Supplementary Materials

I. TECH

A. High-level Task and Motion Planning via LLM

In our work, we develop an LLM/LVM-based object motion planner that can generate collision-free dense trajectories that follow high-level language instructions $\mathcal{I}$ from users that are long-horizon and require contextual understanding of the embodied world. In our work, the planner consists of the following components: 1) an LLM/LVM that is prompted with the description of the task to generate sparse 6D arrays that might contain infeasible waypoints or collision, 2) An open-vocabulary object detector to obtain spatial-geometrical information of the relevant objects, or even portions of the object (e.g. "the body of the microwave" or "the handle of the cup"), and 3) a sampling-based motion planner to generate collision-free dense trajectory. To obtain a sparse trajectory that roughly describes the pose change of manipulated objects from the initial states to the goal states, the planner first transforms the $\mathcal{I}$ (e.g. "Put the cup in the microwave" and "Pull the rope into the shape of N, U and S") into several decompositions $\mathcal{I} \rightarrow (i_1, i_2, ..., i_n)$. To obtain precise geometrical information about each object, We employ the multimodal detector [1] to generate 2D bounding boxes around each object, using object names as prompts, these 2D coordinates are then projected onto 3D space to gather the spatial-geometry information corresponding to the robot perspective coordinates. With the augmentation geometric information of objects, the LLM then generates a trajectory τ_i of the manipulated objects for each manipulation phase described by the sub-instruction i_t in a chain-of-thought [2], [3] style, where each waypoint consists of 6-DoF object pose. However, trajectories proposed by the LLM are not entirely reliable, as the text description and image of the environment does not contain all the comprehensive physics information of the embodied environment, and as the environment is not fully observable to the LLM/VLM, so that the waypoints may contain infeasible states or collisions with the surrounding environment. To avoid giving the foundation model infeasible waypoints, we adopt a sampling-based motion planning method to interpolate the sparse trajectory τ_i to a collision-free and feasible dense trajectory $OPT(\tau_i)$.

B. Middle-level Motion Planning via Simulation

To enable flexible collision checking while motion planning, we represent all the objects with their point clouds and load them in the Jade simulator [4]. We adopt fcl [5] for real-time collision checking for objects with the surroundings in the simulation environment. We use the following RRT-connect motion planning algorithm 1 to find a collision-free dense trajectory.

Algorithm 1 Generate Collision-Free Dense Trajectory using RRT-Connect

Require: Sparse trajectory τ_i from LLM/LVM, initial state s_{init}, goal state s_{goal}, obstacle set $\mathcal{O}$
Ensure: Dense trajectory $OPT(\tau_i)$ that is collision-free
  Initialize T_{start} with root s_{init}
  Initialize T_{goal} with root s_{goal}
  while T_{start} and T_{goal} not connected **do**
    $s_{\text{rand}} \leftarrow$ SampleRandomState()
    $s_{\text{near}} \leftarrow$ NearestNeighbor(T_{start}, s_{rand})
    $s_{\text{new}} \leftarrow$ Steer(s_{near}, s_{rand})
    if not InCollision(s_{new}, $\mathcal{O}$) **then**
      Add s_{new} to T_{start}
      if CanConnect(s_{new}, T_{goal}) **then**
        Connect T_{start} and T_{goal} through s_{new}
      end if
    end if
    Swap(T_{start}, T_{goal})
  end while
  $OPT(\tau_i) \leftarrow$ PathBetweenTrees(T_{start}, T_{goal})
  $OPT(\tau_i) \leftarrow$ OptimizePath($OPT(\tau_i)$)
  return $OPT(\tau_i)$

II. DATASET

We create a large-scale comprehensive annotated dataset that includes articulated/rigid objects and deformable objects in 1D/2D/3D forms, as shown in II.

A. Articulated/ Rigid Bodies

Object Pre-processing We collect 100K+ object models from 1K+ categories in Objaverse [6], ShapeNet [7], ABC [8], Thingi10K [9], and GAPartNet [10]. From these datasets, we select 100K+ objects in 1K+ categories and normalize all models into a unit box and augment each object by randomly scaling them with 4 sizes between 0.05 and 0.4. Then we remesh them into manifolds [11], abandon those with low volume and translate the obtained mesh so that its coordinate origin coincides with its center of mass. Finally, for simulation purposes, we create collision meshes for every object mesh through convex decomposition using CoACD [12].

Annotations Our dataset contains millions of training examples. Each training example is composed of the input and the ground truth label. The input of each example is the object oriented point cloud $\mathcal{P}_o$, robot manipulator's point cloud $\mathcal{P}_h$ at its rest pose, task motion $\mathcal{M}$, manipulation

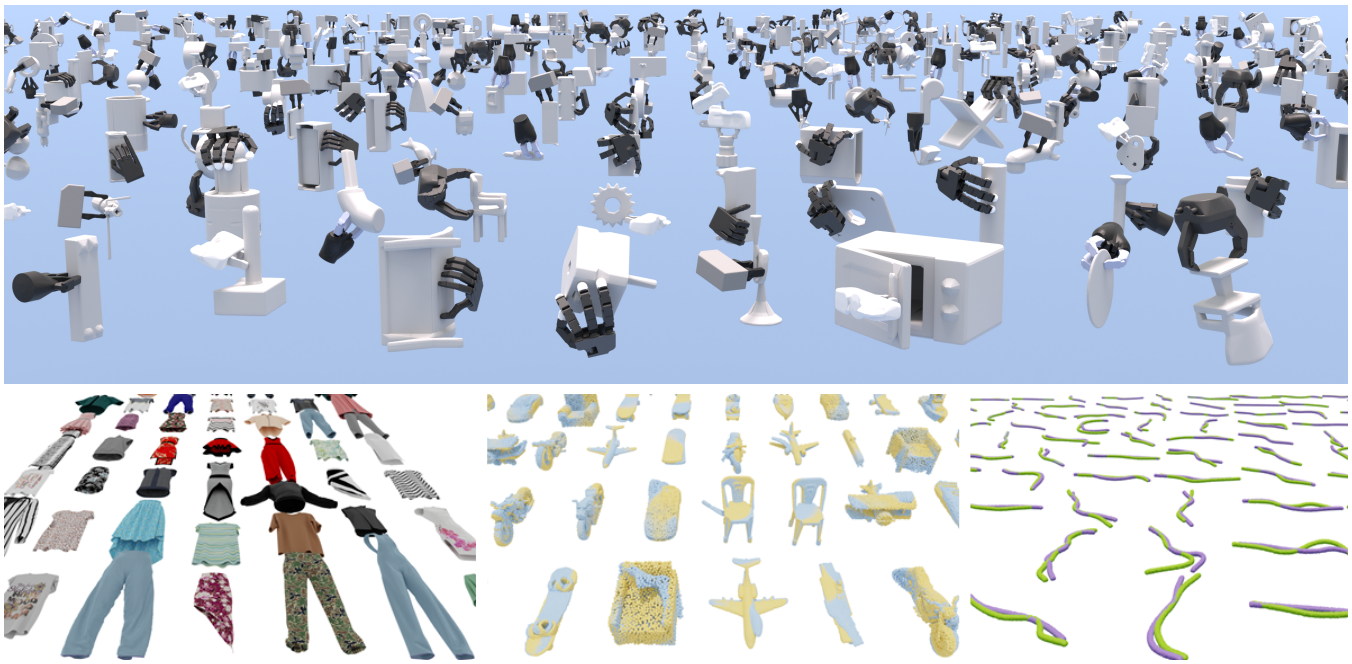

Fig. 1. Dataset visualization. Our large-scale comprehensive annotated dataset includes articulated/rigid and deformable objects in 1D/2D/3D forms.

region $\mathcal{R}$, and friction coefficient. The ground truth label contains the ground truth contact point heatmap and contact force heatmap. We propose a unified sampling-based method for large-scale contact synthesis. We adopted a hierarchical sampling approach to obtain suitable contact training examples. For a scaled object model, we sample points on its outer surface and obtain their corresponding normals. Since solving the inverse kinematics (IK) for multiple fingers with a floating base is relatively challenging and time consuming, we propose first sample palm poses facing the object, solve IK for each finger, and perform combinations on all finger solutions. We then sample contact forces within all the contact friction cones. At last, we calculate the sum of the wrenches exerted by all fingers to the object. Here we focus on point contact between the tips of robotic hands and the object surface. For the manipulation region defined as $\mathcal{R} = \{r_i\}_{i=1}^{N}$, there are 50% probability that $r_i = 1$ for all $i = 1 \dots N$, representing no constraints on the output contact heatmap. Conversely, the points where $r_i = 1$ encompass the ground truth contact points along with points in their vicinity.

B. 2D Deformable Object

Object Pre-processing We collect 2K+ clothes meshes from 4 categories and 19 sub-categories in ClothesNet [13]. For each clothes mesh, we normalize it into a unit box and simplify the mesh with quadric decimation [14] to under 3500 vertices. Objects with more than one connected component are filtered out.

Annotations Our dataset contains 100K+ training examples. Each training example has the same data format as rigid body. The physical properties of the cloth include its density, frictional coefficient, and elasticity coefficients. The manipulation region are all set to one. Our 2D deformable dataset have two parts. One part is collected during clothes folding. The other is collected during random drags. Specifically, we first rotate the clothing so that the front side faces upwards, and let it free-falls in the DiffCloth [15] simulator until it lands flat on a plane, defining this as the initial object state. Then for the first part, we define a set of contact point based on key points detected by Skeleton Merger [16] and generate target motion trajectories for folding the cloth.

C. 3D Deformable Object

Object Pre-processing For 3D Deformable objects, we collect 1K object models from 10 categories in ShapeNet [7]. All the objects are normalized into a unit ball. Then, we use farthest point sampling (FPS) to sample 2048 points on the surface. We use SoftMAC [17], a particle-based deformable object simulator, to simulate the motion of the objects when external forces are added.

Annotations For each object, we collect 100 training examplse. So, the total size of our 3D deformable object dataset is 100K. Each training example has the same data format as rigid body. The physics property includes friction coefficient, Young's modulus, and Poisson ratio. Our data annotation process is structured as follows: For an object composed of particles, we initially apply the K-means clustering method to divide all points into 100 clusters. Subsequently, we randomly select one group and then randomly choose a force to apply to all particles in this group. Clustering is adopted because the entire object is composed of 2048 particles. Applying a force to just one particle would have a minimal impact on the object's overall shape. Therefore, for a noticeable deformation in the entire object, we apply the same force to all particles within a selected cluster simultaneously.

Letter	Mean Dis(m)	Max Dis(m)	Success Rate
N	0.0210	0.0376	10/10
U	0.0186	0.0394	9/10
S	0.0183	0.0384	8/10

TABLE I

ROPE REARRANGEMENT EXPERIMENT RESULTS.

III. EXPERIMENT

A. Simulation Experiments Metric for Deformable Objects

We apply the predicted contacts on the objects in simulation. We use DiffCloth [15] for 2D deformable objects and SoftMAC [17] for 1D/3D deformable objects. We can calculate the object point cloud after applying contact forces in simulators. The success rate is defined based on the distance between the results from the object point cloud and the target point cloud.

Specifically, for 2D deformable objects, we evaluate using 50 points with the most motion. A prediction fails if the max L2 error exceeds 0.5m, or over 50% of the selected points have an L2 error over 0.3m. For 3D deformable objects, we select 100 points. Success requires an average L2 error under 0.5 cm and an average distance for selected points under 5cm. During evaluation, all point clouds are normalized into a cube with side length of 2m.

B. Real World Experiments

Fig. 2. Realworld experiment settings.

In this section, we evaluate our system's performance of manipulating various rigid, articulated rigid and deformable objects in several real-world settings.

Rope Rearrangement We design a rope rearrangement experiment for a Kinova MOVO robot arm to rearrange a random reset rope to letters "NUS". We perform 10 times for each letter in "NUS". Specifically, we take RGBD images for both the current scene and the goal scene, and get the rope point cloud with SAM [18]. We then approximate the rope point clouds with several cylinders so that we can calculate the per-point target motion by obtaining the cylinder's planar rotation and translation from current to goal scene. We iteratively grasp the point with the highest predicted contact heatmap value and move to the predicted target point. A test case visualization is shown in 3. As a result, our model achieves 90% success rate, as shown in III-B, indicates that our model is accurate in generating grasp points for deformation objects like ropes.

Breakfast Preparation We design a complex breakfast preparation experiment for a single Flexiv robot arm and a LeapHand as the manipulator. In this setting, the robot will open the fridge door, take out the milk box, place the milk box on the table, pick up a piece of bread, put the bread into the toaster, open the toaster, take out the cooked bread and place it to the plate. Among these objects, the fridge door is an articulated object, while the milk box is a normal rigid body object, and the bread is treated as a 3D deformation object.

Cloth Folding We design a cloth folding experiment for a Kinova MOVO robot arm to fold a T-shirt. We extract the T-shirt point clouds using SAM [18]. Movo will then grasp the point with the highest heatmap value and move to the predicted target point.

REFERENCES

[1] T. Ren, S. Liu, A. Zeng, J. Lin, K. Li, H. Cao, J. Chen, X. Huang, Y. Chen, F. Yan, Z. Zeng, H. Zhang, F. Li, J. Yang, H. Li, Q. Jiang, and L. Zhang, "Grounded sam: Assembling open-world models for diverse visual tasks," 2024.

[2] J. Liang, W. Huang, F. Xia, P. Xu, K. Hausman, B. Ichter, P. Florence, and A. Zeng, "Code as policies: Language model programs for embodied control," in *arXiv preprint arXiv:2209.07753*, 2022.

[3] Z. Yu, J. Fu, Y. Mu, C. Wang, L. Shao, and Y. Yang, "Multireact: Multimodal tools augmented reasoning-acting traces for embodied agent planning," in *6th Robot Learning Workshop NeurIPS 2023: Pretraining, Fine-Tuning, and Generalization with Large Scale Models*, 2023. [Online]. Available: https://openreview.net/forum?id=pXDr36kovo

[4] G. Yang, S. Luo, and L. Shao, "Jade: A differentiable physics engine for articulated rigid bodies with intersection-free frictional contact," *arXiv preprint arXiv:2309.04710*, 2023.

[5] J. Pan, S. Chitta, and D. Manocha, "Fcl: A general purpose library for collision and proximity queries," in *2012 IEEE International Conference on Robotics and Automation*, 2012, pp. 3859–3866.

[6] M. Deitke, D. Schwenk, J. Salvador, L. Weihs, O. Michel, E. VanderBilt, L. Schmidt, K. Ehsani, A. Kembhavi, and A. Farhadi, "Objaverse: A universe of annotated 3d objects," *arXiv preprint arXiv:2212.08051*, 2022.

[7] A. X. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su, *et al.*, "Shapenet: An information-rich 3d model repository," *arXiv preprint arXiv:1512.03012*, 2015.

[8] S. Koch, A. Matveev, Z. Jiang, F. Williams, A. Artemov, E. Burnaev, M. Alexa, D. Zorin, and D. Panozzo, "Abc: A big cad model dataset for geometric deep learning," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 9601–9611.

[9] Q. Zhou and A. Jacobson, "Thingi10k: A dataset of 10,000 3d-printing models," *arXiv preprint arXiv:1605.04797*, 2016.

[10] H. Geng, H. Xu, C. Zhao, C. Xu, L. Yi, S. Huang, and H. Wang, "Gapartnet: Cross-category domain-generalizable object perception and manipulation via generalizable and actionable parts," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 7081–7091.

[11] J. Huang, Y. Zhou, and L. Guibas, "Manifoldplus: A robust and scalable watertight manifold surface generation method for triangle soups," *arXiv preprint arXiv:2005.11621*, 2020.

[12] X. Wei, M. Liu, Z. Ling, and H. Su, "Approximate convex decomposition for 3d meshes with collision-aware concavity and tree search," *ACM Transactions on Graphics (TOG)*, vol. 41, no. 4, pp. 1–18, 2022.

[13] B. Zhou, H. Zhou, T. Liang, Q. Yu, S. Zhao, Y. Zeng, J. Lv, S. Luo, Q. Wang, X. Yu, H. Chen, C. Lu, and L. Shao, "Clothesnet: An information-rich 3d garment model repository with simulated clothes environment," in *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023.

[14] M. Garland and P. S. Heckbert, "Surface simplification using quadric error metrics," in *Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, 1997, pp. 209–216.

[15] Y. Li, T. Du, K. Wu, J. Xu, and W. Matusik, "Diffcloth: Differentiable cloth simulation with dry frictional contact," *ACM Transactions on Graphics (TOG)*, vol. 42, no. 1, pp. 1–20, 2022.

[16] R. Shi, Z. Xue, Y. You, and C. Lu, "Skeleton merger: an unsupervised aligned keypoint detector," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 43–52.

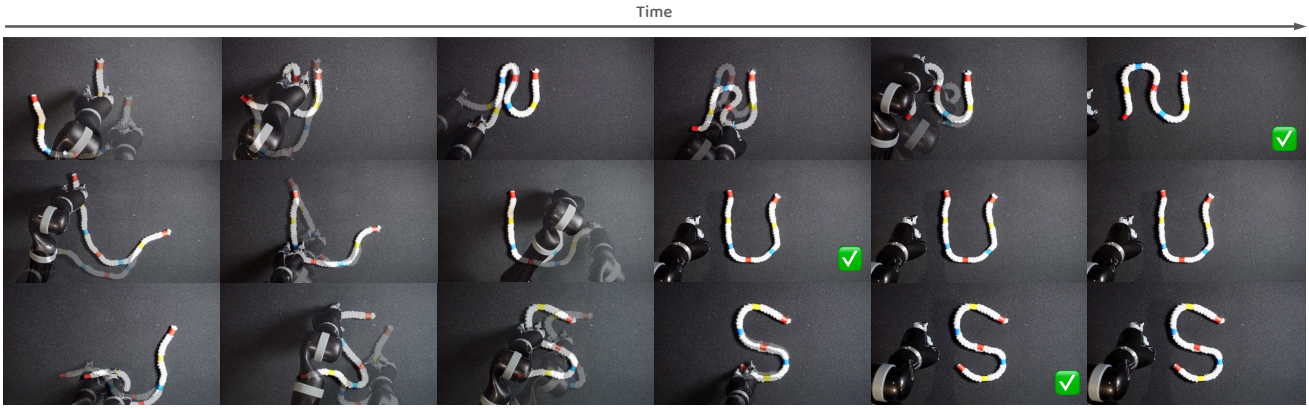

Fig. 3. Test Cases of Rope Rearrangement.

[17] M. Liu, G. Yang, S. Luo, C. Yu, and L. Shao, "Softmac: Differentiable soft body simulation with forecast-based contact model and two-way coupling with articulated rigid bodies and clothes," 2023.

[18] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, P. Dollár, and R. Girshick, "Segment anything," *arXiv:2304.02643*, 2023.