Fixing Malfunctional Objects With Learned Physical Simulation and Functional Prediction

Yining Hong
UCLA

Kaichun Mo
Stanford University

Li Yi
Tsinghua University

Leonidas J. Guibas
Stanford University

Antonio Torralba
MIT

Joshua B. Tenenbaum
MIT BCS, CBMM, CSAIL

Chuang Gan
MIT-IBM Watson AI Lab

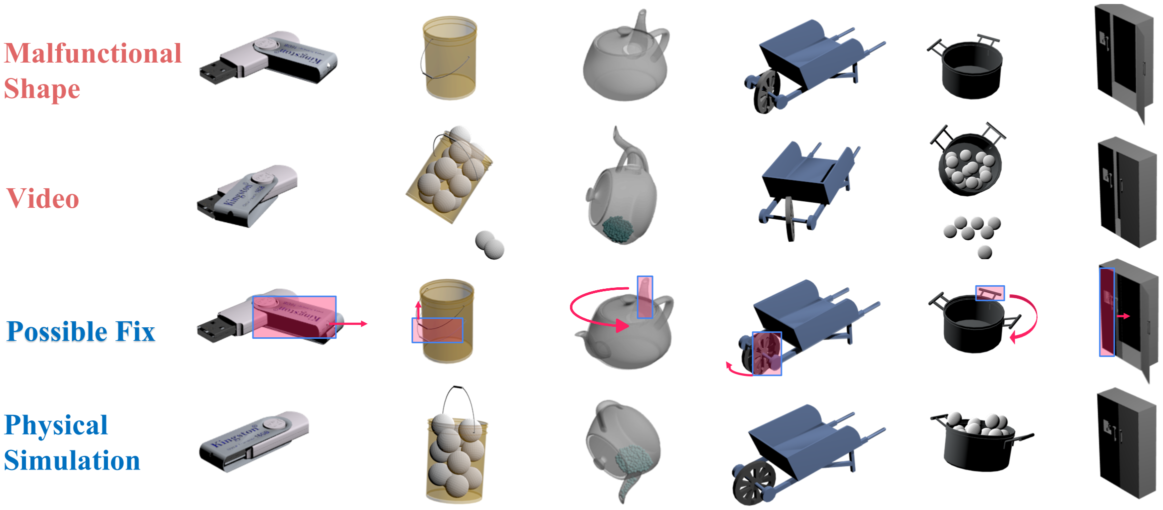

Figure 1. We introduce FixIt, a dataset that requires machines to fix malfunctional objects based on functionality. Each malfunctional object is paired with a video presenting how the object is interacted. Functionality of fixed objects can be evaluated via physical simulation.

Abstract

This paper studies the problem of fixing malfunctional 3D objects. While previous works focus on building passive perception models to learn the functionality from static 3D objects, we argue that functionality is reckoned with respect to the physical interactions between the object and the user. Given a malfunctional object, humans can perform mental simulations to reason about its functionality and figure out how to fix it. Inspired by this, we propose FixIt, *a dataset that contains about 5k poorly-designed 3D physical objects paired with choices to fix them. To mimic humans' mental simulation process, we present FixNet, a novel framework that seamlessly incorporates perception and physical dynamics. Specifically, FixNet consists of a perception module to extract the structured representation from the 3D point cloud, a physical dynamics prediction module to simulate the results of interactions on 3D objects, and a functionality prediction module to evaluate the functionality and choose the correct fix. Experimental results show that our framework outperforms baseline models by a large margin, and can generalize well to objects with similar interaction types. Code and dataset are publicly available*[1].

1. Introduction

What defines a good 3D object shape? Aspects like aesthetics, comfort and fun *etc* play critical roles in shape design. However, these perspectives are all rendered meaningless without the utmost consideration of an object for

[1] http://fixing-malfunctional.csail.mit.edu

everyday use - its functionality. Functionality is a relation between the goal of users' interaction and the behavior of the object. Each part of the object relates its behavior to the functionality of the entire object shape. For example, the third column in Figure 1 shows the notorious "Coffeepot for Masochists" by the French artist Jacques Carelman [4], in which the rotation of the spout affects the functionality of the coffeepot. There have been recent efforts on learning the functionality of 3D object shapes [33–35, 52, 53]. These works usually treat the functionality as a property of the 3D objects, and use *perception systems* to predict the property.

This practice is consistent with the early "ecological theory of perception" which claims that people could simply pick up clues from the world through direct perception and make predictions about functionality [19]. However, a more widely-accepted notion is that to evaluate functionality, we need to involve physics which are sometimes unperceivable [25, 58]. In fact, functionality is reckoned with respect to the *physical interactions* between the object and the user, instead of being a property associated with the shape only. For example, in Figure 1, a person has to interact with the USB to find that the shell fails to protect the chip.

Fixing designs based on functionality is termed as "functional reasoning in design" [68]. Specifically, to fix an object, one has to modify a part and interact with the object to verify whether the fixed object exhibits the desired function [66]. However, in real-life scenarios, it is unrealistic to try out all possible fixes and interact with all the fixed object shapes. Therefore, humans are inclined to perform mental simulation to simulate the effects of interaction [9]. For example, once we have seen how the USB can be interacted, we come up with some fixing ideas and mentally simulate the interaction of the fixed USB. Similar mechanisms have been achieved for machines via dynamics models [2, 5, 45, 46]. These models are able to simulate future states for a given object and interaction. Equipping machines with the ability to fix objects based on physical dynamics and functionality benefits many real-world applications. For example, it helps predict the outcome of interactions on the objects and recommend fixes to malfunctional objects. When 3D models are designed for virtual reality (VR), it helps guarantee that these 3D models function well.

Inspired by the above ideas, we propose a novel task that requires machines to fix malfunctional objects. To study this problem, we have created a new 3D synthetic dataset, FIXIT, which contains approximately 5k synthetic point cloud videos of 3D objects. The point cloud videos present the simulations of how the 3D objects are interacted. Most of the interactions are not successful, indicating that the objects do not function well, while a small set of videos demonstrate successful interactions. We pair each video with five choices indicating how to fix the 3D objects. Only one of the five choices is correct.

As a first attempt at this challenging task, we propose FixNet, a framework that could learn physical dynamics and functional prediction from 3D point cloud videos. Since previous physical dynamics models require full access to particle states and groupings [43], a major challenge to apply them for this new task is how to predict physical dynamics from raw point cloud videos. Our idea is that the point cloud in 3D objects, after going through perception system that provides structured representations and point correspondences, can be compatible with the particles (*i.e.*, small localized objects to which can be ascribed physical properties) in the physical dynamics system. Specifically, FixNet is composed of three modules: the perception module, the physical dynamics prediction module and the functionality prediction module. The perception module has two parts: a) a flow prediction network that takes a point cloud video as inputs, and proposes the flows of the points, which are used as pseudo-labels for training the dynamics module; and b) a segmentation network that takes the point cloud and the predicted flows, and proposes the parts of the objects, which are used for fixing. The point clouds of the parts are modified according to the fixing choices. The dynamics prediction module then takes the fixed point clouds, performs physical simulation and outputs an interacting video. Finally, the last step of the simulated video is fed into the functionality prediction module to evaluate whether the fixed object functions well.

Experiments on the FIXIT dataset suggest that our FixNet outperforms several baseline models by a large margin. Moreover, it can generalize well to novel categories with the same interaction type. Model diagnosis and qualitative examples show that the challenge of FIXIT lies in providing the accurate segmentations for dynamics prediction. Our contributions can be summarized as follows:

- We propose a novel task of fixing 3D object shapes based on functionality that involves 3D perception, physical and functionality reasoning.
- We propose a new dataset, FIXIT, which contains around 5k object shapes across seven categories for fixing.
- We propose a modular framework, FixNet, which incorporates perception, physical dynamics and functional prediction for fixing.
- Experimental results show that our FixNet outperforms baseline models by a large margin.

2. Related Work

Functionality Modeling. The rich space of 3D object shapes, especially man-made ones in our daily lives, results from the diverse functionalities they need to provide for accomplishing various downstream tasks. It is therefore an important yet challenging research topic to study shape functionality [32] and affordance [20, 26] as a highly relevant concept. Previous works have explored learning shape functionality and affordance from human annotations [10, 13, 67], by watching videos or human demonstrations [16, 39, 55, 57, 76], and learning from interaction by humans [21,33–35,38,59] or robot agents [17,52,53,56,61]. Many works [22, 31, 51, 72] have also demonstrated the importance of parts and structures for well-functional shapes. However, these works mostly focus on perceiving, modeling, and generating shapes with functionalities. Our work instead proposes a new problem formulation of diagnosing and fixing malfunctional objects.

Physical Scene Understanding. Physical Reasoning is an important aspect of cognitive reasoning [8, 27–29]. Recently, researchers have focused on using neural networks to predict physical dynamics [2,5–7,12,30,42,43,46,47,54, 71]. Particle-based dynamic systems have been applied to simulate objects of various materials [36, 37, 43, 50, 54, 69]. However, these works usually assume that they have access to all states, clusters and physical properties of the physical systems, which presents a gap between perception and physics. More often, we are presented with raw, irregularly sampled and variant point clouds. There are also dynamic models over latent representations [1, 18, 23, 24, 73]. However, these implicit models fail to capture the complex physical properties and thus do not show good performances in predicting future states. [44] proposes to learn visual priors from images. In contrast, we propose to learn perception from 3D point clouds, and use physical dynamics to fix malfunctional 3D object shapes.

3. The FixIt dataset

Cat.	Shapes	Func.	Success Definition
Fridge	1290	Close	No Collision; No interior exposed
Bucket	624	Lift	Height Change; No Water Out
USB	1096	Sheild	Chip is not exposed
Kettle	213	Pour	Water can be poured out
Cart	654	Move	Move forward without rotation
Pot	751	Lift	Height Change; No Water Out
Box	327	Close	No interior exposed

Table 1. Statistics and characteristics of the object categories covered by the FixIt dataset.

We create a new dataset which contains 4,955 3D object instances represented as point clouds, called FixIt. Each object is composed of various parts that can be modified. The objects are paired with point cloud videos showing how the objects are interacted and the dynamic outcomes. Choices indicating possible fixes to the parts of the objects are represented as Domain-Specific Language (DSL).

3.1. Dataset Design

Object Categories. Our dataset contains 7 object categories: `Refrigerator`, `Bucket`, `USB`, `Kettle`, `Cart`, `KitchenPot`, `Box`. The 3D models are from the PartNet-Mobility [74] dataset. We choose these categories since they either have rich articulated parts to interact with (*e.g.*, `Refrigerator`, `Box`, `USB`) or the physical interactions are complex (*e.g.*, `Bucket`, `Kettle`, `kitchenPot`). We purposely break some of the objects by scaling, translating or rotating the parts to make them malfunctional. In Table 1, we define the functionality of each object category. In Figure 1, we show some exemplar objects in our dataset. For more examples and details about each category, please refer to the supplementary material.

Point Cloud Video Generation. We use PyBullet [2] to simulate the physical interactions for our video dataset, as well as to verify the functionality of the objects. For each object, we use an end effector to interact with the objects (for kitchenpot, we use two because there are two handles). We hard code a pre-defined trajectory of the end effector. We use small balls to replace the water in the buckets, kettles and kitchenpots. After the simulation is done, we check the position and rotation change of each object to evaluate whether the object is functional. We finally extract 10 frames out of all the simulation steps to construct the videos. We use furthest-point-sampling to sample a point cloud of size 2048 of each frame. We extract 16 interacting points between the end effector and the object, serving as an extra input to tell machines how we can interact with the object.

Domain-Specific Language. Each fixing choice is represented as a 4-tuple domain-specific language (Type, Part, Axis, Value). There are three fixing types: “scale”, “translate” and “rotate”; and six axes: “+x”, “-x”, “+y”, “-y”, “+z”, “-z”. The value is uniformly sampled within a range given the object shape. For each part of the object, we use one root point to represent this part and give it an index. A choice refers to the part to be fixed by specifying the index. A choice can also be “functional” indicating the object is already functional and does not need a fix. Figure 2 shows an example of choices in dataset.

3.2. Problem Formulation

For an original point cloud $\mathcal{P}_1$ of the 3D object to be fixed, our framework takes as input a simulated point cloud video $\mathbf{P} = \{\mathcal{P}_1, \mathcal{P}_2, ..., \mathcal{P}_T\}$, where T denotes the number of frames in the video, which is 10 for our dataset. The point cloud in the t-th frame $\mathcal{P}_i$ can be represented as a set of points $\mathcal{P}_i = \{p_{t1}, p_{t2}, ..., p_{tN}\}$ where N equals 2048. We also take in an additional set of interacting points $\mathcal{IP}_t = \{ip_{t1}, ip_{t2}, ..., ip_{tK}\}$ (K = 16) indicating where the end effector operates on the initial point cloud, and $\mathbf{IP} = \{\mathcal{IP}_1, ..., \mathcal{IP}_T\}$. For an original point cloud $\mathcal{P}_1$ of L parts, we define a set of root point indices $\mathcal{R} = \{r_1, r_2, ..., r_L\}$, where each root is the index in $\mathcal{P}_1$ representing the indicator to a part of the object. Our framework also takes as inputs a set of five choices $\mathcal{C} = \{c_1, c_2, ..., c_5\}$ to fix the original

[2] https://pybullet.org/

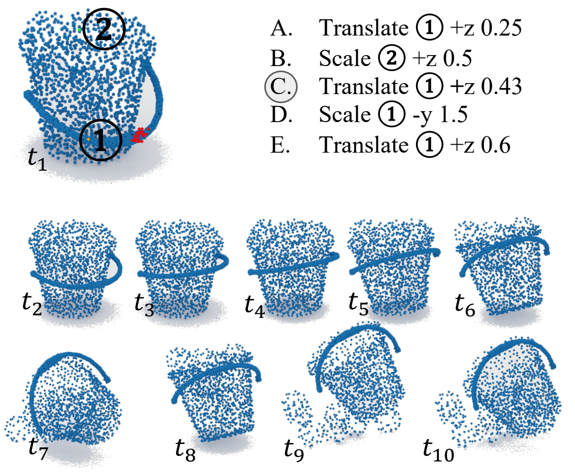

Figure 2. An example of our FIXIT dataset. It has several components: 1) 3D point cloud of the shape to be fixed; 2) a point cloud video showing the interaction of this object; 3) root points representing the parts and part indexes; 4) interacting points (the red points); 5) a set of five choices to fix it. Each choice refers to one of the parts via the part index.

object $\mathcal{P}_1$. Each choice refers to the part to be fixed using one of the root points.

4. FixNet

Inspired by humans' mental simulation process, we aim to design AI models that can perform physical simulation of interactions on objects and evaluate the functionality of the fixed objects. There have been works that could predict the dynamics of physical objects accurately [2, 5, 45, 46]. However, they require full access to the particle representations, point correspondences and groupings, which are often unobtainable in real-world scenarios. The major challenge resides in learning particle-based dynamics models from raw point cloud videos. Our idea to tackle this challenge is to use the perception module to provide structured representations and point correspondences for the physical dynamics prediction module. Therefore, we present FixNet, a framework that seamlessly bridges the gap between 3D point cloud videos and physical dynamics. As shown in Figure 3, our proposed FixNet consists of three modules: a perception module, a physical dynamics prediction module and a functionality prediction module. The perception module consists of two networks: a flow proposal network to extract the flows from the point cloud video of the object, and an instance segmentation network to estimate the parts of the object based on the flow. The physical dynamics prediction module takes a segmented object as an input and learns to approximate the physical simulations of its interactions. Finally, the functionality prediction module takes the outcome of simulation and measures if a modified object is well functional or not.

4.1. 3D Visual Perception

The perception module aims to provide perceptual cues crucial for training the physical dynamics prediction module. It contains two networks: the flow proposal network and the instance segmentation network.

Flow Proposal Network. Since the points in each frame are irregularly sampled, we are unaware of the point correspondences between two frames, thus restricting us from both training the physics simulation module and improving the segmentation network. Therefore, we propose to learn the flow of the points from scratch.

We leverage the scene flow estimation methods [3, 11, 64, 70] to recover flow. Specifically, our flow proposal network is based on FlowNet3D [48], which consists of a PointNet++ to learn embedding, an embedding layer for point mixture and set upconv layers to predict the scene flows.

We re-organize the input point cloud video $\mathbf{P} = \{\mathcal{P}_1, \mathcal{P}_2, ..., \mathcal{P}_T\}$ into pairs of source points $\mathcal{P}_t$ and target points $\mathcal{P}_{t+1}$, where $t = 1, 2, ..., T-1$. The flow proposal network f_T outputs the estimated flow for the source point cloud : $\Delta\tilde{\mathcal{P}}_t = \{(\Delta x_{ti}, \Delta y_{ti}, \Delta z_{ti})\}_{i=1}^N = f_T(\mathcal{P}_t, \mathcal{P}_{t+1})$.

Note that the estimated flow here is not the actual flow, since $(\mathcal{P}_t + \Delta\tilde{\mathcal{P}}_t) \neq \mathcal{P}_{t+1}$. To rectify the flow, we compute a $N \times N \times 3$ disparity matrix $\Delta\mathcal{D}_i = \{d_t^{jk}\}_{j,k=1}^N$ for each pair of points between the t-th frame and $t+1$-th frame, where $d_t^{jk} = p_{(t+1)k} - p_{tj}$. We expand $\Delta\tilde{\mathcal{P}}_t$ $(N \times 1 \times 3)$ to $[\Delta\tilde{\mathcal{P}}_t]$ $(N \times N \times 3)$, which have the same dim as $\Delta\mathcal{D}_t$, and compute the cost matrix $\mathcal{C}_t = ||[\Delta\tilde{\mathcal{P}}_t] - \mathcal{D}_t||^2$. We apply hungarian algorithm [41] on the cost matrix $\mathcal{C}$ to find a bipartite matching $\mathcal{M}_p : \{i \rightarrow \mathcal{M}_p(i) \mid i = 1, 2, \cdots, N\}$ between the source points $\mathcal{P}_t$ and target points $\mathcal{P}_{t+1}$. This is to minimize the overall error between the estimated flow and the actual flow. We then calculate the rectified flow $\Delta\hat{\mathcal{P}}_t$ from point correspondences:$\Delta\hat{\mathcal{P}}_t = \{p_{(t+1)M(i)} - p_{ti}\}_{i=1}^N$.

Instance Segmentation Network. The instance segmentation network proposes part instance segmentations for the succeeding modules. As suggested by [65, 75], the co-segmentation methods which leverage motion flows of articulated objects achieve good performances. Thus, we devise an instance segmentation network that takes perception information and motion information as inputs and outputs the part instances. We first sum up the flows of all frames as the accumulated flow: $\Delta\hat{\mathcal{P}} = \sum_{t=1}^{T-1} \Delta\hat{\mathcal{P}}_t$. We concatenate the point cloud of the original object $\mathcal{P}_1$, the accumulated flow to construct the $N \times 6$ inputs for our instance segmentation network f_I. Our instance segmentation network applies a PointNet++ with multi-scale grouping [60] as backbone network for extracting features and predicting L instance segmentation masks over the input point cloud of size L: $\hat{\mathcal{S}} = \{\hat{s}_l \in [0,1]^N | l = 1, 2, ..., L\} = f_I([\mathcal{P}_1, \Delta\hat{\mathcal{P}}])$. A softmax activation layer is applied such that $\hat{s}_1 + \hat{s}_2 + ... + \hat{s}_L = 1$. We use Hungarian algorithm to find a bipartite match-

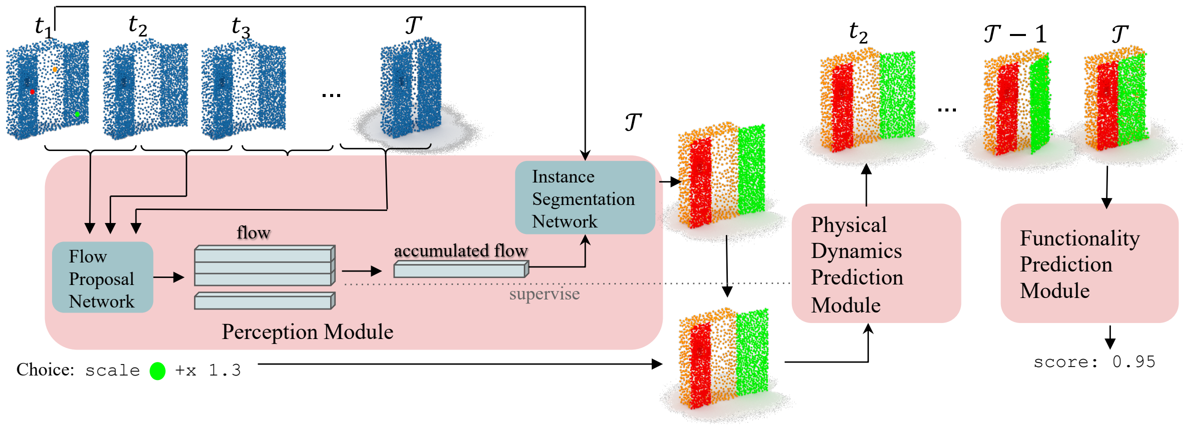

Figure 3. Our proposed FixNet. The point cloud videos are first fed into the flow proposal network which outputs the flows of the points. The flows, and the point cloud of the malfunctional object (which is also the first frame of the video) are input to the segmentation network to produce part instances. Given a choice represented as domain-specific language (DSL), the part referred to in the choice is retrieved via the root point, and modified according to the DSL. The point cloud of the fixed object is then fed into the physical dynamics prediction module, together with the part instances, to predict the future states of the object. The physical dynamics module is trained on the pseudo labels provided by the flows. The last state is input to a functionality prediction module which outputs a functionality score.

ing $\mathcal{M}_s : \{l \to \mathcal{M}_s(l) \mid l = 1, 2, \cdots, L\}$ between the predicted masks $\{\hat{s}_l \mid l = 1, 2, \cdots, L\}$ and the ground-truth masks $\{s_l \mid l = 1, 2, \cdots, L\}$. For the metric of Hungarian algorithm, we use a relaxed IOU [40].

4.2. Physical and Functionality Prediction

Physical Dynamics Prediction Module. We introduce how we utilize the perception prior to simulate physics below. The perception module provides point flows and segmentations that are compatible with particle-based dynamics model [45, 62, 63], and the point clouds of the parts referred in the choices to be fixed. There have been numerous dynamics prediction models proposed recently [2, 5, 46]. We chose DPI-Net [45], since the particle-based physical dynamics system can naturally leverage the points from the perception systems, and the hierarchical modeling paradigm suits the 3D objects of multiple parts.

The interactions within the physics model can be represented as a directed graph, $G = (\langle \mathcal{P}, E \rangle)$, where $\mathcal{P}$ is the set of points, which are called particles in the physics world, and E is a set of relations between the points. The edges E between particles are dynamically generated over time.

Three types of edges are defined in DPI-Net. The first is to establish relationships among neighbors within a predefined distance. The second type is called hierarchical modeling, where the particles are clustered into non-overlapping clusters, and a random particle in the cluster is selected as root, and other particles are leaf nodes. Directed edges include $E_{LeafToRoot}$, $E_{RootToLeaf}$ and $E_{RootToLeaf}$. DPI-Net employs a multi-stage propagation paradigm: propagation among leaf nodes $\phi_{LeafToLeaf}$; from leaf nodes to roots $\phi_{LeafToRoot}$; between roots $\phi_{RootToRoot}$ and root to leaf $\phi_{RootToLeaf}$. We use the segmentation results $\hat{\mathcal{S}}$ as our clusters and use our root points $\mathcal{R}$ as the root particles. The third type of edges is designed for control. We follow the implementation of [45], in which the control inputs are also vertices of the interaction graph, and have directed edges to the points controlled (in real implementation, they choose the points that are close to the positions of the controlling objects as points to be controlled). The dynamics of the interacting points are pre-defined, and only the dynamics of the particles $\mathcal{P}$ are predicted.

We input the point cloud $\mathcal{P}_1$, together with the interacting points $\mathcal{IP}$ for control and the segmentations $\hat{\mathcal{S}}$ for hierarchical modeling. DPI-Net f_P predicts future trajectory of the physical interaction, $\hat{\mathbf{P}} = \{\hat{\mathcal{P}}_2, ..., \hat{\mathcal{P}}_T\}$ step by step given an initial object: $\hat{\mathbf{P}} = f_P(\mathcal{P}_1, \hat{\mathcal{S}}, \mathbf{IP})$.

Functionality Prediction Module. The functionality prediction module finally takes in the last frame output by the physical dynamics prediction module and predicts its functionality score by examining whether the goal of the interaction has been achieved.

4.3. Training and Inference

Training. The flow proposal network is pretrained on the Flyingthings3D dataset (as suggested by [48]) and finetuned on 10% ground-truth flows of videos in the training set. The instance segmentation network is trained on 20% of the ground-truth segmentations of the training set using an IoU loss and a l2,1-norm regularization loss. The physical dynamics prediction module is trained with the pseudo-labels from the flows of the videos predicted by the flow proposal network. The functionality prediction module is trained on the last frame of the simulated video of each choice, and supervised on the correctness of the five choices.

Inference. Given a malfunctional object and its interaction

	Refrigerator	Bucket	USB	Kettle	Cart	KitchenPot	Box	All
DSL-only	24.1	22.5	16.7	18.8	21.4	18.7	20.4	20.6
PointNet++	21.3	23.5	20.1	26.6	24.0	24.4	20.4	22.3
MeteorNet	33.1	37.4	25.5	35.9	27.0	33.8	30.3	30.6
PST-Net	20.8	21.9	36.8	28.1	31.1	26.2	23.5	27.1
P4-Transformer	29.2	41.2	41.0	31.3	31.6	34.2	29.6	34.5
Fix+PointNet++	54.6	52.4	40.1	51.6	49.5	40.0	46.9	47.6
FixNet	**67.4**	**61.0**	**56.2**	**56.3**	**69.4**	**52.4**	**71.4**	**62.3**

Table 2. The accuracies of different models on FIXIT. Our FixNet outperforms all baselines by a large margin.

	Fridge	Bucket	Kettle	USB	Cart	KitchenPot	Box
EPE	0.01	0.09	0.13	0.06	0.03	0.11	0.02
$\mathcal{L}_{IOU}$ (-)	92.1	43.7	47.9	73.2	53.5	56.5	79.8

Table 3. Validation EPE of the flow proposal network, and IOU loss of the instance segmentation network.

video, we first feed the video into the flow proposal network to get the flow. Then we input the point cloud and the predicted flow into the instance segmentation network to get the instance segmentation.

We then try to fix the object. Each choice in our choice set specifies a part index indicating the part to be modified. Retrieving from the root points, we get the root point index, and find its instance from the segmentation. We can then reconstruct the whole point cloud by selecting the points that are assigned the same instance as the root point in the choice. The DSL in each choice can be translated into a transformation matrix and applied on the part point cloud, which together with other parts constitute the fixed point cloud. With the object modified, the interacting points also need to be modified. However, the set of points to be controlled stay the same. We get a position offset as the average position change of the controlled points, and apply the same offset to our interacting points to get the revised interacting points. The revised point cloud, the segmentation and the revised interacting points are fed into the physical dynamics prediction module to output the simulated video. We take the point cloud of the last frame and input it to the functionality prediction module to get the functionality score. The choice with the maximum score is selected.

5. Experiments

5.1. Experimental Setup

Setup. The train/val/test split is approximately 6:1:3. All models select one choice out of the five candidates based on their scores. The evaluation metric is to calculate the percentage of the object instances correctly fixed by selecting the right choice.

Baselines. We implement several baselines for this task.

- **DSL-only** The DSL-only baseline takes only the choices written in domain-specific language (DSL) indicating the object after fixing is functional or not. And functionality scores are predicted based on the DSL features.
- **PointNet++** [60] works on the single frame point cloud, which is used to examine whether the dynamics data in video assists in finding the correct fix.
- **MeteorNet** [49] adds a temporal dimension to PointNet++ to process 4D points and use chain-flow grouping.
- **PSTNet** [15] uses a point spatio-temporal (PST) convolution to represent point cloud sequences.
- **Point 4D Transformer** (P4-Transformer) [14] uses a point 4D convolution to embed the spatio-temporal local structures along with a transformer to capture the appearance and motion information.
- **Fix+PointNet++** Directly inputs the fixed object into PointNet++ to predict functionality.

For the baselines taking 3D point cloud inputs (except Fix+PointNet++), we concatenate the feature of point cloud and DSL to output the functionality score of each fix.

Implementation Details. For a fair comparison, all the point-cloud baselines use the segmentations from the perception module in Section 4. In addition to the 3 dims representing the point positions, we add a mask dim which specifies the points to be fixed and other points, and another dim for specifying interacting points. All the baselines use the same parameters described in the original papers, and are trained for 100 epochs. The training parameters in the individual modules of FixNet are listed in the supplementary.

5.2. Results and Analysis

Main Results. We show the multiple-choice accuracy in Table 2. As we can see, our model outperforms all baselines on point cloud video processing by a large margin. It excels in categories that involve multiple articulated parts, such as refrigerator and box. We also notice that for objects that are more complex in terms of physics and interactions (*e.g.*, buckets, kettles and kitchenpots), the results are lower than objects with more uncomplicated physics in general. However, neural networks such as P4-Transformer and MeteorNet seem to achieve better results in these objects. The reason for this might be that the structures are more fixed for these types than others (*e.g.*, the box can have arbitrary lids, but the bucket only has one handle), thus easier for neural models to memorize. FixNet is superior to Fix+PointNet++, demonstrating that physical dynamics is essential.

To get more insights about our model, we present some intermediate results on the validation set of the flow proposal network and the instance segmentation network, as in

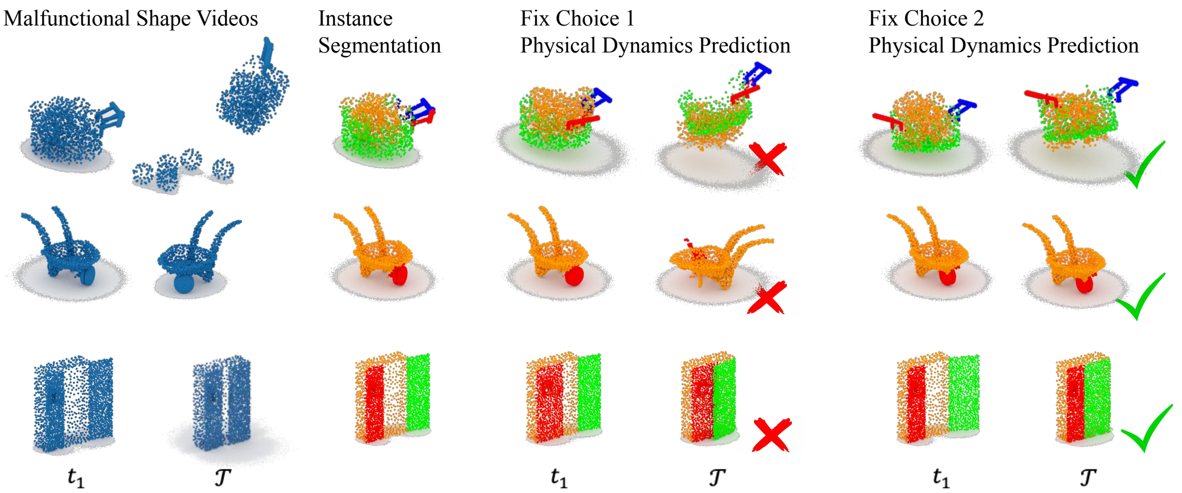
Malfunctional Shape Videos
Instance Segmentation
Fix Choice 1 Physical Dynamics Prediction
Fix Choice 2 Physical Dynamics Prediction
t_1
$\mathcal{T}$
t_1
$\mathcal{T}$
t_1
$\mathcal{T}$

Figure 4. Qualitative examples by FixNet. Red crosses indicate unchosen fixes and green marks represent chosen fixes. As can be seen, our FixNet achieves satisfying segmentation and simulation performances.

Table 3. We can see that objects with more complex physical interactions have larger errors for the flow proposal network, probably because they experience more drastic motion changes. Since the instance segmentation network is based on the flow proposal network, larger errors in the flow result in worse performances of the instance segmentation network. The inaccuracies of the perception module lead to the underperformances of the subsequent modules. How to extract precise structured representations for complex physical objects remains to be solved.

Qualitative Examples. Figure 4 shows per-module visualization results of FixNet. We can see that the perception module achieves satisfying results on segmentation for rigid body parts. However, for the kichenpot full of water, FixNet has trouble separating the water particles and the kichenpot. For the physical dynamics prediction module, FixNet can simulate different dynamics for various fixes and thus distinguishes functional fixed objects from malfunctional objects. Although the simulated dynamics are not perfect (*e.g.*, some of the water leaks out of the pot, and the wheel of the cart rotates too much), it does not prevent the functionality prediction module from predicting the right choice.

5.3. Discussions

Failure Cases and Challenges. In Fig. 5, we show two failure cases. We discuss the limitations of our FixNet and indicate future directions for improvements below. The first weakness lies in the inaccuracy of the perception module. Specifically, DPI-Net requires the particles of different clusters to be non-overlapping. However, this is not always the case when it comes to the segmentations provided by neural networks. When two parts are close enough, overlapping inevitably occurs. Moreover, some parts are inherently embedded in others in articulated objects, making the non-overlapping requirement unrealizable. For example, the shell of the USB in Figure 5 is embedded in its body, making the segmentation extremely hard. Since the hierarchical modeling in DPI-Net forces the particles in an instance to have similar dynamics, one part can be blocked by another static part due to intermediate overlapping particles. Therefore, the USB shell that is embedded in the body fails to rotate, while some USBs with separating shell and body manage to function well. In another case, we show the bucket full of water. It's hard to tell the water apart from the bucket body. We find that a large amount of the water particles are segmented into the body part. Since DPI-Net tends to unify the transformations within an instance, the bucket body might be dragged by the water when it spreads out, making the functionality prediction incorrect. As we can see, the fixed bucket should be lifted vertically, but it leans away. The second weakness is that the physical dynamics predictor fails to simulate articulated parts very well. For example, the handle of the bucket is disjointed. This also happens to the cart wheel in the qualitative examples. How to adjust dynamics model for articulated objects and objects with multiple parts is worth delving into.

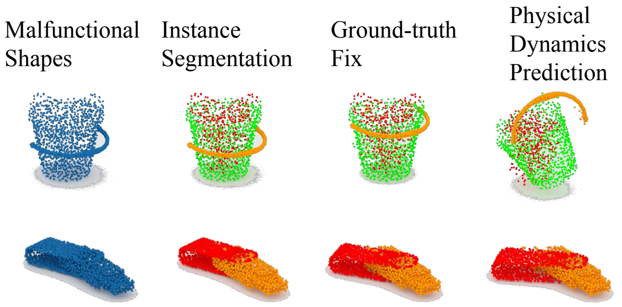

Figure 5. Failure cases by FixNet. For the bucket, the instance segmentation network cannot segment water from the bucket, and the dynamics prediction module disjoints the parts. For the USB, the overlapping of parts result in inaccurate physical simulation.

Model Diagnosis. Benefiting from our modular design, we

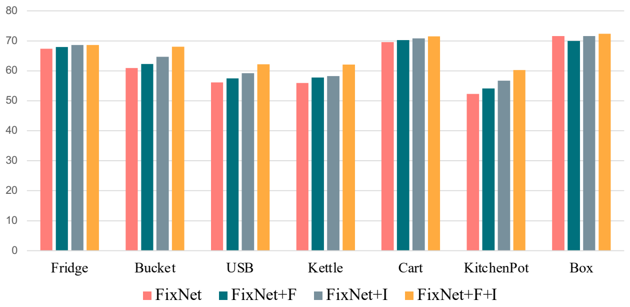

Figure 6. Model Diagnosis. Y-axis represents test accuracy. Adding ground-truth perception can boost the accuracy of FixNet to some extent, especially for categories with low accuracies.

can easily diagnose the model by replacing individual components with the ground truth data from the simulation. In Figure 6, we show the results where we use the ground-truth flows (+F) instead of flows predicted by the flow proposal network, or ground-truth instances (+I) instead of the segmentations provided by the instance segmentation network. We can see that for categories with low accuracies by FixNet, adding the ground-truth flows or segmentations significantly improve the results. However, for categories with high accuracies, adding additional ground truths does not lead to much improvement. This suggests that the underperformances of some categories are probably due to the perception module. We notice that adding ground-truth instances leads to better performances than adding ground-truth flows. Therefore, the major challenge of this dataset might be to predict dynamics based on inaccurate segmentations. This provides insights for future explorations: improving the segmentation performances or designing a dynamics model that can take noisy perception inputs are crucial for the object-fixing task.

5.4. Generalization

In order to evaluate our model's ability to generalize to novel categories, we conduct experiments on three unseen categories using models trained on categories with similar functionalities. Table 4 shows the generalization results. Figure 7 shows some examples of the unseen objects.

Overall, our model achieves satisfying results, outperforming P4-Transformer by a large margin. This might be credited to the generalization ability of particle-based dynamics model. Like the mental simulation of humans, the physical dynamics predictor does not simply memorize patterns, but takes into account the physical interactions among particles. Therefore, when presented with a novel object, it is able to imagine its physical states regardless of what the object looks like. The perception module, however, does not exhibit the same generalization ability. In Figure 7, the segmentations are incorrect for all the three objects. For the first door, the incorrect perception poses great negative impacts on the physics module. For the second door, although the segmentation is also incorrect, the physical dynamics predictor manages to simulate the perfect results. For the USB, the model simulates the first half of the trajectory accurately, but then stops. This might be due to the slightly different interaction ways of USBs and knives.

	Door	Kettle (Revolute Handle)	Knife
Train Category	Fridge	Bucket	USB
P4-Transformer	21.3	24.5	23.0
FixNet	60.4	48.9	36.5

Table 4. Generalization Results. Training categories denote the categories that the models are trained on. Our FixNet shows satisfying accuracies.

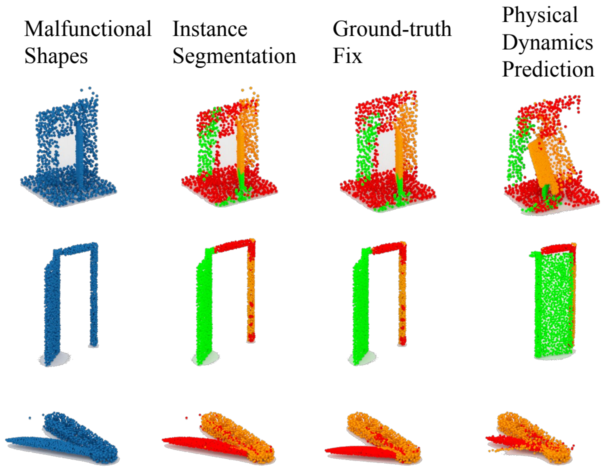

Figure 7. Examples of generalization. For the first door, both segmentation and simulation are incorrect. For the second the door, the segmentation is incorrect but the simulation is right. For the USB, the segmentation is wrong and the simulation is half-right.

6. Conclusions

We study a novel problem of learning to fix malfunctional 3D objects and create a large-scale dataset FixIt to benchmark seven types of object functionality. We design a novel framework FixNet that incorporates perception and physical dynamics to tackle the task. Experiments show that our method outperforms several baseline methods.

Limitations and Future Works. We observe some failure cases when the articulated part is not well-segmented or the dynamic simulation for the articulated part and joint is inaccurate. Future works shall propose better part segmentation and dynamic models.

Acknowledgement. This work was supported by MIT-IBM Watson AI Lab and its member company Nexplore, ONR MURI (N00014-13-1-0333), DARPA Machine Common Sense program, ONR (N00014-18-1-2847), NSF Grant BCS 1921501, Vannevar Bush faculty fellowship, ARL grant W911NF2120104, and gifts from MERL, Autodesk and Snap corporations.

References

[1] Mohammad Babaeizadeh, Chelsea Finn, Dumitru Erhan, Roy H Campbell, and Sergey Levine. Stochastic variational video prediction. *arXiv preprint arXiv:1710.11252*, 2017. 3

[2] Peter W Battaglia, Razvan Pascanu, Matthew Lai, Danilo Rezende, and Koray Kavukcuoglu. Interaction networks for learning about objects, relations and physics. *arXiv preprint arXiv:1612.00222*, 2016. 2, 3, 4, 5

[3] Aseem Behl, Despoina Paschalidou, Simon Donné, and Andreas Geiger. Pointflownet: Learning representations for rigid motion estimation from point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7962–7971, 2019. 4

[4] Carelman. *Catalogue d'objets introuvables, et cependant indispensables aux personnes telles que: acrobates, ajusteurs, amateurs d'art... Yogis, zingueurs et bricoleurs en tous genres...* A. Balland, 1969. 2

[5] Michael B Chang, Tomer Ullman, Antonio Torralba, and Joshua B Tenenbaum. A compositional object-based approach to learning physical dynamics. *arXiv preprint arXiv:1612.00341*, 2016. 2, 3, 4, 5

[6] Sheng Chen, Stephen A Billings, and PM Grant. Non-linear system identification using neural networks. *International journal of control*, 51(6):1191–1214, 1990. 3

[7] Zhenfang Chen, Jiayuan Mao, Jiajun Wu, Kwan-Yee K Wong, Joshua B. Tenenbaum, and Chuang Gan. Grounding physical concepts of objects and events through dynamic visual reasoning. In *International Conference on Learning Representations*, 2021. 3

[8] Zhenfang Chen, Peng Wang, Lin Ma, Kwan-Yee K Wong, and Qi Wu. Cops-ref: A new dataset and task on compositional referring expression comprehension. In *CVPR*, 2020. 3

[9] Bo T Christensen and Christian D Schunn. The role and impact of mental simulation in design. *Applied Cognitive Psychology: The Official Journal of the Society for Applied Research in Memory and Cognition*, 23(3):327–344, 2009. 2

[10] Shengheng Deng, Xun Xu, Chaozheng Wu, Ke Chen, and Kui Jia. 3d affordancenet: A benchmark for visual object affordance understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1778–1787, 2021. 3

[11] Ayush Dewan, Tim Caselitz, Gian Diego Tipaldi, and Wolfram Burgard. Rigid scene flow for 3d lidar scans. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1765–1770. IEEE, 2016. 4

[12] Mingyu Ding, Zhenfang Chen, Tao Du, Ping Luo, Joshua B Tenenbaum, and Chuang Gan. Dynamic visual reasoning by learning differentiable physics models from video and language. In *Advances In Neural Information Processing Systems*, 2021. 3

[13] Thanh-Toan Do, Anh Nguyen, and Ian Reid. Affordancenet: An end-to-end deep learning approach for object affordance detection. In *2018 IEEE international conference on robotics and automation (ICRA)*, pages 5882–5889. IEEE, 2018. 3

[14] Hehe Fan, Yi Yang, and Mohan Kankanhalli. Point 4d transformer networks for spatio-temporal modeling in point cloud videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14204–14213, 2021. 6

[15] Hehe Fan, Xin Yu, Yuhang Ding, Yi Yang, and Mohan Kankanhalli. Pstnet: Point spatio-temporal convolution on point cloud sequences. In *International Conference on Learning Representations*, 2020. 6

[16] Kuan Fang, Te-Lin Wu, Daniel Yang, Silvio Savarese, and Joseph J Lim. Demo2vec: Reasoning object affordances from online videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2139–2147, 2018. 3

[17] Kuan Fang, Yuke Zhu, Animesh Garg, Andrey Kurenkov, Viraj Mehta, Li Fei-Fei, and Silvio Savarese. Learning task-oriented grasping for tool manipulation from simulated self-supervision. *The International Journal of Robotics Research*, 39(2-3):202–216, 2020. 3

[18] Chelsea Finn and Sergey Levine. Deep visual foresight for planning robot motion. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2786–2793. IEEE, 2017. 3

[19] James J Gibson. The theory of affordances. *Hilldale, USA*, 1(2):67–82, 1977. 2

[20] James J Gibson. *The ecological approach to visual perception: classic edition*. Psychology Press, 2014. 2

[21] Helmut Grabner, Juergen Gall, and Luc Van Gool. What makes a chair a chair? In *CVPR 2011*, pages 1529–1536. IEEE, 2011. 3

[22] Yanran Guans, Han Liu, Kun Liu, Kangxue Yin, Ruizhen Hu, Oliver van Kaick, Yan Zhang, Ersin Yumer, Nathan Carr, Radomir Mech, and Hao Zhang. FAME: 3d shape generation via functionality-aware model evolution. *IEEE Trans. on Visualization and Computer Graphics*, 2020. 3

[23] David Ha and Jürgen Schmidhuber. Recurrent world models facilitate policy evolution. *arXiv preprint arXiv:1809.01999*, 2018. 3

[24] Danijar Hafner, Timothy Lillicrap, Ian Fischer, Ruben Villegas, David Ha, Honglak Lee, and James Davidson. Learning latent dynamics for planning from pixels. In *International Conference on Machine Learning*, pages 2555–2565. PMLR, 2019. 3

[25] Rex Hartson. Cognitive, physical, sensory, and functional affordances in interaction design. *Behaviour & information technology*, 22(5):315–338, 2003. 2

[26] Mohammed Hassanin, Salman Khan, and Murat Tahtali. Visual affordance and function understanding: A survey. *arXiv preprint arXiv:1807.06775*, 2018. 2

[27] Yining Hong, Qing Li, Daniel Ciao, Siyuan Huang, and Song-Chun. Zhu. Learning by fixing: Solving math word problems with weak supervision. In *Thirty-Fifth AAAI Conference on Artificial Intelligence*, 2021. 3

[28] Yining Hong, Qing Li, Ran Gong, Daniel Ciao, Siyuan Huang, and Song-Chun. Zhu. Smart: A situation model for algebra story problems via attributed grammar. In *The Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI-21*, 2021. 3

[29] Yining Hong, Qing Li, Song-Chun Zhu, and Siyuan Huang. Vlgrammar: Grounded grammar induction of vision and language, 2021. 3

[30] Yining Hong, Li Yi, Joshua B Tenenbaum, Antonio Torralba, and Chuang Gan. Ptr: A benchmark for part-based conceptual, relational, and physical reasoning. In *Advances In Neural Information Processing Systems*, 2021. 3

[31] Ruizhen Hu, Wenchao Li, Oliver Van Kaick, Ariel Shamir, Hao Zhang, and Hui Huang. Learning to predict part mobility from a single static snapshot. *ACM Transactions on Graphics (TOG)*, 36(6):1–13, 2017. 3

[32] Ruizhen Hu, Manolis Savva, and Oliver van Kaick. Functionality representations and applications for shape analysis. In *Computer Graphics Forum*, volume 37, pages 603–624. Wiley Online Library, 2018. 2

[33] Ruizhen Hu, Oliver van Kaick, Bojian Wu, Hui Huang, Ariel Shamir, and Hao Zhang. Learning how objects function via co-analysis of interactions. *ACM Transactions on Graphics (TOG)*, 35(4):1–13, 2016. 2, 3

[34] Ruizhen Hu, Zihao Yan, Jingwen Zhang, Oliver Van Kaick, Ariel Shamir, Hao Zhang, and Hui Huang. Predictive and generative neural networks for object functionality. *arXiv preprint arXiv:2006.15520*, 2020. 2, 3

[35] Ruizhen Hu, Chenyang Zhu, Oliver van Kaick, Ligang Liu, Ariel Shamir, and Hao Zhang. Interaction context (icon) towards a geometric functionality descriptor. *ACM Transactions on Graphics (TOG)*, 34(4):1–12, 2015. 2, 3

[36] Zijie Huang, Yizhou Sun, and Wei Wang. Learning continuous system dynamics from irregularly-sampled partial observations. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 16177–16187. Curran Associates, Inc., 2020. 3

[37] Michael Janner, Sergey Levine, William T Freeman, Joshua B Tenenbaum, Chelsea Finn, and Jiajun Wu. Reasoning about physical interactions with object-oriented prediction and planning. *arXiv preprint arXiv:1812.10972*, 2018. 3

[38] Vladimir G Kim, Siddhartha Chaudhuri, Leonidas Guibas, and Thomas Funkhouser. Shape2pose: Human-centric shape analysis. *ACM Transactions on Graphics (TOG)*, 33(4):1–12, 2014. 3

[39] Hema S Koppula and Ashutosh Saxena. Physically grounded spatio-temporal object affordances. In *European Conference on Computer Vision*, pages 831–847. Springer, 2014. 3

[40] Philipp Krähenbühl and Vladlen Koltun. Parameter learning and convergent inference for dense random fields. In *International Conference on Machine Learning*, pages 513–521. PMLR, 2013. 5

[41] Harold W Kuhn. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97, 1955. 4

[42] Sizhe Li, Zhiao Huang, Tao Du, Hao Su, Joshua B. Tenenbaum, and Chuang Gan. Contact points discovery for soft-body manipulations with differentiable physics. In *International Conference on Learning Representations*, 2022. 3

[43] Yunzhu Li, Hao He, Jiajun Wu, Dina Katabi, and Antonio Torralba. Learning compositional koopman operators for model-based control. *arXiv preprint arXiv:1910.08264*, 2019. 2, 3

[44] Yunzhu Li, Toru Lin, Kexin Yi, Daniel Bear, Daniel Yamins, Jiajun Wu, Joshua Tenenbaum, and Antonio Torralba. Visual grounding of learned physical models. In *International conference on machine learning*, pages 5927–5936. PMLR, 2020. 3

[45] Yunzhu Li, Jiajun Wu, Russ Tedrake, Joshua B Tenenbaum, and Antonio Torralba. Learning particle dynamics for manipulating rigid bodies, deformable objects, and fluids. In *ICLR*, 2019. 2, 4, 5

[46] Yunzhu Li, Jiajun Wu, Jun-Yan Zhu, Joshua B Tenenbaum, Antonio Torralba, and Russ Tedrake. Propagation networks for model-based control under partial observation. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 1205–1211. IEEE, 2019. 2, 3, 4, 5

[47] Xingyu Lin, Zhiao Huang, Yunzhu Li, David Held, Joshua B. Tenenbaum, and Chuang Gan. Diffskill: Skill abstraction from differentiable physics for deformable object manipulations with tools. In *International Conference on Learning Representations*, 2022. 3

[48] Xingyu Liu, Charles R Qi, and Leonidas J Guibas. Flownet3d: Learning scene flow in 3d point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 529–537, 2019. 4, 5

[49] Xingyu Liu, Mengyuan Yan, and Jeannette Bohg. Meteornet: Deep learning on dynamic 3d point cloud sequences. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9246–9255, 2019. 6

[50] Pingchuan Ma, Tao Du, John Z Zhang, Kui Wu, Andrew Spielberg, Robert K Katzschmann, and Wojciech Matusik. Diffaqua: A differentiable computational design pipeline for soft underwater swimmers with shape interpolation. *ACM Transactions on Graphics (TOG)*, 40(4):132, 2021. 3

[51] Mariem Mezghanni, Malika Boulkenafed, Andre Lieutier, and Maks Ovsjanikov. Physically-aware generative network for 3d shape modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9330–9341, 2021. 3

[52] Kaichun Mo, Leonidas J. Guibas, Mustafa Mukadam, Abhinav Gupta, and Shubham Tulsiani. Where2act: From pixels to actions for articulated 3d objects. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 6813–6823, October 2021. 2, 3

[53] Kaichun Mo, Yuzhe Qin, Fanbo Xiang, Hao Su, and Leonidas Guibas. O2O-Afford: Annotation-free large-scale object-object affordance learning. In *Conference on Robot Learning (CoRL)*, 2021. 2, 3

[54] Damian Mrowca, Chengxu Zhuang, Elias Wang, Nick Haber, Li Fei-Fei, Joshua B Tenenbaum, and Daniel LK Yamins. Flexible neural representation for physics prediction. *arXiv preprint arXiv:1806.08047*, 2018. 3

[55] Tushar Nagarajan, Christoph Feichtenhofer, and Kristen Grauman. Grounded human-object interaction hotspots from video. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8688–8697, 2019. 3

[56] Tushar Nagarajan and Kristen Grauman. Learning affordance landscapes for interaction exploration in 3d environments. In *NeurIPS*, 2020. 3

[57] Tushar Nagarajan, Yanghao Li, Christoph Feichtenhofer, and Kristen Grauman. Ego-topo: Environment affordances from egocentric video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 163–172, 2020. 3

[58] Don Norman. *The design of everyday things: Revised and expanded edition*. Basic books, 2013. 2

[59] Sören Pirk, Vojtech Krs, Kaimo Hu, Suren Deepak Rajasekaran, Hao Kang, Yusuke Yoshiyasu, Bedrich Benes, and Leonidas J Guibas. Understanding and exploiting object interaction landscapes. *ACM Transactions on Graphics (TOG)*, 36(3):1–14, 2017. 3

[60] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in Neural Information Processing Systems*, 30, 2017. 4, 6

[61] Zengyi Qin, Kuan Fang, Yuke Zhu, Li Fei-Fei, and Silvio Savarese. Keto: Learning keypoint representations for tool manipulation. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 7278–7285. IEEE, 2020. 3

[62] Alvaro Sanchez-Gonzalez, Jonathan Godwin, Tobias Pfaff, Rex Ying, Jure Leskovec, and Peter Battaglia. Learning to simulate complex physics with graph networks. In *International Conference on Machine Learning*, pages 8459–8468. PMLR, 2020. 5

[63] Alvaro Sanchez-Gonzalez, Nicolas Heess, Jost Tobias Springenberg, Josh Merel, Martin Riedmiller, Raia Hadsell, and Peter Battaglia. Graph networks as learnable physics engines for inference and control. In *International Conference on Machine Learning*, pages 4470–4479. PMLR, 2018. 5

[64] Lin Shao, Parth Shah, Vikranth Dwaracherla, and Jeannette Bohg. Motion-based object segmentation based on dense rgb-d scene flow. *IEEE Robotics and Automation Letters*, 3(4):3797–3804, 2018. 4

[65] Yahao Shi, Xinyu Cao, and Bin Zhou. Self-supervised learning of part mobility from point cloud sequence. In *Computer Graphics Forum*. Wiley Online Library, 2021. 4

[66] Diomidis H Stamatis. *Failure mode and effect analysis: FMEA from theory to execution*. Quality Press, 2003. 2

[67] Chao Tang, Jingwen Yu, Weinan Chen, and Hong Zhang. Interactive affordance learning through manipulation relationship graph construction. *arXiv preprint arXiv:2110.14137*, 2021. 3

[68] Yasushi Umeda and Tetsuo Tomiyama. Functional reasoning in design. *IEEE expert*, 12(2):42–48, 1997. 2

[69] Benjamin Ummenhofer, Lukas Prantl, Nils Thuerey, and Vladlen Koltun. Lagrangian fluid simulation with continuous convolutions. In *International Conference on Learning Representations*, 2019. 3

[70] Arash K Ushani, Ryan W Wolcott, Jeffrey M Walls, and Ryan M Eustice. A learning approach for real-time temporal scene flow estimation from lidar data. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5666–5673. IEEE, 2017. 4

[71] Eric Wan, Antonio Baptista, Magnus Carlsson, Richard Kiebutz, Yinglong Zhang, and Alexander Bogdanov. Model predictive neural control of a high-fidelity helicopter model. In *AIAA Guidance, Navigation, and Control Conference and Exhibit*, page 4164, 2001. 3

[72] Xiaogang Wang, Bin Zhou, Yahao Shi, Xiaowu Chen, Qinping Zhao, and Kai Xu. Shape2motion: Joint analysis of motion parts and attributes from 3d shapes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8876–8884, 2019. 3

[73] Jiajun Wu, Erika Lu, Pushmeet Kohli, Bill Freeman, and Josh Tenenbaum. Learning to see physics via visual deanimation. *Advances in Neural Information Processing Systems*, 30:153–164, 2017. 3

[74] Fanbo Xiang, Yuzhe Qin, Kaichun Mo, Yikuan Xia, Hao Zhu, Fangchen Liu, Minghua Liu, Hanxiao Jiang, Yifu Yuan, He Wang, et al. Sapien: A simulated part-based interactive environment. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11097–11107, 2020. 3

[75] Li Yi, Haibin Huang, Difan Liu, Evangelos Kalogerakis, Hao Su, and Leonidas Guibas. Deep part induction from articulated object pairs. *arXiv preprint arXiv:1809.07417*, 2018. 4

[76] Yixin Zhu, Yibiao Zhao, and Song Chun Zhu. Understanding tools: Task-oriented object modeling, learning and recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2855–2864, 2015. 3