SAPIEN: a SimulAted Part-based Interactive ENvironment
Supplementary Material

Fanbo Xiang[1] Yuzhe Qin[1] Kaichun Mo[2] Yikuan Xia[1] Hao Zhu[1]
Fangchen Liu[1] Minghua Liu[1] Hanxiao Jiang[3] Yifu Yuan[5] He Wang[2] Li Yi[4]
Angel X. Chang[3] Leonidas Guibas[2] Hao Su[1]
[1]UC San Diego [2]Stanford University [3]Simon Fraser University [4]Google Research [5]UC Los Angeles
Website: https://sapien.ucsd.edu
Video Demo: https://youtu.be/K2yOeJhJXzM

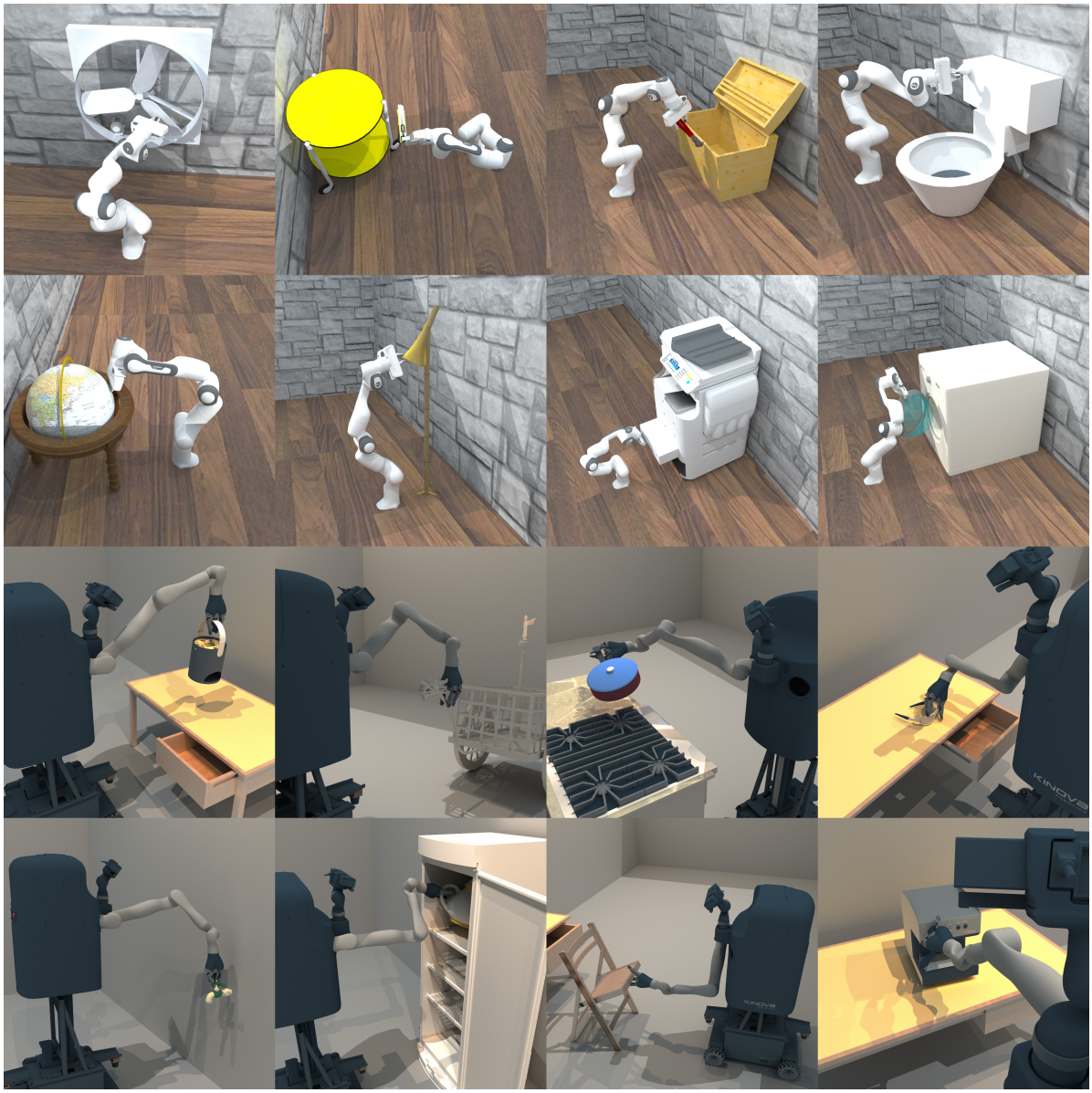

Figure 1: **Diverse manipulation tasks supported by SAPIEN**

Table of Contents

Appendix A: Annotation System

We developed a web interface (Figure 2) for mobility annotation. This tool is a question answering (QA) system, which proposes questions based on current stage of annotation. It exploits the hierarchical structures of PartNet to propose objects without relative mobility, and generates new questions based on past annotations. Using this tool, annotators will not miss any movable parts if they answer every question correctly, and they will not face any redundant questions by design. The output mobility annotations are guaranteed to satisfy tree properties suitable for simulation.

The annotation procedure has the following steps:

- We start with a PartNet semantic tree, and traverse the tree nodes. Annotators are prompted with questions asking if current subtree has relative motion. If it does not, all parts in this tree will be fixed together; otherwise, the same question is asked again on the child nodes of this subtree.
- When the PartNet semantic tree traversal is finished, annotators are asked to choose parts that are fixed together.
- Next, annotators are asked to choose parts that are connected with a hinge (rotational) joint. They will then choose parent-child relation, and annotate axis position/motion limit with our 3D annotation tool.
- Next, annotators are asked to choose parts that are connected with a slider (translational) joint. They will similarly choose motion parameters and decide if this axis also bears rotation (screw joint).
- Finally, annotators will annotate each separate object in the scene as "fixed base", "free"", or "out lier".

The procedure is summarized in the following pseudo-code block.

Annotating PartNet-Mobility dataset

```
Propose fixed parts based on PartNet tree
for There are parts can be fixed together do
    Select a group of relatively fixed parts
end for
for Rotation relationship exists do
    Select parent and child
    Pick rotation axis
    Input motion range
end for
for Translation relationship exists do
    Select parent and child
    Pick translation axis
    Input motion range / whether it can also rotate
end for
Choose whether root nodes are fixed/free
```

Appendix B: Movable Part Segmentation and Motion Recognition

Movable Part Segmentation: complete results

Table 1 shows the movable part segmentation results for all categories in PartNet-Mobility dataset.

Motion Recognition: experiment details

For this task, we normalize the $[0, 2\pi]$ hinge joint range to $[0, 1]$. For sliders, we normalize by the maximum motion range over the dataset to make the motion range prediction within $[0, 1]$.

Algorithm. The baseline algorithm we use is a a ResNet[1] classification and Regression network. The input is the ground truth RGB-D image and the segmentation mask for the target movable part. The output has 7 terms:
$\hat{T}_r \in \{0, 1\}$, whether this part has a rotational joint.
$\hat{T}_t \in \{0, 1\}$, whether this part has a translational joint.
$\hat{\mathbf{p}}^r \in \mathbb{R}^3$, pivot of a predicted rotational axis.
$\hat{\mathbf{d}}^r \in [-1, 1]^3$, direction of a predicted rotational axis.
$\hat{\mathbf{d}}^t \in [-1, 1]^3$, direction of a predicted translational axis.
$\hat{x}_{\text{door}} \in [0, 1]$, predicted joint position for a door.
$\hat{x}_{\text{drawer}} \in [0, 1]$, predicted joint position for a drawer.

In the following, letters without hat indicates their corresponding ground-truth labels.

In our experiment, we modify the input layer of a ResNet50 network to accept 5 channels, and output layer to output 13 numbers. In addition, we apply tanh activation to produce $\hat{\mathbf{d}}^r, \hat{\mathbf{d}}^t$, and sigmoid activation to produce $\hat{x}_{\text{door}}, \hat{x}_{\text{drawer}}$. The loss has 7 terms:
Axis alignment loss, measured by cosine distance:

$$L_{dr} = \sum_{T_r=1} 1 - |\frac{\mathbf{d}^r \cdot \hat{\mathbf{d}}^r}{||\mathbf{d}^r|| ||\hat{\mathbf{d}}^r||}| \quad L_{dt} = \sum_{T_t=1} 1 - |\frac{\mathbf{d}^t \cdot \hat{\mathbf{d}}^t}{||\mathbf{d}^t|| ||\hat{\mathbf{d}}^t||}|$$

Pivot loss, measured by the distance from predicted pivot to ground truth joint axis:

$$L_p = \sum_{T_r=1} ||\hat{\mathbf{p}}^r - \mathbf{p}^r - ((\hat{\mathbf{p}}^r - \mathbf{p}^r) \cdot \mathbf{d}^r)\mathbf{d}^r||_2^2$$

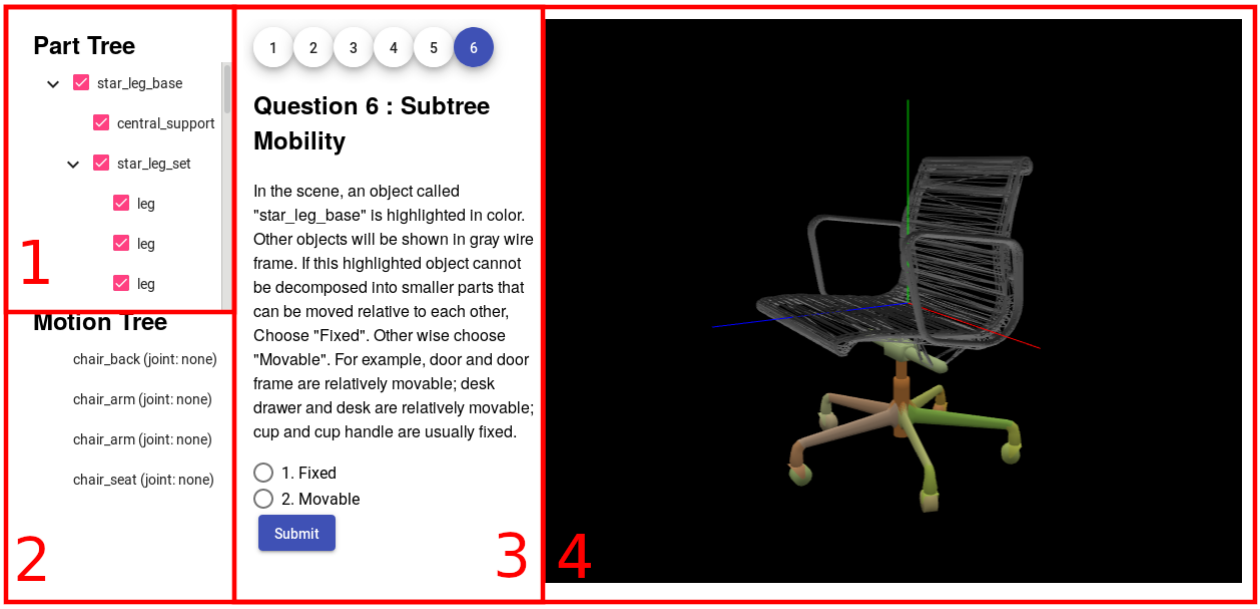

Figure 2: **Annotation interface.** 1) Part Tree: PartNet semantic tree that proposes fixed parts. 2) Motion tree: annotated movable parts. 3) Question: auto-generated exhaustive questions. 4) Visualization for current question and for motion axis annotation.

Joint type prediction loss:

$$L_{T_r} = -\sum T_r \log \hat{T}_r + (1 - T_r) \log(1 - \hat{T}_r)$$

$$L_{T_t} = -\sum T_t \log \hat{T}_t + (1 - T_t) \log(1 - \hat{T}_t)$$

Joint position loss, L_2 loss between predicted position and ground truth position.

$$L_{\text{door}} = \sum_{\text{valid hinge}} (x_{\text{door}} - \hat{x}_{\text{door}})^2$$

$$L_{\text{drawer}} = \sum_{\text{valid slider}} (x_{\text{drawer}} - \hat{x}_{\text{drawer}})^2$$

The final loss is a summation of all the losses above:

$$L = L_{dr} + L_{dt} + L_p + L_{T_r} + L_{T_t} + L_{\text{door}} + L_{\text{drawer}}$$

This objective is optimized on mini-batches using proper masking based on H and S values.

We repeat this experiment with PointNet++[4] operating on 3D RGB-point cloud produced by the same images. For each image, we sample 10,000 points from the partial point cloud (create random copies if the total number of points is less than 10,000). Figure 3 shows the network structure for the motion recognition tasks.

Appendix C: Terminology

SAPIEN Engine

- **Articulation:** An articulation is composed of a set of links connected together with transnational or rotational joints [3]. The most common articulation is a robot.
- **Kinematic/Dynamic joint system:** Both joint systems are an assembly of rigid bodies connected by pairwise constraints. Kinematic system does not respond to external forces while dynamic objects do.
- **Force/Joint/Velocity Controller:** Controller which can control the force/position/velocity of one or multiple joints at once. Like real robot, controller may fail depending on whether the target is reachable.
- **Inertial Measurement Unit(IMU):** A sensor which can measure the orientation, acceleration and angular velocity of the mounted link.
- **Trajectory Controller:** A controller which receive trajectory command and execute to move through the trajectory points. Note that trajectory consist of a sequence of position, velocity and acceleration, while path is simply a set of points without a schedule for reaching each point [2].
- **End-effector:** End-effector is a manipulator that performs the task required of the robot, The most common

		Bottle			Box		Bucket		Cabinet				Camera				Cart		Chair		
Algorithm	Setting	tr. lid	body	rot. lid	rot. lid	body	handle	body	door	body	door	drawer	lens	button	body	knob	wheel	body	wheel	seat	leg
Mask-	RGB	0.0%	57.4%	69.3%	49.3%	65.7%	2.7%	91.7%	62.0%	94.2%	27.7%	66.4%	26.7%	20.9%	79.0%	4.8%	54.6%	95.6%	25.1%	97.0%	88.3%
RCNN	RGB-D	13.9%	68.3%	67.8%	51.5%	66.5%	1.6%	100.0%	61.7%	93.0%	26.3%	63.0%	26.4%	17.0%	92.6%	8.1%	55.3%	93.9%	23.1%	99.0%	85.2%
PartNet	XYZ	24.5%	47.7%	53.5%	27.6%	46.2%	63.4%	99.7%	20.6%	65.9%	9.8%	35.1%	17.0%	0.0%	51.4%	0.0%	6.2%	71.7%	1.2%	93.0%	86.4%
InsSeg	XYZRGB	5.9%	41.3%	54.8%	24.2%	36.8%	60.7%	98.9%	17.4%	64.3%	5.0%	23.6%	10.5%	0.0%	46.1%	1.0%	9.4%	77.3%	1.9%	95.7%	89.2%

		Chair			Clock		CoffeeMachine						Dishwasher		Dispenser		Display			Door	
Algorithm	Setting	knob	caster	lever	hand	body	button	lid	body	lever	knob	container	rot. Door	body	lid	body	rot. screen	base	button	frame	rot. door
Mask-	RGB	0.0%	2.5%	20.0%	11.4%	61.4%	14.7%	73.4%	65.7%	0.0%	43.0%	100.0%	70.4%	90.0%	74.9%	90.1%	74.4%	34.7%	0.0%	39.4%	40.7%
RCNN	RGB-D	0.0%	3.6%	13.4%	12.5%	68.3%	10.4%	61.4%	67.4%	1.0%	35.6%	98.0%	66.8%	87.8%	73.2%	88.1%	71.3%	33.4%	0.0%	35.7%	54.6%
PartNet	XYZ	0.0%	1.0%	0.0%	0.0%	77.0%	0.0%	43.6%	62.4%	0.0%	0.0%	94.0%	50.5%	67.0%	49.1%	57.6%	66.1%	37.1%	0.0%	49.2%	35.3%
InsSeg	XYZRGB	0.0%	1.0%	0.0%	0.0%	79.4%	0.0%	81.2%	45.8%	0.0%	0.0%	85.1%	58.2%	73.3%	27.4%	39.5%	58.2%	39.1%	0.0%	34.6%	24.6%

		Eyeglasses		Fan		Faucet			FoldingChair		Globe		Kettle		Keyboard		KitchenPot		Knife		
Algorithm	Setting	leg	body	rotor	frame	switch	base	spout	seat	leg	sphere	frame	lid	body	base	key	lid	body	blade	body	blade
Mask-	RGB	51.2%	85.2%	54.4%	67.5%	52.5%	47.9%	99.7%	90.6%	46.1%	98.0%	71.1%	75.2%	99.4%	15.0%	17.5%	99.0%	94.5%	11.7%	88.5%	33.4%
RCNN	RGB-D	49.2%	84.9%	39.4%	67.4%	52.1%	55.8%	98.9%	93.8%	47.2%	96.0%	69.6%	94.1%	100.0%	8.8%	5.1%	100.0%	95.0%	10.0%	77.8%	34.5%
PartNet	XYZ	62.1%	93.8%	50.9%	74.8%	34.4%	55.9%	64.2%	91.2%	79.4%	83.0%	77.6%	71.1%	74.1%	6.8%	1.0%	94.6%	94.4%	3.1%	80.1%	9.4%
InsSeg	XYZRGB	80.6%	92.4%	42.0%	63.5%	29.9%	64.1%	78.0%	86.3%	75.6%	79.0%	82.0%	87.2%	90.7%	4.0%	1.0%	93.5%	95.0%	5.0%	82.7%	10.1%

		Lamp			Laptop		Lighter				Microwave			Mouse			Oven				
Algorithm	Setting	base	rot. bar	head	base	screen	wheel	button	body	rot. lid	door	body	button	button	wheel	body	door	knob	body	tr. tray	button
Mask-	RGB	54.6%	14.6%	64.5%	51.9%	93.1%	35.0%	80.8%	96.8%	97.0%	53.8%	94.0%	0.0%	0.0%	46.5%	98.0%	54.0%	49.9%	86.8%	1.0%	0.0%
RCNN	RGB-D	48.8%	10.8%	69.5%	47.2%	92.8%	57.2%	94.1%	89.2%	92.1%	49.5%	97.1%	0.0%	1.0%	45.3%	95.2%	53.4%	42.3%	93.3%	1.0%	0.0%
PartNet	XYZ	51.8%	8.8%	38.5%	93.0%	97.7%	1.0%	0.0%	77.4%	80.9%	25.9%	45.8%	0.0%	1.0%	0.0%	76.0%	23.1%	0.0%	36.6%	1.0%	0.0%
InsSeg	XYZRGB	50.6%	9.3%	39.7%	89.8%	96.1%	9.5%	61.4%	82.5%	84.9%	24.3%	48.7%	0.0%	1.0%	1.0%	61.1%	26.9%	0.0%	49.1%	0.0%	0.0%

		Pen			Phone		Pliers	Printer		Refrigerator		Remote		Safe				Scissors	Stapler		
Algorithm	Setting	cap	body	button	button	base	leg	button	body	body	door	button	base	knob	button	body	door	leg	body	lid	base
Mask-	RGB	94.1%	91.0%	52.8%	18.4%	51.4%	79.9%	2.8%	87.1%	83.0%	60.7%	35.6%	75.2%	34.1%	0.0%	88.5%	68.5%	34.2%	32.1%	60.2%	84.6%
RCNN	RGB-D	94.1%	96.2%	57.6%	12.8%	50.2%	78.7%	1.5%	72.3%	81.2%	55.0%	25.6%	78.2%	24.5%	0.0%	92.1%	74.6%	57.4%	33.6%	75.0%	90.5%
PartNet	XYZ	67.9%	98.0%	53.0%	1.0%	38.0%	37.9%	0.0%	34.8%	30.0%	16.2%	1.0%	63.2%	0.0%	0.0%	40.5%	30.5%	20.6%	31.7%	49.2%	76.7%
InsSeg	XYZRGB	15.0%	96.2%	25.4%	0.0%	27.0%	46.0%	0.0%	48.5%	40.2%	27.7%	1.0%	75.9%	0.0%	0.0%	60.8%	42.3%	36.4%	28.5%	83.3%	89.5%

		Suitcase					Switch				Table					Toaster				Toilet	
Algorithm	Setting	rot. handle	body	tr. handle	wheel	caster	frame	lever	button	slider	drawer	body	wheel	door	caster	knob	slider	body	button	lever	lid
Mask-	RGB	25.5%	81.7%	74.3%	6.2%	0.0%	85.9%	24.3%	73.6%	60.8%	54.3%	88.0%	3.4%	6.3%	0.0%	40.1%	39.0%	90.1%	5.9%	51.6%	98.3%
RCNN	RGB-D	36.4%	97.3%	70.0%	18.1%	0.0%	74.0%	26.0%	65.8%	22.8%	58.6%	89.9%	1.4%	13.2%	0.0%	40.6%	33.0%	94.1%	4.0%	36.4%	98.0%
PartNet	XYZ	3.7%	53.7%	63.6%	1.4%	0.0%	52.3%	2.3%	4.9%	1.0%	15.7%	71.3%	1.7%	1.0%	0.0%	0.0%	9.9%	79.3%	0.0%	0.0%	69.3%
InsSeg	XYZRGB	4.3%	53.2%	64.5%	2.0%	0.0%	53.5%	1.0%	2.1%	1.7%	16.4%	81.8%	1.3%	2.0%	1.0%	2.6%	20.3%	72.9%	0.0%	0.0%	89.6%

		Toilet				TrashCan					USB		WashingMachine				Window		All
Algorithm	Setting	body	lid	seat	button	pad	lid	body	door	wheel	rotation	body	lid	door	knob	button	body	window	frame	mAP
Mask-	RGB	95.3%	64.3%	61.1%	8.9%	43.4%	68.1%	85.6%	35.9%	73.7%	59.8%	65.7%	71.3%	52.0%	6.8%	4.4%	53.5%	55.9%	12.2%		53.0%
RCNN	RGB-D	91.8%	64.4%	62.5%	3.0%	37.1%	69.7%	84.9%	29.7%	69.3%	74.4%	62.8%	68.6%	41.4%	4.0%	0.0%	73.3%	48.7%	13.4%		52.8%
PartNet	XYZ	83.2%	17.6%	1.4%	0.0%	12.7%	67.1%	57.7%	12.1%	5.5%	30.0%	27.1%	22.2%	22.4%	0.0%	0.0%	30.5%	22.6%	83.5%		36.1%
InsSeg	XYZRGB	86.5%	25.1%	5.2%	0.0%	21.8%	75.4%	73.5%	3.9%	5.0%	23.9%	42.9%	12.2%	14.5%	0.0%	0.0%	22.9%	24.0%	85.2%		37.1%

Table 1: **Movable part segmentation results for all categories**

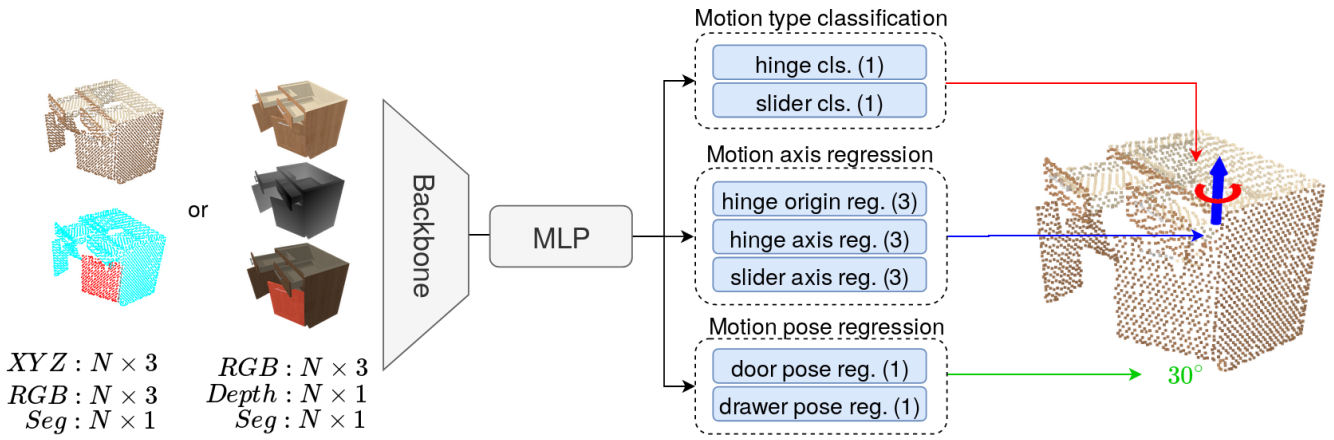

Figure 3: **Vision Tasks.** Show 2 vision task definitions: inputs + outputs.

end-effector is gripper.

- **Inverse Kinematics:** Determine the joint position corresponding to a given end-effector position and orientation [5].
- **Inverse Dynamics:** Determining the joint torques which are needed to generate a given motion. Usualy, the input of inverse dynamics is the output of inverse kinematics or motion planning.

SAPIEN Renderer

- **GLSL** is OpenGL's shading language with describes how the GPU draws visuals.
- **Rasterization** is the process of converting shapes to pixels. It is the pipeline used by most real-time graphics applications.
- **Ray tracing** is a rendering technique by simulating light-rays, reflections, refractions, etc. It can achieve physically accurate images at the cost of rendering time. OptiX is Nvidia's GPU based ray-tracing framework.

References

[1] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016. 2

[2] Seth Hutchinson, Gregory D Hager, and Peter I Corke. A tutorial on visual servo control. *IEEE transactions on robotics and automation*, 12(5):651–670, 1996. 3

[3] Nvidia. PhysX physics engine. https://www.geforce.com/hardware/technology/physx. 3

[4] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. PointNet++: Deep hierarchical feature learning on point sets in a metric space. In *Advances in Neural Information Processing Systems*, pages 5099–5108, 2017. 3

[5] Bruno Siciliano, Lorenzo Sciavicco, Luigi Villani, and Giuseppe Oriolo. *Robotics: modelling, planning and control*. Springer Science & Business Media, 2010. 5