OBJECTFOLDER: A Dataset of Objects with Implicit Visual, Auditory, and Tactile Representations (Supplementary Materials)

Ruohan Gao **Yen-Yu Chang*** **Shivani Mall*** **Li Fei-Fei** **Jiajun Wu**

Stanford University

The supplementary materials for [1] consist of:

A. Supplementary video.
B. Dataset details.
C. Details for vision, audio, and touch simulations.
D. User study on acoustic fidelity.
E. Implementation details for implicit neural representation networks.
F. More results on Object Files vs. ground-truth simulations.
G. Single image 3D shape reconstruction on real-world images.
H. Implicit vs. explicit representation for grasp stability prediction.
I. Experiment setup details for the four benchmark tasks.
J. Object manipulation using OBJECTFOLDER.

A Supplementary Video

In the supplementary video, we show 1) the motivation and goal of our dataset; 2) visualization of the visual appearance for all 100 objects obtained from *Object Files* under different camera viewpoints or lighting conditions; 3) examples of the auditory data (impact sounds) obtained from *Object Files* for some sample objects; 4) examples of the tactile images obtained from *Object Files* when the DIGIT sensor touches different surface locations of some sample objects; 5) robotic grasping demos.

B Dataset Details

OBJECTFOLDER contains 100 high quality 3D objects from online repositories. See Fig. 1 for the visualization of all 100 objects in the dataset. Fig. 2a shows the category distribution of the objects, and we use categories similar to the YCB dataset [2]. The categories of objects include 1) kitchen items (e.g., bowl, mug, plate), shape items (e.g., dice, barrel, cup), furniture items (e.g., chair, cabinet, table), task items (e.g., nine-hole peg test, toy airplane), tool items (e.g., wrench, clamp, marker), appliance/electronics (e.g., lantern, television, toaster), and other household items (e.g., bell, medicine bottle). The 100 objects are all of approximately homogeneous material property, and the material types include ceramic, glass, wood, plastic, iron, polycarbonate, and steel. See Fig. 2b for the material type distribution of the objects. The objects in OBJECTFOLDER are also of diverse scale (length of the longest side of the smallest cube that the object can fit in). Fig. 2c shows the distribution of the scale of objects in meter. Both the scale and material type of the objects are used in modal analysis for realistic audio simulation.

C Details for Vision, Audio, and Touch Simulations

Vision: We use Blender's Cycles path tracer [3] to render images. For each object, we first normalize it into a unit cube and use a point light source at a random location on a unit sphere with radiance

*indicates equal contribution.

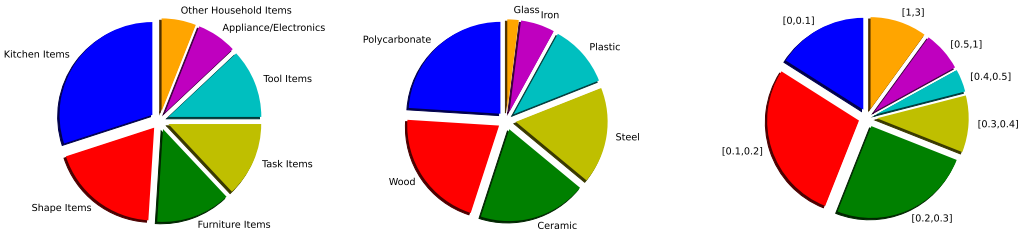
5th Conference on Robot Learning (CoRL 2021), London, UK.

Figure 1: Visualization of all 100 objects in OBJECTFOLDER. Our objects are diverse in categories and material types, and they have realistic visual textures.

of (1, 1, 1). We then render images of the object on a white background from camera viewpoints randomly sampled on a full sphere with a radius of 2.5. We render 500 images each for training, validation, and testing.

Audio: We use linear modal analysis for physics-based rigid-body sound synthesis and simulate the vibration modes excited by unit forces for all vertexes. We first convert the original polygon mesh of each object to a volumetric hexahedron mesh of resolution $32 \times 32 \times 32$. With the annotated material type and scale of the object, we use finite element method (FEM) to perform modal analysis. We use the same set of material parameters as used in [4]. As shown in Table 1, each material type is mapped to values of density, Young's Modulus, Posisson ratio, and Rayleigh damping parameters. We mask the frequency range of the modes signals to be [20HZ, 20KHz], which is the human audible range. We use a sampling rate of 16KHz and 2 seconds in length for each modes signal. Similar to [5], we directly use the obtained modes signal as the simulated impact sound.

Touch: We use TACTO [6], a state-of-the-art vision-based touch simulator, for touch simulation. To obtain touch simulations, we place each object at rest and move the DIGIT sensor to touch each object vertex in the vertex normal direction. We specify the amount of force DIGIT's base joint (motor) should move by for DIGIT (end-effector) to reach a desired pose and orientation, and constrain the force to range within a small threshold such that the rendered tactile images do not



(a) Category (b) Material type (c) Scale

Figure 2: Statistics of categories, materials types, and scale for objects in OBJECTFOLDER.

Material Type	ρ	E	v	α	β
Ceramic	2.70×10^3	7.20×10^{10}	0.19	6	1×10^{-7}
Glass	2.60×10^3	6.20×10^{10}	0.20	1	1×10^{-7}
Wood	7.50×10^2	1.10×10^{10}	0.25	60	2×10^{-6}
Plastic	1.07×10^3	1.40×10^9	0.35	30	1×10^{-6}
Iron	8.00×10^3	2.10×10^{11}	0.28	5	1×10^{-7}
Polycarbonate	1.19×10^3	2.40×10^9	0.37	0.5	4×10^{-7}
Steel	7.85×10^3	2.00×10^{11}	0.29	5	3×10^{-8}

Table 1: Material parameters for audio simulation. $\rho, E, v, \alpha, \beta$ denote density, Young's Modulus, Posisson ratio, and Rayleigh damping parameters, respectively. All parameters are in SI units.

vary significantly with different forces. Controlling the exact amount of force at the desired touch position is non-trivial and may require operational space control, which we leave as future work.

D User Study on Acoustic Fidelity

Our physics-based rigid-body sound synthesis pipeline realistically models the impact sound of every vertex on the object based on its shape, size, material type, external force, and the contact location. To further demonstrate the realism of our simulated audio, we perform a user study on acoustic fidelity that asks participants to distinguish between the real audio and simulated audios. For real audio, We use recordings of striking an everyday object with a tool (a ceramic mug, a steel bowl, a wood napkin holder, a polycarbonate tumbler cup). For the paired simulated audio, we use an impact sound rendered for an object in OBJECTFOLDER of the same material type and category. We recruit 10 volunteers to listen to 16 pairs of audios (one simulated and one real), and ask each listener to make a binary decision on which one sounds more realistic. Results show that audios obtained by querying our Object Files are preferred by 42% of the total responses, comparing closely to real-world audio recordings.

E Implementation Details for Implicit Neural Representation Networks

VisionNet: We follow the prior work on object-centric neural radiance fields [7] to represent each of our object as a 7D object-centric neural scattering function. We reproduce the same settings as their "Furniture-Single" setup using PyTorch. We use an eight-layer MLP with 256 channels to predict the density value σ, which is view invariant; and a four-layer MLP with 128 channels to predict the fraction of the incoming light that is scattered in the outgoing direction $\rho = (\rho_r, \rho_g, \rho_b)$. We use a batch size of 1024 rays, and sample $N_c = 64$ coarse samples and $N_f = 128$ fine samples per ray. We use the Adam optimizer with a learning rate of 0.0005, $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\epsilon = 10^{-7}$. See [7] for more details.

AudioNet: For each force direction, we train a separate branch to encode the corresponding modes signals at all vertexes. Each branch is an eight-layer MLP with 256 channels. For each branch, we obtain the audio spectrogram for the corresponding modes signal with window length of 25 ms, hop length of 10 ms, and FFT size of 512. The dimension of the complex spectrogram we obtain is $257 \times 201 \times 2$, where each time-frequency bin contains the real and imaginary part of the

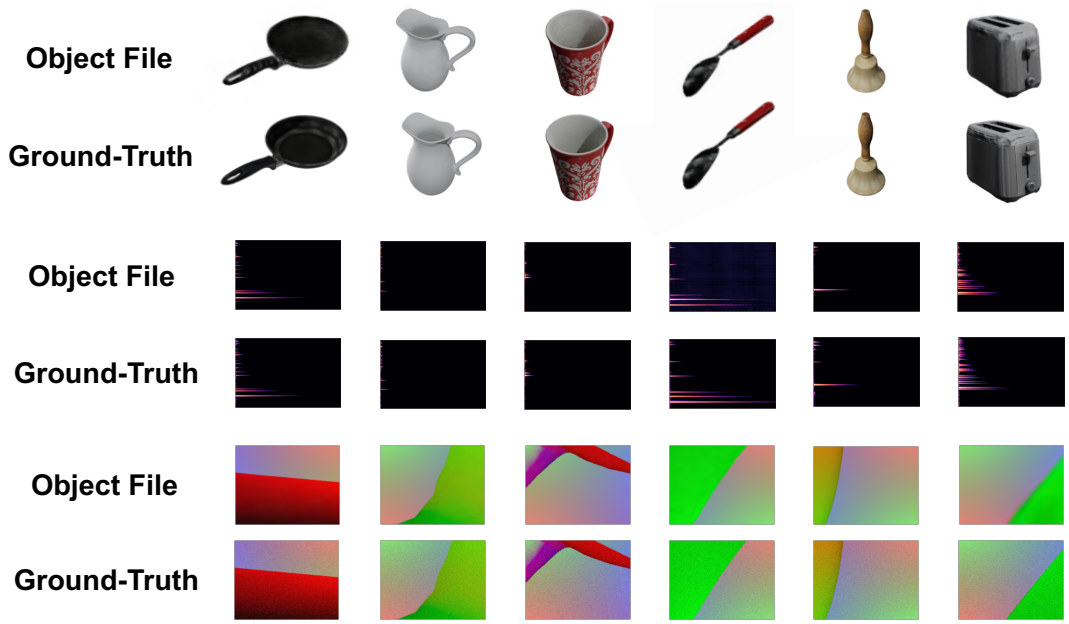


Figure 3: Comparison of the visual, auditory, and tactile data obtained from Object Files and the corresponding ground-truth simulations for six example objects. Our implicit neural representations accurately encode the multisensory data for each object.

corresponding complex spectrogram value. For each object, we normalize the complex spectrogram value to $[0, 1]$ as the prediction target. We use the same positional encoding scheme as in [8].

TouchNet: For each surface location of the polygon mesh of the object, we obtain a RGB tactile image of dimension $160 \times 120 \times 3$ that captures the local geometry information. Similarly, TouchNet is an eight-layer MLP with 256 channels that takes the spatial coordinate (x, y, z) of the vertex and the spatial location (w, h) in the tactile image as input. We normalize the three channels of the target tactile RGB image values to $[-1, 1]$ as the prediction target. The other settings are the same as AudioNet.

F More Results on Object Files vs. Ground-Truth Simulations

In Fig. 4 in the main paper, we have shown some examples of the visual, auditory, and tactile sensory data obtained from *Object Files* compared with the ground-truth simulations. Fig. 3 shows more examples for six different objects. We can see that the rendered images, impact sounds, and touch readings well match the ground-truth simulations, demonstrating the accuracy of our implicit neural representations to encode the multisensory data for the objects.

The results above show the accuracy of the implicit representations qualitatively. Next, we perform an additional experimental analysis to compare the visual, auditory, and tactile data obtained from Object Files with the ground-truth simulations quantitatively: 1) For vision, we render 500 testing images under different camera viewpoints and lighting conditions for each of the 100 objects and compute the average PSNR with respect to the ground-truth simulations; 2) For audio, we obtain 500 impact sounds for each object and compute the average PSNR between the audio spectrograms encoded by the implicit representation networks and the ground-truth audio simulations; 3) For touch, we render 500 tactile images at different surface locations of each object and and compute the average PSNR with respect to the ground-truth tactile readings. For each sensory modality, we compare with a baseline where the ground-truth simulation is corrupted by very minor white Gaussian noise (AWGN, with mean=0 and variance=0.05) as a reference. Table 2 shows the results. We can see that the sensory data encoded by our implicit representation networks achieves very high PSNR values, and compares favorably against the baseline where negligible noise is present.

	Implicit vs. Ground-Truth	AWGN vs. Ground-Truth
Vision	39.8	32.0
Audio	53.6	26.2
Touch	29.4	24.2

Table 2: Quantitative comparison on the accuracy of the implicit representations. We report peak signal-to-noise ratio (PSNR) (in dB, higher is better).

G Single Image 3D Shape Reconstruction on Real-World Images

In Sec. 4.3 in the main paper, we have shown some results on applying our IMAGE2MESH model trained on objects in OBJECTFOLDER to real-world images. Fig. 4 shows more results. Only trained on simulated data of the 100 objects in our dataset, our model generalizes reasonably well to the real-world images of these objects, demonstrating the realism of the objects in our dataset.

Figure 4: 3D shape reconstruction using the model trained on OBJECTFOLDER on real-world images. The last column shows a typical failure case.

H Implicit vs. Explicit Representation for Grasp Stability Prediction

In Sec. 4.4 in the main paper, we have shown how we combine vision and touch for prediction of grasp stability. For touch data, we obtain RGB tactile images from the corresponding Object File implicit representation network. To demonstrate the effectiveness of the implicit representations, we perform an experiment that directly uses the explicit tactile representations (ground-truth tactile simulations) as input for the grasp stability prediction experiment. All the other settings are the same and our goal here is to do a direct comparison of implicit and explicit representations in an apples-to-apples manner.

Figure 5 shows the results. We can see that Object Files in the form of compact implicit representations achieves similar results as using the explicit tactile representations. This shows that our implicit representations accurately encodes the tactile sensory data of the objects while also having the advantage of being flexible to use and easy to share.

I Experiment Setup Details for the Four Benchmark Tasks

I.1 Multisensory Instance Recognition

To perform multisensory instance recognition, for each object we generate 500 images from different camera viewpoints and lighting conditions, 500 impact sounds with random forces for different vertexes, and 500 tactile RGB images at random surface locations from its corresponding *Object File*, respectively. For each sensory modality, we use 80% of the data for training, 10% for validation, and 10% for testing. For the vision modality, we use the ImageNet pre-trained ResNet-18 [9] network that takes an RGB image of the object as input and predicts the instance label. We use images of resolution 256×256 and perform random cropping to obtain images of resolution 224×224 as input to the network. After the final pooling layer, we use a fully-connected layer to map the feature vector of dimension 512 to the label space of 100 categories. For the audio modality, we also use ResNet-18 except that we change the first layer to take a one-channel magnitude spectrogram of dimension 257×201 as input. The other settings are the same as the vision modality. For touch, the settings are similar to vision and audio except that a local tactile RGB image of dimension $160 \times 120 \times 3$ is used as input instead.

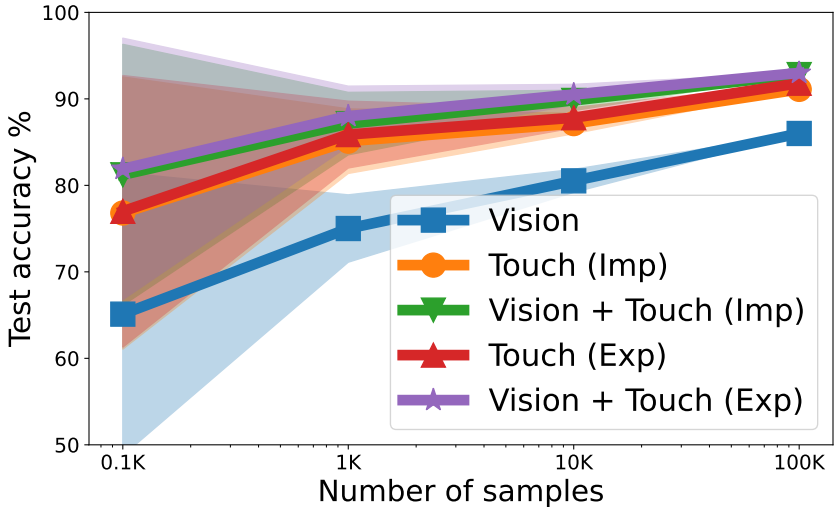


Figure 5: Grasp stability prediction using implicit and explicit representations.

I.2 Cross-Sensory Retrieval

We design a two-stream network with each stream taking one sensory modality of the same object as input. We use the same ResNet-18 network as in the multisensory instance recognition task to extract vision, audio, and touch features. The two modalities have disjoint pathways to extract the modality-specific feature of dimension 512. The extracted feature from both streams then passes through the modality-agnostic network, which consists of two fully-connected (fc) layers. The first fc layer maps features from both streams to a cross-sensory embedding space with feature dimension of 128. After the Rectified Linear Unit (ReLU), the second fc layer maps the feature to the label space for training with cross-entropy classification loss.

To map the features of both modalities from the last hidden state to the cross-sensory embedding space, we use the following triplet loss taking cross-sensory learning with vision and audio as an example:

$$L_t(\mathbf{v}, \mathbf{a}^+, \mathbf{v}^-) = \max\{0, D(\mathbf{v}, \mathbf{a}^+) - D(\mathbf{v}, \mathbf{v}^-) + \mathtt{m}\}, \tag{1}$$

where $\mathbf{v}$ is the anchor visual feature of the object and $\mathbf{a}^+$ is the audio feature for the same object. $\mathbf{v}^-$ represents the visual feature of a different object, which serves as the negative feature for embedding learning. $D(\cdot, \cdot)$ represents the cosine distance of the two embeddings, and $\mathtt{m}$ represents the margin between the two distances. This loss forces the network to learn cross-sensory vision-audio embeddings such that the distance between the embeddings of the same object should be smaller than that between different objects, by a margin $\mathtt{m}$.

We either sample an image or audio from another object as the negative example, and pull the embeddings of the same object to be closer in the cross-sensory embedding space and push the embeddings from different objects apart. Therefore, the final cross-sensory learning loss we use is as follows:

$$L_{\textit{cross-sensory}} = L_t(\mathbf{m_1}^{\mathcal{A}}, \mathbf{m_2}^{\mathcal{A}}, \mathbf{m_1}^{\mathcal{B}}) + L_t(\mathbf{m_2}^{\mathcal{A}}, \mathbf{m_1}^{\mathcal{A}}, \mathbf{m_2}^{\mathcal{B}}), \tag{2}$$

where $\mathbf{m_1}$, $\mathbf{m_2}$ represent two different modalities, and $\mathcal{A}$, $\mathcal{B}$ represent two different objects. The full model is trained jointly with both the cross-sensory loss and the cross-entropy classification loss. The training/validation/testing splits are the same as in the multisensory instance recognition experiments.

I.3 Audio-Visual 3D Reconstruction

We leverage Occupancy Network [10] as a testbed for this task and adapt its architecture design for audio-visual 3D shape reconstruction (IMAGE+AUDIO2MESH). The image encoder and the audio encode are the same as above, each extracting a visual or an audio feature vector of dimension 512.

We concatenate the vision and the audio features to obtain an audio-visual feature vector of dimension 1,024, which is then passed through the fusion layer (two fully-connected layers) that projects the feature map to a 256-dimensional embedding. The occupancy network processes input points (x, y, z) through multiple fully-connected ResNet blocks, and Conditional Batch-Normalization (CBN) is used to condition the network on the audio-visual feature vector extracted from the object. The output of the network is finally projected to one dimension using a fully-connected layer followed by a Sigmoid layer. Finally, we can obtain the occupancy probabilities that represent the 3D shape of the object implicitly. The IMAGE2MESH and the AUDIO2MESH variants are similar except that the fusion layer directly projects the 512 single-modality feature to the 256-dimensional embedding. See [10] for details.

I.4 Robotic Grasping with Vision and Touch

Grasp Stability Prediction: The goal of robotic grasping is to use a robotic arm to grasp and hold an object between the robot's left and right fingers. The complete steps for a successful grasp is as follows: the robot arm first reaches the object and attempts to grasp it by closing its two fingers, then lifts the grasped object and holds the object in the air, and finally puts the object back and releases its two fingers. Specifically, we use the WSG-50 parallel-jaw gripper with two DIGIT sensors mounted on its left and right fingers for the grasping experiments.

Out general setup for grasp stability prediction is similar to that in [6] except that we obtain tactile RGB images from OBJECT FILES—our implicit neural representation networks that encode the multisensory data. We determine the contact positions between the object and the end effector loaded with the DIGIT sensors, and then apply translation and rotation to transform the contact point in the world coordinates system to the local coordinates system of the object. Next we find the closest vertex on the object mesh associated with the contact position and query TouchNet to obtain the corresponding tactile RGB images.

We use 10 objects from OBJECTFOLDER for the grasp stability prediction experiments: 5-HTP Medicine Bottle, Sugar Sprinkles, LTyrosine Medicine Bottle, 2 Slice Toaster, Marvel Avengers Toy, Cajun Jerky Tonic Bottle, Wood Block, Dice, Baby Stacking Cups, and School Bell. We perform grasp data collection with random positions and orientations using these 10 objects and obtain 10K data samples for each object. We collect both the vision and touch data: For touch data, we obtain the RGB tactile images at the grasp contact points from the corresponding Object File; For vision data, we obtain visual images from an externally mounted camera's viewpoint at the grasping moment. We label the data as either Success or Failure based on the grasp outcome.

After data collection, we train binary classifiers using ResNet-18 network under different settings and with different dataset sizes (0.1K, 1K, 10K, 100K) to predict the grasp stability. For the setting with both vision and touch, we use a two-stream network and concatenate the feature vectors produced by each stream. The concatenated feature vector is passed through two fully-connected layers to predict whether the grasp is going to be successful or not.

Policy Learning for Touch-based Robotic Grasping: We simulate robotic grasping of the "Cajun Jerky Tonic Bottle" object with an WSG-50 parallel jaw gripper using Bullet [11]. At the start of each episode, we randomize the object position and rotation. The episodes are of fixed number of time steps ($T = 2$). Namely, the robot makes the attempt to grasp the object by closing its two fingers after 2 time stamps and then the episode ends. We use binary reward at the last time step of the episode with $r = 1$ for a successful grasp and $r = 0$ for a failed grasp. The robot arm moves via position control of the vertically-oriented gripper. We use continuous actions given by $(dx, dy, dz, d\phi)$, where dx, dy, dz are Cartesian displacements added to the position of the end effector in world coordinates and $d\phi$ is the rotation of the end effector around the z axis added to the orientation of the gripper. We use trust region policy optimization (TRPO) [12], which is a policy gradient method that uses conjugate gradient optimization with a KL constraint. We set the damping coefficient $\alpha = 0.01$, discount factor $\gamma = 0.95$, Generalized Advantage Estimation (GAE) $\lambda = 0.97$, and the KL-divergence bound $\delta = 0.01$.

The policy network we use is a MLP with two hidden layers each of 256 hidden units. We concatenate the feature vectors extracted from the tactile RGB images for both the left and right fingers, and use it as input to the policy network. The network directly outputs the 4D displacement vector $(dx, dy, dz, d\phi)$, which represents the action to take. We train our policy over a total of 100K episodes and update the policy network every 2,000 steps. We evaluate our trained policy by per-

forming 200 rollouts and calculate the success rate. We compare our trained policy with a random policy that performs the task with actions obtained from randomly generated displacement vector.

J Object Manipulation using OBJECTFOLDER

In Sec. 4 of the main paper, we have evaluated four benchmark tasks to show the usefulness of our dataset for multisensory perception and control, including instance recognition, cross-sensory retrieval, 3D reconstruction, and robotic grasping. To further demonstrate the potential of using our dataset for real robotic manipulation tasks, we perform an additional experiment on an object manipulation task using Meta-World [13]: *reach*. We follow the standard MT-1 setting in the Meta-World paper, and train a PPO policy for *reach* with three of our objects (cup, bowl, dice). We achieve 100% success rate for each object, demonstrating the usefulness of our objects for robot manipulation tasks.

References

[1] R. Gao, Y.-Y. Chang, S. Mall, L. Fei-Fei, and J. Wu. Objectfolder: A dataset of objects with implicit visual, auditory, and tactile representations. In *CoRL*, 2021.

[2] B. Calli, A. Singh, A. Walsman, S. Srinivasa, P. Abbeel, and A. M. Dollar. The ycb object and model set: Towards common benchmarks for manipulation research. In *ICRA*, 2015.

[3] Blender - a 3d modelling and rendering package. http://www.blender.org.

[4] J.-H. Wang and D. L. James. Kleinpat: optimal mode conflation for time-domain precomputation of acoustic transfer. In *SIGGRAPH*, 2019.

[5] X. Jin, S. Li, T. Qu, D. Manocha, and G. Wang. Deep-modal: real-time impact sound synthesis for arbitrary shapes. In *ACMMM*, 2020.

[6] S. Wang, M. Lambeta, P.-W. Chou, and R. Calandra. Tacto: A fast, flexible and open-source simulator for high-resolution vision-based tactile sensors. *arXiv preprint arXiv:2012.08456*, 2020.

[7] M. Guo, A. Fathi, J. Wu, and T. Funkhouser. Object-centric neural scene rendering. *arXiv preprint arXiv:2012.08503*, 2020.

[8] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *ECCV*, 2020.

[9] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[10] L. Mescheder, M. Oechsle, M. Niemeyer, S. Nowozin, and A. Geiger. Occupancy networks: Learning 3d reconstruction in function space. In *CVPR*, 2019.

[11] E. Coumans et al. Bullet physics library. *Open source: bulletphysics. org*, 2013.

[12] J. Schulman, S. Levine, P. Abbeel, M. Jordan, and P. Moritz. Trust region policy optimization. In *ICML*, 2015.

[13] T. Yu, D. Quillen, Z. He, R. Julian, K. Hausman, C. Finn, and S. Levine. Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning. In *CoRL*, 2020.