Appendix

A. *Analysis of contact margins*

This section intends to justify why the feasibility of the initial and target state according to the proposed SpringGrasp metric is a good heuristic to indicate feasibility of the entire dynamic process. For simplicity of the analysis, we consider a problem with three fingertips that make contact with a 2D object as shown in Figure 14. In this system, the pose of the object can be described by $\boldsymbol{s}(t) = \{x(t), y(t), z(t), \theta(t)\}$. If the damping coefficients of each virtual spring are sufficiently large, no oscillation will happen during the dynamic process. This means that for any time, we can express it as convex combination of initial state and target state with a blending coefficient $\beta(t), t \in [t_0, t_{\text{eq}}]$ where $\beta(t_0) = 0$ and $\beta(t_{\text{eq}}) = 1$:

$$\boldsymbol{s}(t) = (1 - \beta(t))\boldsymbol{s}(t_0) + \beta(t)\boldsymbol{s}(t_{\text{eq}})$$

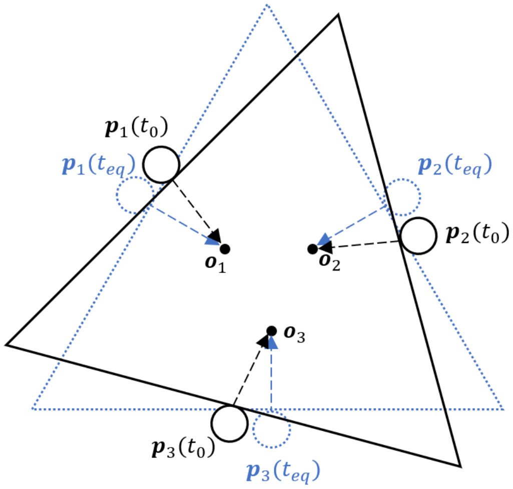

Fig. 14: Three fingertips making contact with a 2D triangle. Fingertips apply force toward $\boldsymbol{o}_i$ and move with the object from $\boldsymbol{p}_i(t_0)$ to $\boldsymbol{p}_i(t_{\text{eq}})$

.

1) *Translation only:*

We define the angle between the force vector $\boldsymbol{f}_i(t)$ and surface normal vector $\boldsymbol{n}_t(t)$ as $\alpha_i(t)$. To examine the temporal evolution of $\alpha_i(t)$, we initially consider a scenario involving only translation. Assuming the object undergoes a rigid translation $\boldsymbol{t}$ towards the equilibrium point, the initial force vector is given by:

$$\boldsymbol{f}_i(t_0) = k_i(\boldsymbol{o}_i - \boldsymbol{p}_i(t_0))$$

where k_i is controller gain of fingertip. At any subsequent time, the force vector will be:

$$\boldsymbol{f}_i(t) = k_i(\boldsymbol{o}_i - \boldsymbol{p}_i(t_0) - \beta(t)\boldsymbol{t}) = \boldsymbol{f}_i(t_0) - k_i\beta(t)\boldsymbol{t}$$

indicating that, during translation, the force vector changes from $\boldsymbol{f}_i(t_0)$ to $\boldsymbol{f}_i(t_0) - k_i\boldsymbol{t}$. The normal vector remains constant as translation does not affect the direction of the normal vector. Consequently, $\alpha_i(t)$ stays within the bounds defined by $\alpha_i(t_0)$ and $\alpha_i(t_{\text{eq}})$. Provided that the force directions at the initial and equilibrium states fall within the friction cone, the force direction at any transient state will also lie within this cone.

2) *Including rotation:*

When the dynamic grasping process involves rotation, the feasibility of the initial and equilibrium state does not guarantee the feasibility of the transient states. Consider a triangle rotating around its center $\boldsymbol{c}$ (see Figure 15). We choose a target location $\boldsymbol{o}_i$, where the distance between $\boldsymbol{c}$ and $\boldsymbol{o}_i$ is $\frac{\sqrt{2}}{2}r$, with r denoting the shortest distance from $\boldsymbol{c}$ to any edge of the triangle. Let the triangle rotate around its center until the force aligns with the surface normal vector. Then, $\alpha_i(t)$ initially increases to a maximum of $\alpha_{\text{max}} = \frac{\pi}{4}$ when $\boldsymbol{f}_i(t)$ is perpendicular to $\boldsymbol{o}_i - \boldsymbol{c}$, and then decreases to 0 at t_{eq}. If initially, angle $\angle \boldsymbol{p}_i(t_0)\boldsymbol{c}\boldsymbol{o}_i$ is less than $\frac{\pi}{4}$,$\alpha_i(t)$ will monotonically decrease and the force direction will always lies between the initial and equilibrium force directions. Bringing $\boldsymbol{o}_i$ closer to $\boldsymbol{c}$ reduces the change in $\alpha_i(t)$ for the same rotation angle on the object orientation $\theta(t)$, necessitating a greater rotation on $\theta(t)$ for $\alpha_i(t)$ to reach its maximum. Thus, adding E_{tar} to the energy function Eq.18 encourages the target location to stay deep inside the object, which allows more rotation on the object. Adding E_{reg} to Eq. 18 regulates movement during the dynamic process and reduces the rotation during the dynamic process. In practice, the feasibility of transient states can usually be inferred from the feasibility of initial and equilibrium states.

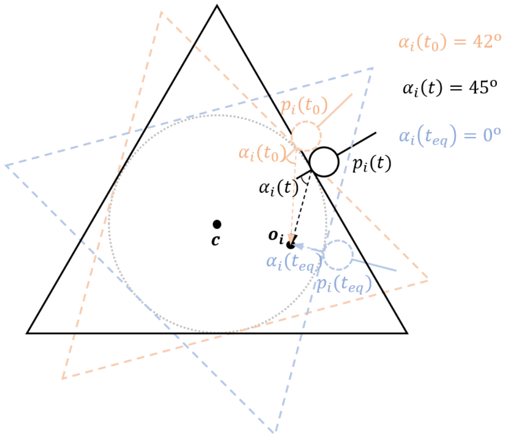

Fig. 15: Triangle rotate around $\boldsymbol{c}$, at time t, the force vector is perpendicular to vector $\boldsymbol{c} - \boldsymbol{o}_i$ and $\alpha_i(t) = \frac{\pi}{4} = 45^o$. In initial state and equilibrium state: $\alpha_i(t_0) = 42^o, \alpha_i(t_{\text{eq}}) = 0$

B. *Collision spheres*

Here we provide details on how we compute the energy terms $E_{\text{self,ho,ht}}$ in Eq 16 that are related to collisions with the hand itself, the object and table. Fig. 16 shows the placement of spheres that we use to approximate the geometry of the hand. $s_1 \sim s_{11}$ have a radius of 1cm and $s_{13} \sim s_{15}$ have radius of 2cm. For computing self-collision energy E_{self}, we find it is sufficient to check the following collision pairs: $(s_1, s_4), (s_1, s_7), (s_1, s_{10}), (s_4, s_7), (s_4, s_{10}), (s_7, s_{10}), (s_2, s_5), (s_5, s_8)$. When computing the hand object collision energy

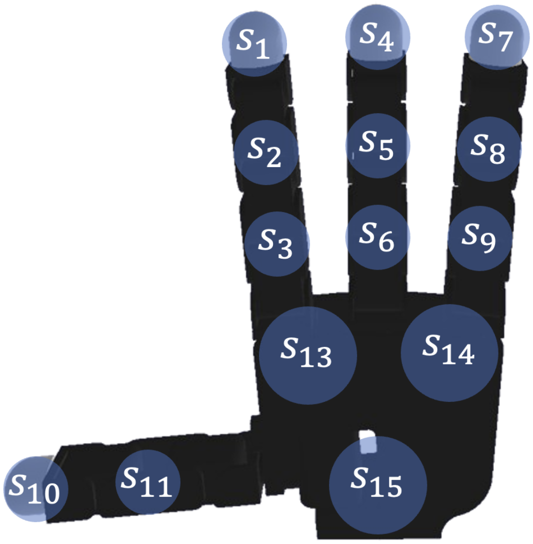

Fig. 16: Placement of collision spheres for computing collision loss, each sphere has radius of 0.015

E_{ho} and hand table collision energy E_{ht}, we use all collision spheres.

C. Energy function of baseline method

For our baseline approach, we define the following energy function:

$$E = w_{\text{fc}}E_{\text{fc}} + w_{\text{dist}}E_{\text{dist}} + w_{\text{col}}E_{\text{col}} + w_{\text{reg}}E_{\text{reg}} + w_{\text{uncer}}E_{\text{uncer}}$$

Most terms in this energy function are directly migrated from our method Eq. 18. We remove E_{gain} as gains are choosen independently hence no other variables depends on the gains in the optimization problem. Therefore, the optimizer would drive the controller gains directly to the lowest possible value. We also remove E_{tar} and E_{force} as force and target locations are the result of the inner force closure solver of the baseline and therefore cannot be controlled by the outer loop. We set $w_{\text{fc}} = 200$ and weights of other energy term is the same as Eq. 18. In [5, 4], E_{fc} is approximated by assuming that each fingertip can only apply a force along the contact normal with a fixed magnitude, which accelerates computation. We replace it with a more accurate force closure metric as defined in [6] and compute it using the differentiable convex optimization solver cvxpylayers[1] as:

$$E_{\text{fc}} = \min\,(||\sum_i^n \boldsymbol{f}_i||_2 + ||\sum_i^n \boldsymbol{p}_i \times \boldsymbol{f}_i||_2)$$

$$s.t.: \boldsymbol{f}_i \cdot \boldsymbol{n_i} \le -\frac{1}{\sqrt{1+\mu^2}}||\boldsymbol{f}_i||, \boldsymbol{f}_i \cdot \boldsymbol{n_i} \le -F_{\text{min}}$$

As the grasps optimized by the baseline method do not involve a dynamic process, we use $\boldsymbol{f}_i$, $\boldsymbol{p}_i$ without the time index. Because the force direction is determined by the force closure solver which cannot be controlled to reduce pregrasp uncertainty, we instead focus on reducing uncertainty at the fingertip contact location $\boldsymbol{p}_i$ and set the uncertainty energy term E_{uncer} as follows:

	Triangle 1	Triangle 2
config 1	(0.5,0.0), (0.75,0.5), (0.25,0.5)	(0.5,0.0), (1.0,0.5), (0.5,0.5)
config 2	(0.4,0.0), (0.8,0.4), (0.2,0.4)	(0.4,0.0), (1.0,0.4), (0.4,0.4)
config 3	(0.6,0.0), (0.7,0.6), (0.3,0.6)	(0.6,0.0), (1.0,0.6), (0.6,0.6)

Table V: Summary of fingertip contact configurations

$$E_{\text{uncer}} = \sum_i d_\sigma(\boldsymbol{p}_i)$$

Where d_σ is variance value function of GPIS as shown in Sec. III-D.

D. Grasp coverage experiment setup

Fig. 17 shows dimension of two triangles and positions of different contact configurations we used in experiment IV-I. To ensure the best coverage of our method during optimization, we randomly initialize 2000 target positions and controller gains when optimizing grasp for each object pose.

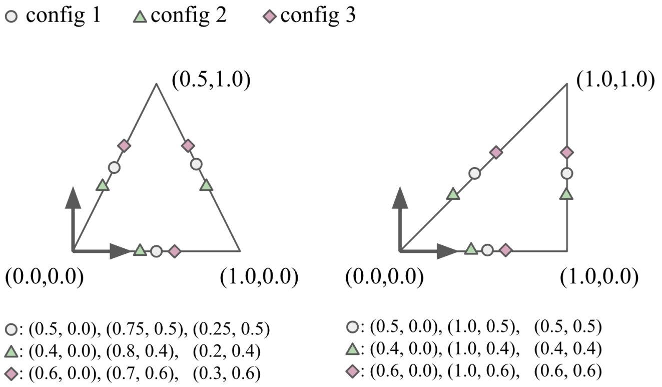

Fig. 17: Specification of two triangles. Vertices are expressed in the local object frame. Contact points in each contact configurations are expressed as different markers.

E. Clipping of spring grasp energy function

Here we illustrate implementation details of our spring grasp energy function E_{eq}. As the value of the contact margins $\epsilon_i(t_0), \epsilon_i(t_{\text{eq}})$ is in the range of $[-2, 1]$, in rare cases if contact margins are below -1 and logarithm mapping becomes undefined, directly clipping the value of the margin will set the gradient of the energy function to zero. Inspired by Leaky Relu [7], we switch to an auxiliary energy to encourage the contact force to stay close to surface normal if contact margins are below -1. Therefore, E_{sp} is defined as:

$$E_{\text{sp}} = -\sum_i^m \begin{cases} \log(\epsilon_i(t_0)+1) & \epsilon_i(t_0) > -1 \\ \log \frac{\boldsymbol{f}_i(t_0)}{||\boldsymbol{f}_i(t_0)||_2^2} \cdot \boldsymbol{n}_i(t_0) & \epsilon_i(t_0) \le -1 \end{cases}$$
$$-\sum_i^m \begin{cases} \log(\epsilon_i(t_{\text{eq}})+1) & \epsilon_i(t_{\text{eq}}) > -1 \\ \log \frac{\boldsymbol{f}_i(t_{\text{eq}})}{||\boldsymbol{f}_i(t_{\text{eq}})||_2^2} \cdot \boldsymbol{n}_i(t_{\text{eq}}) & \epsilon_i(t_{\text{eq}}) \le -1 \end{cases}$$

F. Fitting GPIS from partial point cloud

We estimate the true surface of an object from a partial point cloud by fitting a GPIS to it. Following [3], we use three sets of points to fit GPIS: a) Surface points, b) Exterior points, and c) Interior points. We assign a different value and noise level to each set of points.

1) Surface points: We use points from the partial point cloud as surface points. As the object surface is represented by the zero-level set of GPIS, we assign each point in the point cloud the value 0. We set the noise level of each point to be 0.005m.

2) Exterior points: To generate exterior points, we initially determine the axis-aligned bounding box of the surface points and scale it by 120% relative to the surface points' center. In total we have 14 exterior points which are located at each corner of the upscaled bounding box, as well as the midpoint of each bounding box edge. 14 points is sufficient for GPIS to distinguish between outside region and internal region. When fitting the GPIS, we found empirically that setting each exterior point value to be equal to half the length of the longest edge in the scaled bounding box works well. The noise level for each point is set to 0.2 meters.

3) Interior points: We compute interior points through convex combination of all surface points with random weights. To distribute these points evenly, rather than simply assigning weights from a uniform distribution to each surface point, we apply a softmax function to the initial random weights. This approach determines the actual weights for calculating interior points. For every object, we generate 50 interior points using this method. We found empirically that assigning each point a negative value equal to a quarter of the length of the longest edge of the upscaled bounding box works well. The noise level for each point is set at 0.05 meters.

G. Initial guesses for grasp planning

We initialize 7 wrist poses around the center of the oriented bounding box of the observed partial point cloud. The 7 poses consist of 5 with the palm facing the table and 2 with the palm perpendicular to the table. Tab.VI illustrates different initial guesses. Moreover, we always initialize joint angles as the relaxed joint pose defined in [17] and initialize the target position as a halfway point from each fingertip toward their center. Lastly, we use $k_1, k_2, k_3 = 80, k_4 = 160$ as initial controller gains.

H. Partial successes

To provide more information for Tab.II, we show the number of partial successes of each entry in Tab. VII.

wrist pose	position(xyz)	orientation(Euler)
1	-0.05, 0.0, 0.06	0, 0, 0
2	-0.04, 0.03, 0.03	0, 0, -45
3	-0.04, -0.03, 0.03	0, 0, 45
4	0.1, 0.06, -0.05	-90, 90, 0
5	0.0, 0.06, 0.05	-90, 0, 0
6	-0.0, -0.06, 0.03	0, 0, 90
7	0.02, -0.04, 0.03	0, 0, 135

Table VI: Initial wrist position and orientation, the unit of position is meter, and the unit of orientation is degree. We use intrinsic convention for Euler angles.

Appendix References

[A1] Akshay Agrawal, Brandon Amos, Shane T. Barratt, Stephen P. Boyd, Steven Diamond, and J. Zico Kolter. Differentiable convex optimization layers. In *Neural Information Processing Systems*, 2019.

[A2] Solak Gokhan. Allegro hand kdl. https://github.com/ARQ-CRISP/allegro_hand_kdl, 2020.

[A3] Miao Li, Kaiyu Hang, Danica Kragic, and Aude Billard. Dexterous grasping under shape uncertainty. *Robotics Auton. Syst.*, 2016.

[A4] Tengyu Liu, Zeyu Liu, Ziyuan Jiao, Yixin Zhu, and Song-Chun Zhu. Synthesizing diverse and physically stable grasps with arbitrary hand structures using differentiable force closure estimator. *RA-L*, 2022.

[A5] Ruicheng Wang, Jialiang Zhang, Jiayi Chen, Yinzhen Xu, Puhao Li, Tengyu Liu, and He Wang. Dexgraspnet: A large-scale robotic dexterous grasp dataset for general objects based on simulation. In *ICRA*, 2023.

[A6] Albert Wu, Michelle Guo, and Karen Liu. Learning diverse and physically feasible dexterous grasps with generative model and bilevel optimization. In *CoRL*, 2022.

[A7] Bing Xu, Naiyan Wang, Tianqi Chen, and Mu Li. Empirical evaluation of rectified activations in convolutional network. 2015.

	Ours	Ours	Ours	w/o uncertainty	w/o pregrasp	Bilevel-high	Bilevel-low	Bilevel-heu
	(3 views)	(2 views)	(1 view)	(1 view)	(1 view)	(2 views)	(2 views)	(2 views)
Mustard bottle	1	0	0	3	1	2	1	1
Lego	1	1	0	1	0	1	3	1
Pyramid	0	0	0	2	1	3	3	2
Campell can	0	0	0	1	0	1	1	2
Cheezit box	0	0	0	2	1	2	3	2
Mug	1	1	2	3	2	1	3	2
Orange	0	2	1	0	0	1	2	0
Coffee bottle	0	0	1	1	1	1	2	2
Spam	2	1	2	2	2	2	2	3
Plane	0	2	1	2	2	4	1	3
Car	1	1	1	3	3	2	2	4
Banana	0	0	1	2	1	1	0	2
Pear	0	0	1	0	2	2	2	2
Small box	0	0	0	1	3	2	2	1
Total	6	8	10	23	19	25	27	27

Table VII: Number of partial success for each experiment.