively rough, making it challenging to achieve precise control over joints. However, we can employ the dataset to learn the hand pose distribution and utilize it as priors to constrain abnormal hand poses.

## III. Methods

Given an object represented by Signed Distance Field (SDF) [14], [15], our method constructs the trajectory configuration $(P^i)_{0 \le i \le T}$ with $T$ frames that guide the hand to interact with the object without collision. We assume the target pose $P^T$ can be generated by existing grasp generators [16]–[19]. We use the Contac2Grasp [19] to generate target poses in our experiments. We set the start pose $P^0$ as a zero-initialized finger pose [20], located $10cm$ away in the direction from the object center to the target palm center. The main focus of our work is to conduct interpolation between the target pose and the start one in an optimizable manner.

Specifically, we adopt MANO parameters [20] to represent the hand. The hand pose parameter $P = (\theta_t, \theta_g, \theta_p)$ consists of translation $\theta_t \in R^3$, global orientation $\theta_g \in R^3$, and finger pose $\theta_p \in R^{45}$. Given $P$ and the shape parameters $\beta \in R^{10}$, the MANO model $\mathcal{M}$ parameterizes the hand mesh $M = (V, F)$ ($V \in R^{778 \times 3}, F \in R^{1538}$ denotes the mesh vertices and faces), i.e. $M = \mathcal{M}(P, \beta)$. In this work, we use the mean shape and use $M = \mathcal{M}(P)$ for brevity.

Figure 2 shows the pipeline of our method. Given the target pose $P^T$ generated by Contac2Grasp [19] and an accordingly initialized start pose $P^0$, we first embed the $\theta_p$ of the start pose $P^0$ and the target pose $P^T$ into latent codes $Z^0$ and $Z^T$ respectively. Then, the temporal-parametric grasp prior optimization module receives the combination of the latent codes $Z$, translations $\theta_t$, and global orientations $\theta_g$ of the start and target pose, producing an optimized parameter sequence $(\hat{\theta}_t^i, \hat{\theta}_g^i, \hat{Z}^i)_{0 \le i \le T}$. Finally, each optimized latent code $\hat{Z}^i$ is decoded by the HPoser decoder, and the hand mesh $M^i$ at each timestep is reconstructed by the MANO model given the hand parameter $\hat{P}^i = (\hat{\theta}_t^i, \hat{\theta}_g^i, \hat{\theta}_p^i)$.

### A. HPoser Network

Due to the high DOFs in the human hand, directly optimizing the pose parameter $P$ without constraints leads

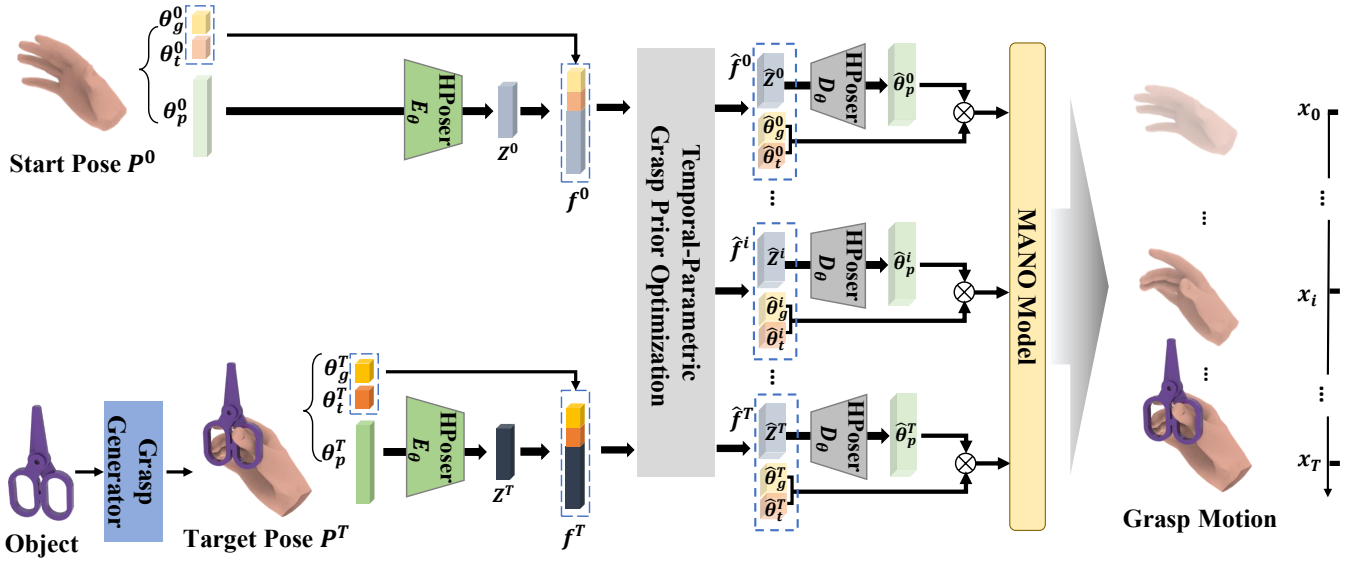


Fig. 2: The framework of our method. It mainly consists of two modules: an HPoser network and a temporal-parametric grasp prior optimization module. Given an object input, we first obtain the target pose $P^T$ by an existing grasp generator and accordingly initialize the start pose $P^0$. The HPoser encoder then embeds the mano parameters $\theta_p^0$ and $\theta_p^T$ into latent code $Z^0$ and $Z^T$. TPLP module takes $Z^0, Z^T$ and extra parameters $\theta_t$ and $\theta_g$ as input, formulating the optimized grasp motion sequence.

to high time costs and may generate unreasonable poses. To address these drawbacks, we adopt a pre-trained model, named HPoser, to embed $\theta_p$ into a latent code $Z$ with lower dimensions.

Similar to the VPoser [21], the HPoser adopts a variational autoencoder [22] architecture, consisting of an encoder $E_\theta$ and a decoder $D_\theta$. A pre-trained HPoser network can acquire grasp priors that encompass the correlation between fingers, leading to natural grasping motion formations. Besides, as HPoser is pre-trained with normal data, it is capable of regularizing illegal inputs into valid latent codes. We pre-train the model on the sequence of the Grab dataset [16] and fix the weight during grasp motion planning. The training loss settings for HPoser are consistent with those of VPoser [21].

### *B. Temporal-Parametric Grasp Prior Optimization*

Given the start and target pose parameters $f^0 = (\theta_t^0, \theta_g^0, Z^0)$ and $f^T = (\theta_t^T, \theta_g^T, Z^T)$, the temporal-parametric grasp prior optimization module interpolates intermediate pose trajectories in an optimizable manner. Specifically, it constructs a mapping function $\mathcal{F}(x_i, \alpha, \lambda)$ parameterized by $x_i$, $\alpha$, and $\lambda$ to modulate the pose parameter $(\theta_t, \theta_g, Z) \triangleq f$ in different time steps, where $x_i$ is the normalized time index, and $\alpha$, $\lambda$ are optimizable parameters. TPGP optimizes $\alpha$ and $\lambda$ between the start and target frames, finally formulating the optimized parameter sequence $(\hat{f}^i)_{0\le i\le T} = (\hat{\theta}_t^i, \hat{\theta}_g^i, \hat{Z}^i)_{0\le i\le T}$.

**Mapping Function.** The mapping function $\mathcal{F}$ is defined as a linear combination of three items:

$$\begin{aligned}\mathcal{F}(x_i, \alpha, \lambda) =& S(\alpha) x_i^{1/\sigma} + S(\lambda) x_i^{\sigma} \\ &+ (1 - S(\alpha) - S(\lambda)) x_i, \end{aligned} \tag{1}$$

where $S(\cdot)$ denotes the sigmoid function and $\sigma$ is a fixed scalar. We empirically set $\sigma$=10 and we receive similar performance when changing $\sigma$ from 3 to 15.

By involving optimizable parameters $\alpha$ and $\lambda$, the mapping function $\mathcal{F}$ supports cost-driven adjustments of accelerating (the first item) and decelerating (the second item) finger movements to avoid collisions. Note that when both $S(\alpha)$ and $S(\lambda)$ are equal to 0 or $\sigma$=1, the mapping function is simplified to $\mathcal{F}(x, \alpha, \lambda) = x$, which denotes the linear interpolation operation.

In practice, we adopt independent parameter groups $(\alpha_t \in R^3, \lambda_t \in R^3)$, $(\alpha_g \in R^3, \lambda_g \in R^3)$ and $(\alpha_z \in R^{Dim_z}, \lambda_z \in R^{Dim_z})$ to modulate $\theta_t$, $\theta_g$ and $Z$, respectively. $Dim_z$ denotes the dimensionality of $Z$. We specify the formulation of $\theta_t^i, \theta_g^i, Z^i$ at time step $i$ as:

$$\hat{\theta}_t^i = \Delta\theta_t \cdot \mathcal{F}(x_i, \alpha_t, \lambda_t) + \theta_t^0, \tag{2}$$

$$\hat{\theta}_g^i = \Delta\theta_g \cdot \mathcal{F}(x_i, \alpha_g, \lambda_g) + \theta_g^0, \tag{3}$$

$$\hat{Z}^i = \Delta Z \cdot \mathcal{F}(x_i, \alpha_z, \lambda_z) + Z^0, \tag{4}$$

where $\Delta\theta_t = \theta_t^T - \theta_t^0$ , $\Delta\theta_g = \theta_g^T - \theta_g^0$, and $\Delta Z = Z^T - Z^0$. For simplicity, we denote $\alpha = (\alpha_t, \alpha_g, \alpha_z)$ and $\lambda = (\lambda_t, \lambda_g, \lambda_z)$ blow.

**Trajectory Optimization.** We optimize $\alpha$ and $\lambda$ in a cost-driven manner. We denote $\alpha = (\alpha_t, \alpha_g, \alpha_z)$ the optimized hand parameter $P^i$ of the $i$-th frame as:

$$\hat{P}^i = cat(\hat{\theta}_t^i, \hat{\theta}_g^i, D_\theta(\hat{Z}^i)). \tag{5}$$

The cost function consists of three items: (1) a penetration cost $C_{ptr}^i = -\sum_{v\in\mathcal{M}(\hat{P}^i)} \min(sdf(v), 0)$ to punish penetrations between the hand and object of the $i$-th frame where $sdf(v)$ denotes signed distance value between hand

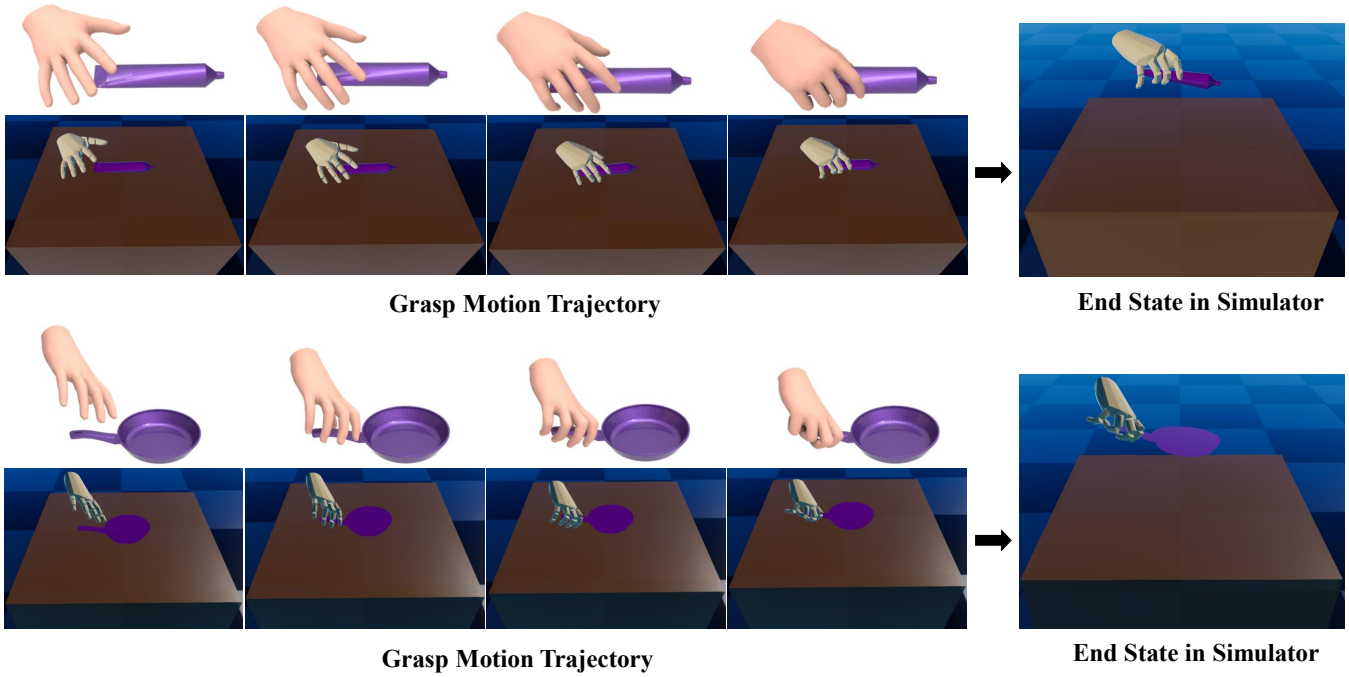


Fig. 3: The visualization of two different hand-object interaction sequences with the end state in the simulator.

vertice $v$ to the nearest object mesh triangle face; (2) a direction loss $C^i_{dir} = \max(\cos(n^i, n^T), \epsilon)$ to regularize the variation of hand global orientation within the trajectory where $n^i$=$\mathcal{N}(\mathcal{M}(\hat{P}^i))$ denotes the palm normal vector of the $i$-th frame, $\epsilon$ is a constant term and is set to $\frac{\sqrt{3}}{2}$; and (3) a regularization item $C_{reg} = \max(S(\alpha) + S(\lambda), 1)$. In summary, the overall cost function for the trajectory optimization $C_{traj}$ is the weighted sum of the above loss terms:

$$\begin{aligned} C_{traj}(\alpha, \lambda) =& \gamma_0 \frac{1}{T} \sum_{i=0}^{T} C^i_{ptr}(\alpha, \lambda) \\ &+ \gamma_1 \frac{1}{T-1} \sum_{i=0}^{T-1} C_{dir}(\alpha, \lambda) \\ &+ C_{reg}(\alpha, \lambda), \end{aligned} \tag{6}$$

where $\gamma_0 = 15$ and $\gamma_1 = 1$ are hyper-parameters.

Note that the trajectory optimization only affects the $(f^i)_{0 \le i < T}$, leaving $f^T$ being excluded. Considering that simply optimizing the target pose may result in unsuccessful grasping, we leave the target pose $f^T$ unchanged in this stage and we introduce the joint target optimization strategy later.

**Joint Target Optimization.** Acknowledging the significance of the target poses in grasp motion, we propose to jointly optimize the target pose $P^T$ by introducing additional cost functions. Formally, the cost function for target pose optimization is formulated as:

$$\begin{aligned} C_{grasp}(\hat{P}^T) =& \omega_0 C_{ptr}(\mathcal{M}(\hat{P}^T)) + \omega_1 C_{att}(\mathcal{M}(\hat{P}^T)) \\ &+ \omega_2 C_h(P^T, \hat{P}^T). \end{aligned} \tag{7}$$

$C_{ptr}$ penalizes the penetration between the target hand $\mathcal{M}(\hat{P}^T)$ and the object. $C_{att} = \sum_{v \in \bar{\mathcal{M}}(\hat{P}^T)} abs(sdf(v))$, where $\bar{\mathcal{M}}(\cdot)$ denotes the vertices on the thumb, index finger, and middle finger. $abs(\cdot)$ represents the absolute value operation. $C_h = \|P^T - \hat{P}^T\|$ regularizes the pose hypothesis $\hat{P}^T$ to stay close to $P^T$. We set $\omega_0$=100, $\omega_1$=20, and $\omega_2$=0.2.

The trajectory optimization and target pose optimization are performed in an iterative manner, where the target pose is optimized after each iteration of trajectory optimization. We find that involving the target pose optimization only in the first $N = 50$ steps yields satisfied performance.

## IV. Experiments

### A. Implementation Details

We set the frame number $T$=40. For TPGP optimization, we use the Adam optimizer and set the learning rate as 2e-2. If the decrease in cost value is less than the threshold, the optimization process will terminate prematurely, where the early-stop threshold is set to 1e-3. For the hand poser network, the model is trained using a batch size of 128 examples, and an Adam optimizer with a constant learning rate of 1e-4. The models ran 6 epochs. All the experiments were implemented in PyTorch, in which our method ran 200 steps in a single RTX 3090 GPU.

### B. Datasets

**Grab.** We evaluate our method on Grab [16], a MoCap dataset for whole-body grasping of objects. Grab contains interaction sequences with 51 objects from [23]. We use the same split and data filtering of the datasets with TOCH [13]. Since we are only interested in frames of grasp motion, we filter out frames where the hand center is more than 10 $cm$ away from the object and keep the frames before grasping the object only. To make a fair comparison, we add the same noise distribution in TOCH [13] to the ground truth as the

| Dataset | Methods | GSR (↑) | TPV (↓) | SE (↓) | Times (↓) |
|---|---|---|---|---|---|
| Grab | CEM [3] | 49.16 | 3.10 | 4.27 | 117.38 |
| | GOAL [12] | 29.16 | 7.97 | 2.22 | 29.27 |
| | TOCH [13] | 52.33 | 1.77 | 2.31 | 39.76 |
| | CHOMP [8] | 54.41 | 2.38 | 2.85 | **3.71** |
| | **Ours** | **66.95** | **1.45** | **1.79** | 4.28 |
| ContactPose | CEM [3] | 47.33 | 3.11 | 4.01 | 120.06 |
| | GOAL [12] | 28.19 | 8.66 | 2.01 | 31.41 |
| | TOCH [13] | 51.67 | 2.86 | 2.93 | 42.75 |
| | CHOMP [8] | 56.60 | 2.17 | 3.13 | **4.01** |
| | **Ours** | **71.25** | **1.63** | **1.80** | 4.32 |

TABLE I: Quantitative comparison with state-of-the-art on Grab and ContactPose test set.

target pose input (the last frame of the sequence). For each test object, the length of the sequence can be different. The average frame number is 34.

**ContactPose.** The ContactPose dataset [24] is a real dataset for studying hand-object interaction, which captures both ground-truth thermal contact maps and hand-object poses. The dataset contains 25 household objects and 2306 grasp contacts. We use the generated grasp pose as the input of our method instead of ground truth. Refers to Contact2Grasp [19], we use the test set objects with 120 generated grasp poses for the evaluation. As ContactPose doesn't provide sequence data, we acquire the translation of the start frame by transforming the grasp pose in the direction from the object to the palm center. Note that the finger pose of the start frame defaults to zeros.

### C. Evaluation Metrics

A good grasp motion trajectory should be smooth and exhibit reduced penetration. The target pose should be physically stable and in contact with the object without penetration. In this work, we adopt four metrics to evaluate the quality of the optimized grasp trajectory: (1) **Grasp Success Rate (GSR, $\%$)** The grasp success rate aims to evaluate the rate of grasp motion success. We evaluate all grasp trajectories in the Raisim simulator [25]. The positive sample means the object can be grasped over the table about $20cm$. The success rate is the percentage of those positive samples among all test samples. (2) **Trajectory Penetration Volume (TPV, $cm^3$)** The trajectory penetration is measured by the average volume between the objects and hand meshes over the sequence. Following [26], [27], the volume is measured by voxelizing the hand-object mesh with voxel size $0.5cm$. (3) **Smooth Error (SE, $cm/s^3$)** A high-quality grasp trajectory should be continuous and smooth. Following [28], the smooth error is defined as the third derivative of the trajectory. In this work, We only measure the mean error of the 21 finger joints over the sequence. (4) **Times ($s$)** Time cost metric is used to measure the efficiency of the optimization process. We measure the runtime of all the methods with a single RTX 3090 GPU device.

Considering the significance of the grasping pose (target pose) in the grasp trajectory, we introduce three additional metrics to evaluate the quality of the grasping pose. (1)

| Methods | SD(↓) | GPV(↓) | CR(↑) |
|---|---|---|---|
| RefineNet [16] | **1.34** | 6.29 | 99.86 |
| ContactOpt [29] | 2.03 | 3.67 | **100** |
| GraspTTA [26] | 1.52 | 3.71 | **100** |
| TOCH [13] | 1.97 | 3.46 | 99.97 |
| Ours | 1.49 | **1.39** | **100** |

TABLE II: Grasp pose (the last frame in trajectory) performance on ContactPose test set.

**Simulation Displacement (SD, $cm$)** The simulation displacement is adopted to measure the stability of the grasping pose, which is measured by a physics-based simulator following the same settings as [26]. The displacement is the Euclidean distance between the object centers before and after applying a grasp on the object. (2) **Grasp Penetration Volume (GPV, $cm^3$)** Similar to TPV, grasp penetration is also measured by the volume between the hand and object mesh. However, GPV only focuses on the grasping pose (last frame), rather than the entire sequence. (3) **Contact Rate (CR, $\%$)** A physically plausible grasping pose requires contact with object. We define a sample as positive if the hand-object contact exists, which means that there exists at least a point on the hand surface is on or inside the surface of the object. The contact rate is the percentage of those positive samples over all the test samples.

### D. Comparison with State-of-the-Art Methods

**Trajectories.** We compare our method with the state-of-the-art methods, including the data-driven-based methods (TOCH [13], GOAL [12]), sampled-based method (CEM [3]) and trajectory optimization method (CHOMP [8]). The quantitative results are shown in Table I. Lower TPV means fewer collisions will happen when the grasp trajectories are executed. Our method achieves the lowest TPV ($1.45cm^3$ and $1.63cm^3$) and the highest GSR ($66.95\%$ and $71.25\%$) in simulations, which indicates that optimizing the grasp trajectories with our method can make the trajectories more physically plausible. Another key point is the optimization time. Compared to state-of-the-art methods, our method is about $10\times$ faster than TOCH [13] and $24\times$ faster than CEM [3]. While our method requires a slightly longer time compared to CHOMP [8], ours significantly outperforms it across other metrics. Moreover, our method achieves the lowest smoothing errors, with values of $1.79cm/s^3$ and $1.80cm/s^3$ on the Grab and ContactPose, respectively. Figure 3 visualizes two grasp trajectories optimized with our method in the RaiSim [25] simulator.

**Grasps.** To evaluate the performance of our joint target optimization in target pose refinement, we compare our method with various state-of-the-art refinement methods, shown in Table II. Due to the limitation of the simulator, larger penetration may result in lower displacement [27]. Our method achieves $1.49cm$ SD and $1.39cm^3$ GPV. Though the SD of RefineNet is lower than that of our method, our method remarkably outperforms RefineNet in the GPV metric. It indicates that the grasp poses optimized by our method can lead to stable grasping with small penetration.

| Methods | JTO | HPoser | GSR (↑) | TPV(↓) | SE(↓) |
|---|---|---|---|---|---|
| Ours | ✗ | ✗ | 58.32 | 2.47 | 1.26 |
| Ours | ✗ | ✓ | 67.21 | 2.07 | 1.81 |
| Ours | ✓ | ✗ | 66.04 | 1.66 | **1.22** |
| Ours | ✓ | ✓ | **71.25** | **1.63** | 1.80 |
| CHOMP [8] | ✗ | - | 52.77 | 2.63 | 3.13 |
| CHOMP [8] | ✓ | - | 57.69 | 1.68 | 2.87 |
| Baseline | - | - | 44.33 | 4.78 | 1.41 |

TABLE III: Self-comparison on ContactPose test set.

In addition, our proposed method achieves 100% CR. According to all metrics, our proposed method can produce more stable and physically plausible grasp poses.

### *E. Ablation Study*

We mainly ablate the effectiveness of the HPoser module and the joint target optimization (JTO) strategy in this section. We construct four variants of our method, two variants of CHOMP [8] and a baseline, comparing their performances on the ContactPose test set. The baseline denotes the linear interpolation between the target and the start pose without optimization. The results are shown in Table III.

**Effectiveness of HPoser.** The main objective of the HPoser is to provide the prior constraint, ensuring that the hand pose remains natural throughout the trajectory optimization process. In the variant without the HPoser module, the temporal-based parametric optimization is directly applied to the finger pose $\theta_p \in R^{45}$. By comparing the third and the fourth row in Table III, a performance drop from 71% to 66% is witnessed with the absence of HPoser. By comparing the first and second rows in Table III, we can see that the GSR is improved by 15% when HPoser is employed, which indicates the effectiveness of the prior constraint provided by HPoser. It also can be observed that the smoothness of the trajectory may decrease when involving HPoser. The reason is that the introduction of prior constraints causes the method to prioritize the feasibility of the pose at each time step over the smoothness of the entire trajectory.

| $\text{Dim}_Z$ | GSR(↑) | TPV(↓) | SE (↓) | Time(↓) |
|---|---|---|---|---|
| w\o Z | 66.04 | 1.66 | **1.22** | **3.81** |
| 3 | 59.62 | 1.99 | 1.98 | 4.11 |
| 5 | **71.25** | 1.63 | 1.80 | 4.32 |
| 8 | 69.87 | 1.46 | 1.75 | 5.07 |
| 20 | 69.25 | **1.37** | 1.74 | 7.14 |

TABLE IV: Influence of different dimensionality of latent code **Z** on ContactPose test set.

**Effectiveness of Joint Target Optimization.** The JTO strategy is proposed to jointly refine the target pose during the optimization process. The variant without the JTO means that the target pose optimization is removed from the grasping trajectory optimization process. By comparing the second and fourth rows in Table III, we can see that the JTO strategy improves our method over all the metrics. To evaluate the generalization performance of the JTO strategy, we also combine the JTO with CHOMP [8]. From the last two rows in Table III, we can observe that the GSR is improved from 53% to 58%, and the TPV is improved from $2.63cm^3$ to $1.68cm^3$.

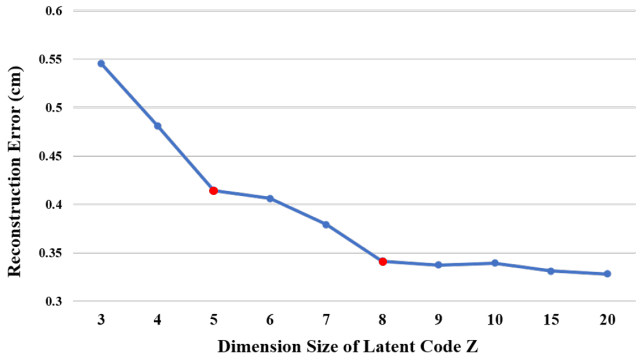


Fig. 4: Visualization of the reconstruction error curve with the growth of latent code size.

**Influence of Latent Code Dimensionality ($\text{Dim}_Z$).** In our method, we embed high-dimensional finger poses into a low-dimensional space. This inevitably results in some loss of fine-grained information, leading to a decrease in reconstruction quality and potentially interfering with trajectory optimization outcomes. In this section, we analyze the influence of the latent code dimensionality on trajectory optimization. We first report the reconstruction error with various $\text{Dim}_Z$ settings in Figure 4 where inflection points are observed at $\text{Dim}_Z$=5 and 8. Based on this, we list the detailed trajectory optimization metrics in Table IV, where we set $\text{Dim}_Z$ to 3, 5, 8, and 20. As a reference, we also duplicate the third row of Table III to Table IV (denoted by w\o Z). In general, it shows continuous improvement in smoothness and reduction in penetration with $\text{Dim}_Z$ increasing from 3 to 20. However, it requires higher time costs but receives no performance gains in terms of the grasp success rate (when $Dim_Z > 5$). Given these observations, we set $Dim_Z$ to 5 in our method as a trade-off.

## V. Conclusion

In this paper, we propose a lightweight yet powerful optimization algorithm for grasp motion planning, By optimizing temporal-based parameters and introducing the joint target optimization strategy, our method can yield smoother and more realistic grasp trajectories. The employment of the HPoser network enhances the understanding of latent hand-object interactions and functions as a robust constraint for optimizing hand pose changes. The proposed method is extensively validated on two public datasets. In terms of smoothness and stability, both quantitative and qualitative evaluations support that, our method has clear advantages over other strong competitors in planning high-quality grasp motion for the dexterous hand.

## REFERENCES


[1] L. E. Kavraki, P. Svestka, J.-C. Latombe, and M. H. Overmars, "Probabilistic roadmaps for path planning in high-dimensional configuration spaces," *IEEE transactions on Robotics and Automation*, vol. 12, no. 4, pp. 566–580, 1996.

[2] L. Janson, E. Schmerling, A. Clark, and M. Pavone, "Fast marching tree: A fast marching sampling-based method for optimal motion planning in many dimensions," *The International journal of robotics research*, vol. 34, no. 7, pp. 883–921, 2015.

[3] K. Lowrey, A. Rajeswaran, S. Kakade, E. Todorov, and I. Mordatch, "Plan online, learn offline: Efficient learning and exploration via model-based control," in *International Conference on Learning Representations*, 2018.

[4] N. Vahrenkamp, M. Do, T. Asfour, and R. Dillmann, "Integrated grasp and motion planning," in *2010 IEEE International Conference on Robotics and Automation*. IEEE, 2010, pp. 2883–2888.

[5] Y. Shu, Z. Li, B. F. Karlsson, Y. Lin, T. Moscibroda, and K. Shin, "Incrementally-deployable Indoor Navigation with Automatic Trace Generation," in *IEEE Conference on Computer Communications (INFOCOM)*, 2019.

[6] L. Wang, Y. Xiang, and D. Fox, "Manipulation trajectory optimization with online grasp synthesis and selection," *arXiv preprint arXiv:1911.10280*, 2019.

[7] 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*, vol. 37, no. 11, pp. 1319–1340, 2018.

[8] M. Zucker, N. Ratliff, A. D. Dragan, M. Pivtoraiko, M. Klingensmith, C. M. Dellin, J. A. Bagnell, and S. S. Srinivasa, "Chomp: Covariant hamiltonian optimization for motion planning," *The International journal of robotics research*, vol. 32, no. 9-10, pp. 1164–1193, 2013.

[9] T. Osa, "Multimodal trajectory optimization for motion planning," *The International Journal of Robotics Research*, vol. 39, no. 8, pp. 983–1001, 2020.

[10] S. M. LaValle, J. J. Kuffner, B. Donald *et al.*, "Rapidly-exploring random trees: Progress and prospects," *Algorithmic and computational robotics: new directions*, vol. 5, pp. 293–308, 2001.

[11] J. Schulman, J. Ho, A. X. Lee, I. Awwal, H. Bradlow, and P. Abbeel, "Finding locally optimal, collision-free trajectories with sequential convex optimization." in *Robotics: science and systems*, vol. 9, no. 1. Berlin, Germany, 2013, pp. 1–10.

[12] O. Taheri, V. Choutas, M. J. Black, and D. Tzionas, "Goal: Generating 4d whole-body motion for hand-object grasping," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 13 263–13 273.

[13] K. Zhou, B. L. Bhatnagar, J. E. Lenssen, and G. Pons-Moll, "Toch: Spatio-temporal object correspondence to hand for motion refinement," in *17th European Conference on Computer Vision*. Springer, 2022, pp. 1–19.

[14] J. J. Park, P. Florence, J. Straub, R. Newcombe, and S. Lovegrove, "Deepsdf: Learning continuous signed distance functions for shape representation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 165–174.

[15] T. Liu, Z. Liu, Z. Jiao, Y. Zhu, and S.-C. Zhu, "Synthesizing diverse and physically stable grasps with arbitrary hand structures using differentiable force closure estimator," *IEEE Robotics and Automation Letters*, vol. 7, no. 1, pp. 470–477, 2021.

[16] O. Taheri, N. Ghorbani, M. J. Black, and D. Tzionas, "Grab: A dataset of whole-body human grasping of objects," in *European conference on computer vision*, 2020, pp. 581–600.

[17] S. Brahmbhatt, A. Handa, J. Hays, and D. Fox, "Contactgrasp: Functional multi-finger grasp synthesis from contact," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2019, pp. 2386–2393.

[18] E. Corona, A. Pumarola, G. Alenya, F. Moreno-Noguer, and G. Rogez, "Ganhand: Predicting human grasp affordances in multi-object scenes," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 5031–5041.

[19] H. Li, X. Lin, Y. Zhou, X. Li, J. Chen, and Q. Ye, "Learning object affordance with contact and grasp generation," *arXiv preprint arXiv:2210.09245*, 2022.

[20] J. Romero, D. Tzionas, and M. J. Black, "Embodied hands: Modeling and capturing hands and bodies together," *ACM Transactions on Graphics*, 2017.

[21] G. Pavlakos, V. Choutas, N. Ghorbani, T. Bolkart, A. A. Osman, D. Tzionas, and M. J. Black, "Expressive body capture: 3d hands, face, and body from a single image," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 10 975–10 985.

[22] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.

[23] S. Brahmbhatt, C. Ham, C. C. Kemp, and J. Hays, "Contactdb: Analyzing and predicting grasp contact via thermal imaging," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 8709–8719.

[24] S. Brahmbhatt, C. Tang, C. D. Twigg, C. C. Kemp, and J. Hays, "Contactpose: A dataset of grasps with object contact and hand pose," in *European Conference on Computer Vision*, 2020, pp. 361–378.

[25] J. Hwangbo, J. Lee, and M. Hutter, "Per-contact iteration method for solving contact dynamics," *IEEE Robotics and Automation Letters*, vol. 3, no. 2, pp. 895–902, 2018. [Online]. Available: www.raisim.com

[26] H. Jiang, S. Liu, J. Wang, and X. Wang, "Hand-object contact consistency reasoning for human grasps generation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 11 107–11 116.

[27] K. Karunratanakul, J. Yang, Y. Zhang, M. J. Black, K. Muandet, and S. Tang, "Grasping field: Learning implicit representations for human grasps," in *2020 International Conference on 3D Vision*, 2020, pp. 333–344.

[28] X. Zhou, Z. Wang, H. Ye, C. Xu, and F. Gao, "Ego-planner: An esdf-free gradient-based local planner for quadrotors," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 478–485, 2020.

[29] P. Grady, C. Tang, C. D. Twigg, M. Vo, S. Brahmbhatt, and C. C. Kemp, "Contactopt: Optimizing contact to improve grasps," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 1471–1481.