Supplementary Material

Haowen Luo[1], Yunze Liu[1,3], Li Yi[1,2,3]
[1]Tsinghua University, [2]Shanghai Artificial Intelligence Laboratory, [3]Shanghai Qi Zhi Institute

In this document, we provide a list of supplemental materials to support the main paper.

Calculation details of physical metrics. In Section 1, we provide the detailed calculation process of our physical metrics.

Choices of physical properties. In Section 2 we discuss our choice of the physical properties such as object mass and friction coefficient.

Additional ablation study. We show results from an additional ablation experiment in Section 3. In this experiment we explore how the data distribution of the dataset used to train the neural physical losses affect their performance.

Implementation details. In Section 4 we provide the implementation details of our work.

Failure cases. In Section 5 we discuss the failure cases of our method.

Limitations In Section 6, we discuss the limitations of our work.

1. Calculation details of physical metrics

To train the manipulation feasibility loss $\mathcal{L}_{\text{manip}}$, we first calculate the FE and ME metrics as hard and soft training targets for hand poses in the dataset.

For the force error metric FE, given object surface normal at the M contact points $\{\vec{n}_j\}_{j=1}^M$ and the target force= $\vec{F}$, to obtain the force error defined as:

$$FE(\{\vec{n}_j\}_{j=1}^M, \vec{F}) = \min_{\frac{\vec{f}_j}{||\vec{f}_j||}\cdot(-\vec{n}_j)\geq\sqrt{\frac{1}{1+\mu^2}}} ||\sum_{j=1}^M \vec{f}_j - \vec{F}||$$

we first simplify the optimization problem by using triangular pyramids to approximate the friction cones. To be specific, for the M contact points, we pick $\{V_j = [\vec{v}_j^1, \vec{v}_j^2, \vec{v}_j^3]\}_{j=1}^M$ as the normalized force basis. The angle between each column in $[\vec{v}_j^1, \vec{v}_j^2, \vec{v}_j^3]$ and $-\vec{n}_j$ is equal to the angle of friction computed with μ. $\vec{v}_j^1, \vec{v}_j^2$ and $\vec{v}_j^3$ form the three lateral edges of a regular triangular pyramid, as shown in Figure 1. The positive span of $\vec{v}_j^1, \vec{v}_j^2$ and $\vec{v}_j^3$ approximates the possible space of force $\vec{f}_j$ exerted at the j-th contact point. With this parametrization, we denote the force parameters at the j-th contact point as $p_j \in \mathbb{R}^3$, the optimization problem is then simplified as:

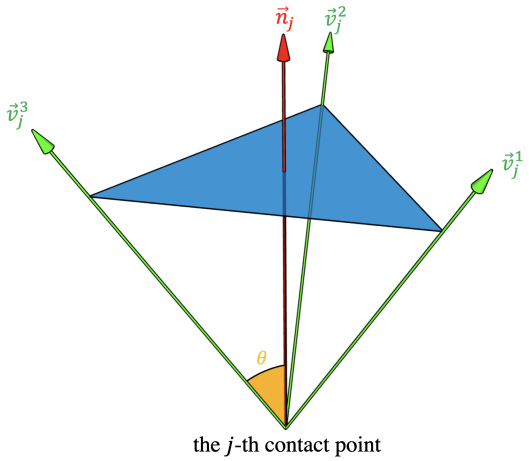

Figure 1. We parametrize the force applied at each contact point for computing the force error metric, θ is the friction angle.

$$FE'(\{\vec{n}_j\}_{j=1}^M, \vec{F}) = \min_{\{p_j\}_{j=1}^M} ||\sum_{j=1}^M V_j p_j - \vec{F}||,$$
$$\text{s.t. } p_j^1, p_j^2, p_j^3 \geq 0, j = 1, 2, \dots M$$

We solve for $\{p_j\}_{j=1}^M$ through an optimization with non-negative least square. Ideally, for hand poses that can manipulate the object feasibly, the objective will be optimized to 0. We threshold the FE value at $c_{FE} = 0.1||\vec{F}||$ to verify whether a given frame from a hand-object interaction sequence is feasible.

For the manipulation expense metric ME, we solve the constrained optimization using the penalty method. Given the object normal $\{\vec{n}_j\}_{j=1}^N$ at N points sampled on the object surface and $\{d_j\}_{j=1}^N$ that denotes the signed distance between these object points and their corresponding

hand points, we first obtain an intermediate representation $u \in \mathbb{R}^{256}$, which fuzzily depict the plausibility of applying forces to the object in 256 discrete directions. We sample 256 directions on the unit sphere and denote them using matrix $W = [\vec{w}_1, \vec{w}_2, \ldots, \vec{w}_{256}] \in \mathbb{R}^{3\times 256}$. The cosine similarity distances between these directions and the object normal directions are depicted by a matrix $Q \in \mathbb{R}^{N\times 256}$. The (j,k) element of Q is defined as:

$$Q_{jk} = \begin{cases} 1 & \text{if } (-\vec{n}_j)\cdot\vec{w}_k \geq \sqrt{\frac{1}{1+\mu^2}} \\ 0 & \text{otherwise} \end{cases}$$

The k-th element u_k of u is then defined as:

$$u_k = \min_{\substack{j=1,2,\ldots,N \\ Q_{jk}=1}} (|d_j| - c_{\text{contact}})^+$$

where $c_{\text{contact}} = 2\,mm$ denotes the contact threshold. Intuitively, the value u_k reflects how implausible it is to apply force in the direction of $\vec{w}_k$ to the object for the given hand pose. With this representation u, let $q \in \mathbb{R}^{256}$ denote the magnitude of forces applied in the 256 directions, then an approximation of the ME metric can be expressed as:

$$ME' = \min_{\substack{q_k \geq 0 \\ Wq=\vec{F}}} u \cdot q$$

We use penalty method to solve for q. To be specific, we turn the constrained optimization problem into an unconstrained one with penalties:

$$ME' = \min\ u\cdot q + M_e||Wq - \vec{F}|| + M_c \sum_{k=1}^{256} max(0, -q_k)$$

where M_e and M_c are large constants to force q to follow the constraint conditions. This unconstrained optimization problem can be solved with gradient descent.

2. Choices of physical properties

Object mass. Essentially, telling whether the given hand poses can feasibly manipulate the object along its trajectory doesn't require the absolute values of forces exerted at each contact point, and it's actually practically impossible to obtain the absolute values without sensors due to force ambiguity, i.e., different force combinations can have the same result. The force calculation process in the two metrics is more of an intermediate means for evaluating whether the given hand pose can possibly supply force in a certain direction (in force error), and if not, how far it is from the closest plausible hand pose (in manipulation expense). Our ultimate goal is to train a neural loss that differentiably quantifies the manipulation feasibility, with the knowledge of the two metrics.

Therefore, setting the mass to a fixed value doesn't influence our de-noising process since the object mass is only a relative value for solving forces, and doesn't affect resultant force distribution of the optimization process. Different mass values will result in the same force error since this metric reflects the relative error. The manipulation expense would be scaled proportional to the mass, but as long as we use the same mass value for the whole dataset, the relative values of manipulation expense can still reflect the plausibility of different hand poses.

Friction coefficient. Setting a fixed friction coefficient is a common practice in previous works [1], [5] though it could theoretically influence the denoising process (for example if we set the friction coefficient to zero it would be very hard to support the manipulation anymore). In practice, we find our method to be quite robust. Our early observation reveals that as long as the selected friction coefficient isn't too off, the result of force error and manipulation expense can align well with human perception concerning the manipulation feasibility. A better solution might be training neural losses with different friction coefficients and using system identification to adapt the method to different objects, yet that would be out of this work's scope.

3. Additional ablation study

Data variation in training neural physical loss. To improve the generalization ability and the smoothness of the loss landscape of our neural physical losses, we use interpolation of the MANO [4] parameters to assure the data variation in the dataset used to train the losses. To verify the effectiveness of this design, we use two different schemes to generate two other training datasets. For a given HOI dataset containing ground truth data $\{(H^i, O^i)\}_{i=1}^{D}$, we i) add $(m-1)$ different sets of Gaussian noise with a fixed standard deviation σ to the hand MANO parameters of each HOI frame, and ii) add Gaussian noise with standard deviations of $\{\frac{j}{m-1}\sigma\}_{j=1}^{m-1}$ to the MANO parameters of each frame to get the perturbed version of the HOI dataset. The perturbed poses generated in these two ways are mixed with the ground truth poses to form two training datasets for the neural loss terms respectively. Notice that both these two dataset have the same size as the dataset obtained with interpolation. The performance of loss terms trained on all three datasets is shown in Table 2.

P

Adding object acceleration as the network input. We evaluate the influence of adding object acceleration as the network input. As the results in 1 show, the information of object acceleration helps with producing physically plausible manipulation, while whether to include that information in stage I makes little difference.

Table 1. Acc- denotes the network that doesn't consume acceleration in either training stage, Acc I + II denotes the network that consumes acceleration in both stage I and stage II, while Acc II only takes acceleration as input in stage II.

	MPJPE	MPVPE	contactIoU	IV	pd	plausible rate
Acc-	7.56	6.89	22.15	**1.11**	**0.43**	0.85
Acc I + II	7.40	**6.76**	23.90	1.13	0.45	**0.91**
Acc II	**7.39**	6.78	**23.94**	1.13	0.44	**0.91**

Table 2. We train the two neural physical loss terms on perturbed datasets with different noise distribution. Fixed SD refers to the dataset obtained by adding Gaussian noise of a single fixed standard deviation to ground truth hand poses, while various SD refers to the dataset obtained by adding noise of various standard deviation to the ground truth. The dataset in our method contains paths from noisy poses to clean ones obtained by interpolation. Neural physical losses trained on this dataset exhibit better performance when exploited.

(a) Distribution of noise in the perturbed dataset used when training grasp credibility loss.

	PD	IV
fixed SD	0.51	1.72
various SD	0.45	1.15
interpolation	0.44	1.13

(b) Distribution of noise in the perturbed dataset used when training manipulation feasibility loss.

	plausible rate
fixed SD	0.84
various SD	0.89
interpolation	0.91

4. Implementation details

Our pipeline is implemented on Python 3 and PyTorch. We use the Adam optimizer [2] and the cosine annealing warm restart [3] learning rate scheduler for training of the neural loss terms and the de-noising network. At inference time, our pipeline runs at approximately 1 fps on Intel Xeon Platinum 8358 CPU @ 2.60GHz.

5. Failure Cases

We observe that when the noisy input deviates largely from the correct pose, e.g. large translation error, our method occasionally fails to recover the correct pose, but rather falls into other plausible results. Besides, as our method only considers one-hand scenario currently, it cannot well handle the cases of two-hand manipulation which involve object-hand interaction and hand-hand interaction.

6. Limitations

We only consider the setting of in-hand manipulation of rigid objects. However, in real human-object interaction, it's common that the object being manipulated also receive forces from the environment beside human hand. For instance, in the scenario of dragging a chair, the chair also receive a supporting force from the ground. Besides, to obtain physically plausible interaction sequences between hand and articulated objects, more sophisticated dynamic model should be used to consider the interaction between different articulated parts.

References

[1] Haoyu Hu, Xinyu Yi, Hao Zhang, Jun-Hai Yong, and Feng Xu. Physical interaction: Reconstructing hand-object interactions with physics. In *SIGGRAPH Asia 2022 Conference Papers*, pages 1–9, 2022. 2

[2] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 3

[3] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016. 3

[4] Javier Romero, Dimitrios Tzionas, and Michael J Black. Embodied hands: Modeling and capturing hands and bodies together. *arXiv preprint arXiv:2201.02610*, 2022. 2

[5] He Zhang, Yuting Ye, Takaaki Shiratori, and Taku Komura. Manipnet: neural manipulation synthesis with a hand-object spatial representation. *ACM Transactions on Graphics (ToG)*, 40(4):1–14, 2021. 2