Recovering Parametric Scenes from Very Few Time-of-Flight Pixels

Supplementary Material

In this supplementary material, we provide (1) a description of loss functions for training our feedforward model (Sec. A); (2) additional results on 6D pose estimation (Sec. B); (3) an analysis of runtime and complexity (Sec. C); (4) experiments and discussion on sensor interference (Sec. D); and (5) additional visualization of our results on 6D pose estimation (Sec. E).

For sections, figures and equations, we use numbers (*e.g.*, Sec. 1) to refer to the main paper and capital letters (*e.g.*, Sec. A) to refer to this supplement.

A. Training Loss of Feedforward Models

A.1. 6D Pose Estimation

As described in Sec. 4.2, we utilize one of two losses to train the feedforward model depending on if the object is symmetrical. For non-symmetrical objects, we utilize a combination rotation, translation, and point matching loss. Given a ground truth object rotation $\mathbf{R}_{\text{gt}}$ (represented by the 6D representation proposed by [51]) and translation $\mathbf{t}_{\text{gt}}$. Given a set of 3D points $\mathbf{x}_i$ on the object, the loss of the predicted rotation $\mathbf{R}$ and translation $\mathbf{t}$ is given by:

$$\mathcal{L} = \lambda_r \mathcal{L}_{rot} + \lambda_t \mathcal{L}_{trans} + \lambda_p \mathcal{L}_{pm}$$

where the loss terms are given by

$$\begin{aligned}
\mathcal{L}_{rot} &= \|\mathbf{R} - \mathbf{R}_{\text{gt}}\|_1, \\
\mathcal{L}_{trans} &= \|\mathbf{t} - \mathbf{t}_{\text{gt}}\|_1, \\
\mathcal{L}_{pm} &= \frac{1}{N} \sum_{i=1}^{N} \|(\mathbf{R}\mathbf{x}_i + \mathbf{t}) - (\mathbf{R}_{\text{gt}}\mathbf{x}_i + \mathbf{t}_{\text{gt}})\|_2
\end{aligned}$$

We set $\lambda_r = 1.0$, $\lambda_t = 0.5$, $\lambda_p = 0.1$ for our experiments.

For symmetric objects, we use ADD-S loss introduced in [47], where $\mathcal{X}$ represents the set of object points:

$$\mathcal{L}_{ADD-S} = \frac{1}{N} \sum_{i=1}^{N} \min_{\mathbf{x}_j \in \mathcal{X}} \|(\mathbf{R}\mathbf{x}_i + \mathbf{t}) - (\mathbf{R}_{\text{gt}}\mathbf{x}_j + \mathbf{t}_{\text{gt}})\|_2$$

A.2. Spherical Object Recovery

For spherical object recovery (Sec. 5.1), the scene is parameterized by the center point $\mathbf{c} \in \mathbb{R}^3$ and diameter d. Our loss function is a simple combination of error in the two components:

$$\mathcal{L} = \|\mathbf{c} - \mathbf{c}_{\text{gt}}\| + \lambda |d - d_{\text{gt}}|$$

We set $\lambda = 1$ for our experiments.

A.3. Human Hand Pose Estimation

For hand pose estimation (Sec. 5.2), we predict the MANO model [36] shape parameters $\boldsymbol{\beta}$, pose parameters $\boldsymbol{\theta}$, global 3D rotation $\mathbf{R}$ (represented by the 6D representation proposed by [51]), and global 3D translation $\mathbf{t}$. The loss for a given prediction is given by:

$$\begin{aligned}
\mathcal{L} = \lambda_s \mathcal{L}_{shape} + \lambda_p \mathcal{L}_{pose} + \lambda_r \mathcal{L}_{rot} \\
+ \lambda_t \mathcal{L}_{trans} + \lambda_j \mathcal{L}_{joint} + \lambda_v \mathcal{L}_{vertex}
\end{aligned}$$

where the loss terms are given by

$$\begin{aligned}
\mathcal{L}_{shape} &= \|\boldsymbol{\beta} - \boldsymbol{\beta}_{\text{gt}}\|_1, \\
\mathcal{L}_{pose} &= \|\boldsymbol{\theta} - \boldsymbol{\theta}_{\text{gt}}\|_1, \\
\mathcal{L}_{rot} &= \|\mathbf{R} - \mathbf{R}_{\text{gt}}\|_1, \\
\mathcal{L}_{trans} &= \|\mathbf{t} - \mathbf{t}_{\text{gt}}\|_1, \\
\mathcal{L}_{j} &= \|(\mathbf{R}\mathcal{M}_j(\boldsymbol{\beta}, \boldsymbol{\theta}) + \mathbf{t}) - (\mathbf{R}_{\text{gt}}\mathcal{M}_j(\boldsymbol{\beta}_{\text{gt}}, \boldsymbol{\theta}_{\text{gt}}) + \mathbf{t}_{\text{gt}})\|_2, \\
\mathcal{L}_{v} &= \|(\mathbf{R}\mathcal{M}_v(\boldsymbol{\beta}, \boldsymbol{\theta}) + \mathbf{t}) - (\mathbf{R}_{\text{gt}}\mathcal{M}_v(\boldsymbol{\beta}_{\text{gt}}, \boldsymbol{\theta}_{\text{gt}}) + \mathbf{t}_{\text{gt}})\|_2
\end{aligned}$$

Where $\mathcal{M}_j$ is the MANO model that outputs joint keypoint positions, and $\mathcal{M}_v$ is the MANO model that outputs mesh vertex positions. We set $\lambda_s = 0.1$, $\lambda_p = 0.1$, $\lambda_r = 1.0$, $\lambda_t = 1.0$, $\lambda_j = 0.1$, $\lambda_v = 0.1$ for our experiments.

B. Additional 6D Pose Estimation Experiments

B.1. Data Visualization

We visualize the transient histograms captured by multiple, distributed ToF sensors across two different 3D scenes in Fig. A. The measurement has a complex relationship with scene geometry. We aim to solve the inverse problem (multi-view transient histogram → geometry) for simple parametric scenes.

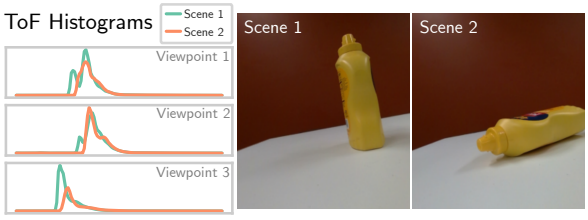


Figure A. Transient histograms from multiple viewpoints alongside corresponding 3D scenes.

B.2. Fine-Tuning on Real Data

We investigate the effects of fine-tuning our feedforward model on real data. To do so, we capture 80 additional measurements of the matte white "2" object used in prior

Training Data	AUC-ADD (↑) Feedforward	FF + Refiner
Fully Sim.	74.67	83.47
Finetune on Real	**86.16**	**90.36**

Table A. Results of fine-tuning the 6D pose estimation method on real data, over 25 measurements of the "2" object.

Figure B. "2" objects with different reflectance properties used in the varying scene reflectance experiment (Sec. B.3. From left to right: matte white, glossy white, and spotted black and white.

experiments, and fine-tune the model trained on simulated data on these measurements. We leave the refiner unmodified.

The results of the fine-tuning experiment are presented in Tab. A. We see a significant improvement in the performance of the feedforward network. We also see a significant improvement in the result after refinement due to the improved starting estimate from the feedforward network. These results are encouraging as they indicate that a minimal amount of real-world data could improve the performance of our method.

B.3. Varying Scene Reflectance

The transient is a product of scene geometry *and* reflectance, so scenes of varying reflectance could affect the performance of our method. We conduct a systematic test in which we modify the reflectance properties of the 3D printed digit "2" and the tabletop surface. We test "2" objects with three surface finishes, as shown in Fig. B. We test two table materials: matte white and matte black.

The results of varying surface properties are presented in Tab. B. A modest decline in performance is observed with the glossy white object and the matte black tabletop, while a significant drop in performance occurs with the spotted black-and-white object. We attribute this drop to the fact that the spotted object has strong low-frequency variations in albedo across the surface. This sort of albedo variation is not included in our domain randomization when generating simulated data, nor is it able to be modeled by our refiner.

B.4. Varying Ambient Light

We evaluate the performance of our method under varying levels of ambient lighting in Tab. C, on a new set of 10 captures of the "2" object at each light level. We see consistent

Obj. Material	Table Material	AUC-ADD (↑) FF	FF + Refiner
Matte White	Matte White	74.67	83.47
Matte White	Matte Black	66.59	79.55
Glossy White	Matte White	69.86	77.74
Spotted B/W	Matte White	50.46	61.49

Table B. 6D Pose Estimation of the "2" object with varying object and tabletop surface reflectance.

Ambient Light Level	AUC-ADD(↑)	AUC-ADD-S(↑)
< 0.1 lux	65.69	90.47
300 lux	72.45	93.10
3000 lux (heavy IR)	25.45	26.53

Table C. 6D Pose Estimation of the "2" object under varying levels of ambient illumination.

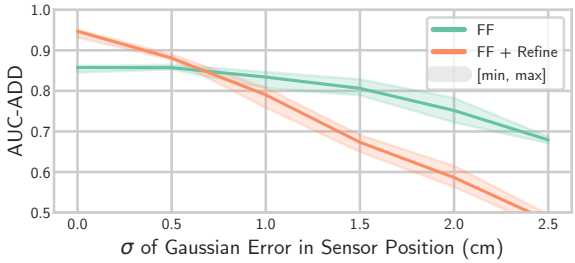


Figure C. Effect of adding Gaussian error to sensor poses on the "2" pose estimation task before feeding into our method.

performance in darkness ($<$0.1 lux) and the same indoor lights as used in other captures (300 lux), but a heavy falloff in performance under a very bright halogen spotlight (3000 lux), which emits high amounts of infrared light, leading to a high DC offset in the transient histogram. This performance drop is expected as we assume negligible ambient light in both synthetic data generation and the refiner. Future work could aim to alleviate this problem by including ambient light level in domain randomization to make the feedforward network more robust, pre-processing histograms to mitigate the effect of ambient light, and/or optimizing for ambient light level in the refiner.

B.5. Sensitivity to Inaccuracy in Sensor Pose

When generating synthetic data to train our method, we add random Gaussian noise to the simulated sensor position to increase robustness to real-world inaccuracies in sensor position. We perform a simulated experiment to test this robustness, the results of which are shown in Fig. C. Both the feedforward model and refiner are robust to modest variations in sensor pose ($<$ 1cm), which are likely achievable in realistic settings. We find that the feedforward method is more robust to variations than the refiner, and when there is high variation in sensor pose, foregoing the refiner leads to higher accuracy in the recovered object pose.

Number of Views	AUC-ADD (AUC-ADD-S) (↑) Feedforward	FF+Refiner
5 Pixels (Views)	74.79 (90.34)	74.80 (90.34)
10 Pixels (Views)	78.29 (90.57)	78.07 (90.63)
15 Pixels (Views)	84.65 (91.27)	84.79 (91.66)
25 Pixels (Views)	87.48 (91.51)	87.40 (91.42)
50 Pixels (Views)	90.54 (94.33)	90.87 (94.59)
100 Pixels (Views)	91.47 (94.58)	91.49 (94.37)

Table D. 6D Pose Estimation with Different Numbers of Views.

Ablation	AUC-ADD (↑) Feedforward	FF + Refiner
Full Model	73.67	83.47
Idealized Jitter Kernel	22.64	24.50
Incorrect Bin Size	49.98	33.23
Incorrect FoV	29.70	44.22

Table E. Results of 6D Pose Estimation under varying sensor model ablations, over a dataset of 25 captures of the "2" object.

B.6. Sensor Model Ablation Study

We perform an ablation study over key components of our sensor model as described in Sec. 3.1. We consider the following variants:

1. **Full:** The full sensor model as described in Sec. 3.1 and used for all previous experiments.
2. **Idealized Jitter Kernel:** The jitter kernel $\boldsymbol{s}$ is replaced by a Dirac delta function at the location of the peak of $\boldsymbol{s}$.
3. **Inaccurate Bin Size:** The temporal bin size Δt of the transient histogram is $\sim$10% smaller than as calibrated (from 1.38cm to 1.2cm).
4. **Inaccurate FoV Size:** The angular size of the FoV is incorrect by $\sim 20\%$, increasing from 32° to 38°. Additionally, the intensity map $I(\boldsymbol{\omega})$ is replaced with a constant function.

For each variant, we train a feedforward model on synthetic data generated with the ablated sensor model, and use the same ablated sensor model in our refiner. Results over the 25-pose "2" digit dataset are shown in Tab. E. The results demonstrate that each of these aspects of sensor modeling are important to achieve good performance.

C. Runtime and Complexity Analysis

While our method foregoes some computation performed by traditional methods (*e.g.* peak finding and ICP), it is replaced by relatively costly neural network inference and iterative pose refinement. Therefore we do not foresee efficiency improvements compared to point cloud-based methods. One feed-forward pass of our network takes $\sim$4.8 ms. The (unoptimized) refiner takes $\sim$2 seconds. With attention paid to efficiency, refiner speed could likely be increased. The costs of both the forward pass and optimization scale linearly with the number of viewpoints.

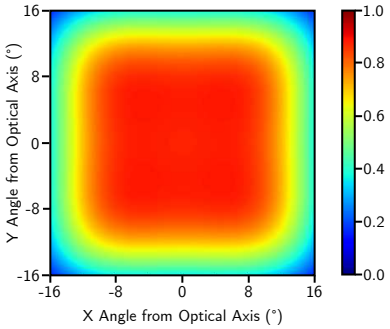
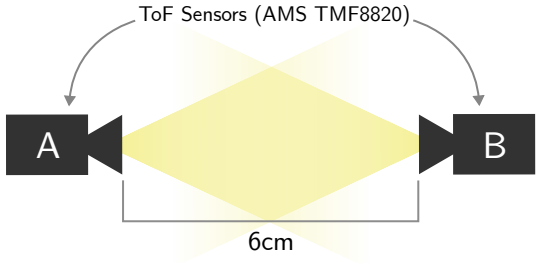
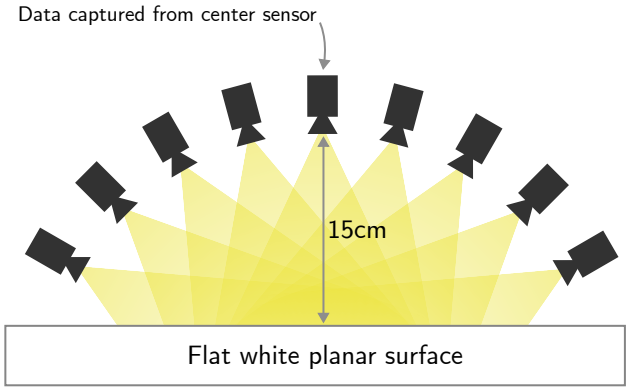


Figure D. Visualization of the laser intensity function $I(\boldsymbol{\omega})$ that we use for the TMF8820 sensor, as given by Eq. (9). We set $K_1 = 0.88$, $K_2 = -3.16$, $K_3 = 250.51$.



(a) Sensor configuration for interference experiment 1.



(b) Sensor configuration for interference experiment 2.

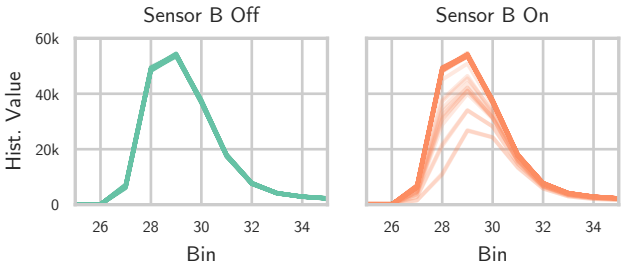
Figure E. Sensor configurations used for interference experiments.

D. Test of Between-Sensor Interference

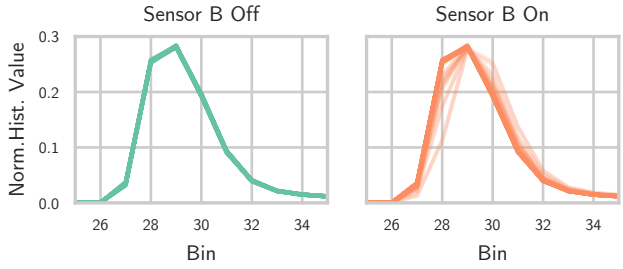
In our prototype system, a single sensor is moved to multiple positions while the scene remains static. However many practical applications for our method may involve multiple sensors imaging the scene at the same time, which could lead to interference between sensors. We perform controlled experiments to investigate the effect of interference.

D.1. Two Sensors Facing Each Other

We position two AMS TMF8820 sensors facing directly at each other at a distance of 6cm, as illustrated in Fig. Ea. We compare measurements captured by sensor A between two conditions: sensor B on and sensor B off. The raw and nor-



(a) Raw histograms



(b) Histograms normalized to have a sum of 1.

Figure F. Comparison of the histograms captured in interference experiment 1. Each plot shows 128 sensor measurements overlaid. About 90% of samples in the right column exhibit no interference artifacts, comprising the dark orange lines.

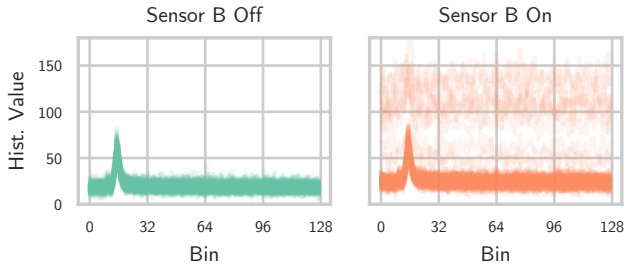
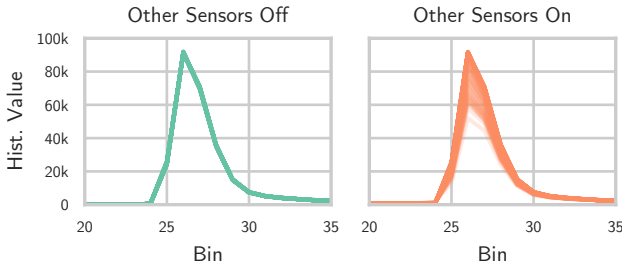


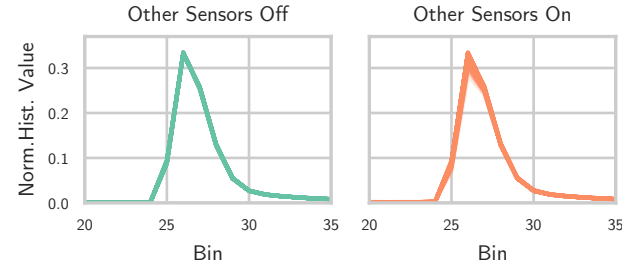
Figure G. Comparison on the histograms captured in interference 1, with the light source of sensor A covered. Each plot shows 128 sensor measurements overlaid. About 90% of the samples in the right column exhibit no interference artifacts, comprising the dark orange line.

malized histograms for both conditions are shown in Fig. F. We find that the operation of sensor B causes an effect in the histogram captured by sensor A $\sim 10\%$ of the time. Even after normalization, the effect is still present. This effect appears similar to the effect caused by ambient light [13], and is consistent with what we would expect to see if sensor B's light source is not correlated with the light source of sensor A; *i.e.*, because the laser pulse trains of the two sensors are not synchronized, sensor B's operation leads to photons arriving uniformly at any time relative to sensor A's pulse train, just as ambient light arrives uniformly.

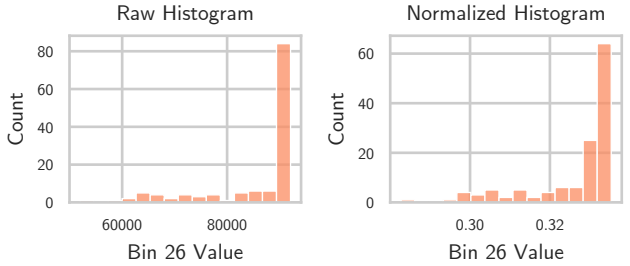
To further validate this hypothesis, we perform another test using the same sensor configuration in which the laser



(a) Raw histograms



(b) Histograms normalized to have a sum of 1



(c) Histogram of values of bin 26 (the peak) for the "Other Sensors On" condition. The bin values are grouped together in about 75% of measurements.

Figure H. Comparison of the histograms captured in interference experiment 2.

light source of sensor A is covered, so that only ambient light and the effect of sensor B are captured by sensor A. The results of this experiment are shown in Fig. G. In this case we can clearly see interference manifest as a DC offset in the captured histogram, again matching the signature of ambient light.

D.2. Nine Sensors Imaging a Plane

We perform a second experiment in which nine sensors are all operating simultaneously and imaging the same portion of a planar surface. The experimental setup is illustrated in Fig. Eb. We position the sensors such that the centers of their optical axes each intersect with a planar surface at the same point, and record data only from the center sensor. Again, we compare between two conditions: the other 8 sensors on, and the other 8 sensors off. The results of this experiment are shown in Fig. H. We see the same effect as

in the previous experiment, but with a slightly higher occurrence rate of $\sim 25\%$.

D.3. Discussion: Between-Sensor Interference

We have demonstrated that, at least for the AMS TMF8820 sensor, the effect of interference between sensors happens only occasionally even in the worst case. In practical scenarios, the rate of interference is likely to be quite low (*i.e.* $< 10\%$). Further, the effect of interference on the histogram appears to be similar to the effect of ambient light. Adjusting captured histograms to account for ambient light is a well-studied problem [13], and it is likely that methods which are robust to changes in ambient light will be robust to between-sensor interference. While future applications should take interference into account, we believe it is unlikely to be a major obstacle for future deployments of distributed miniature ToF sensors.

E. Visualization of 6D Pose Results

We provide visualization of our results on 6D pose estimation in Figures J to R.

Figure I. Objects used for 6D pose estimation experiments.

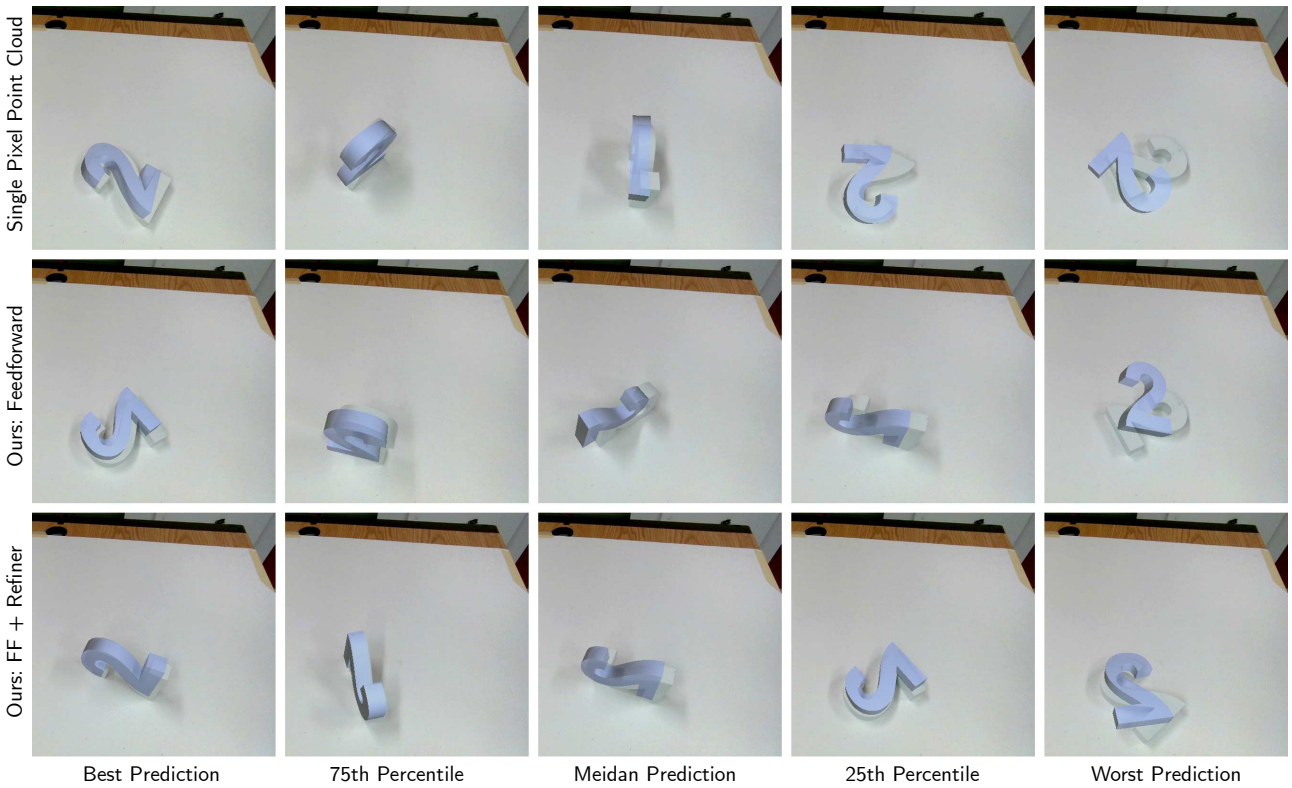


Figure J. Visualization of results on the 3D printed "two" object.

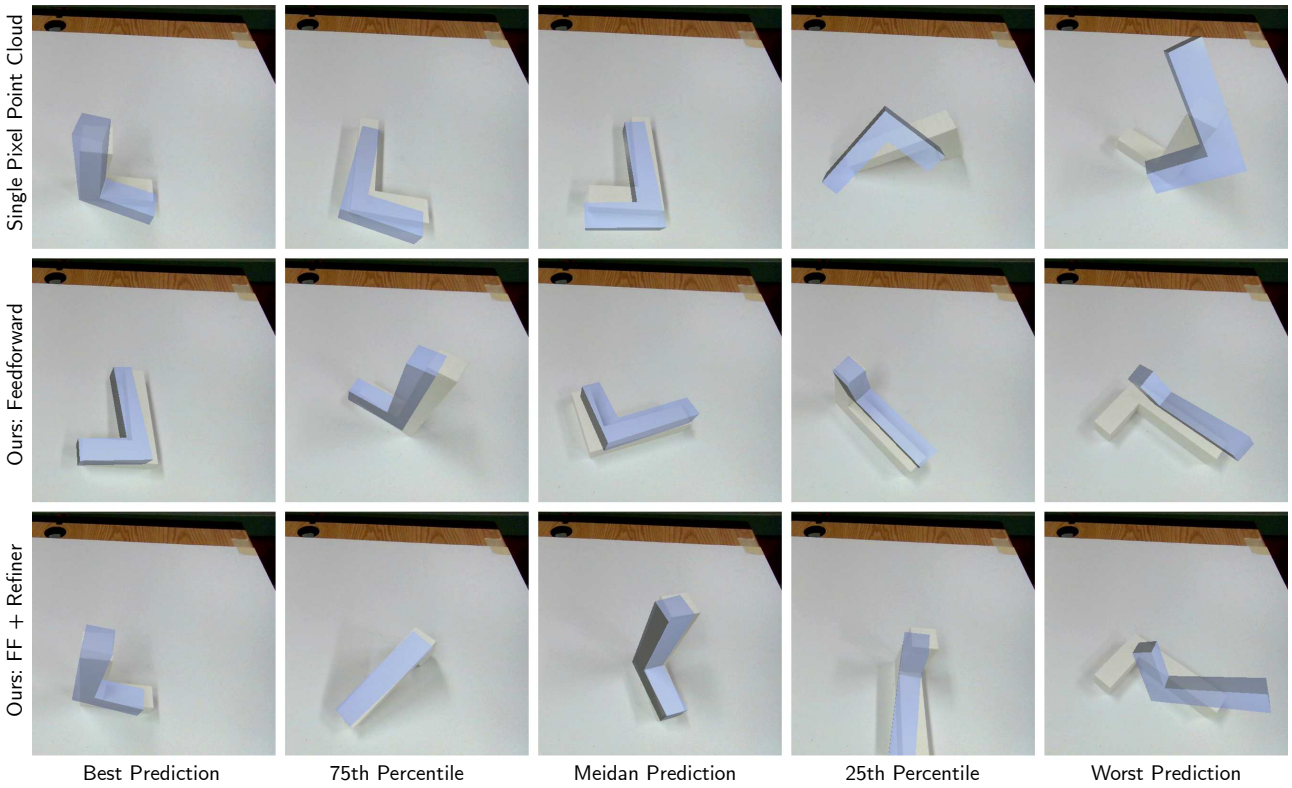


Figure K. Visualization of results on the 3D printed “L” object.

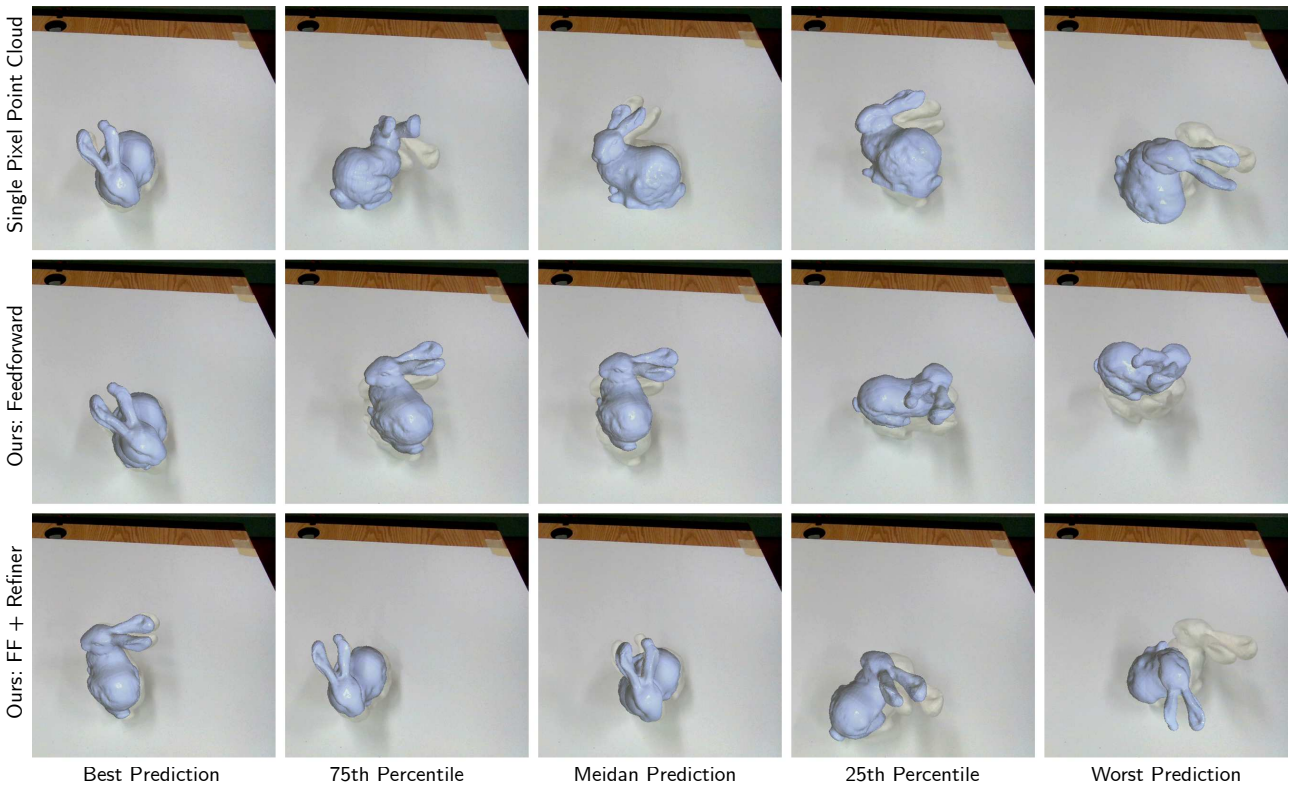


Figure L. Visualization of results on the 3D printed “bunny” object.

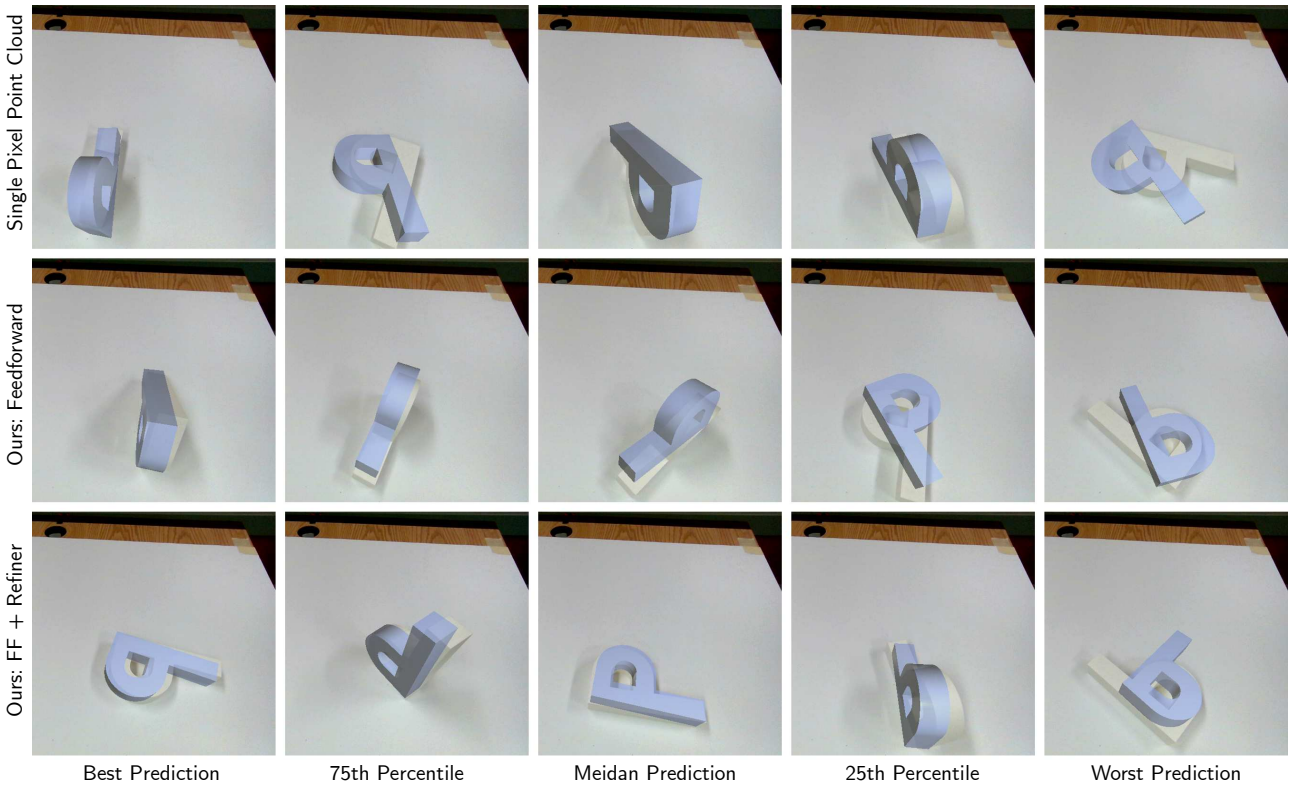


Figure M. Visualization of results on the 3D printed "P" object.

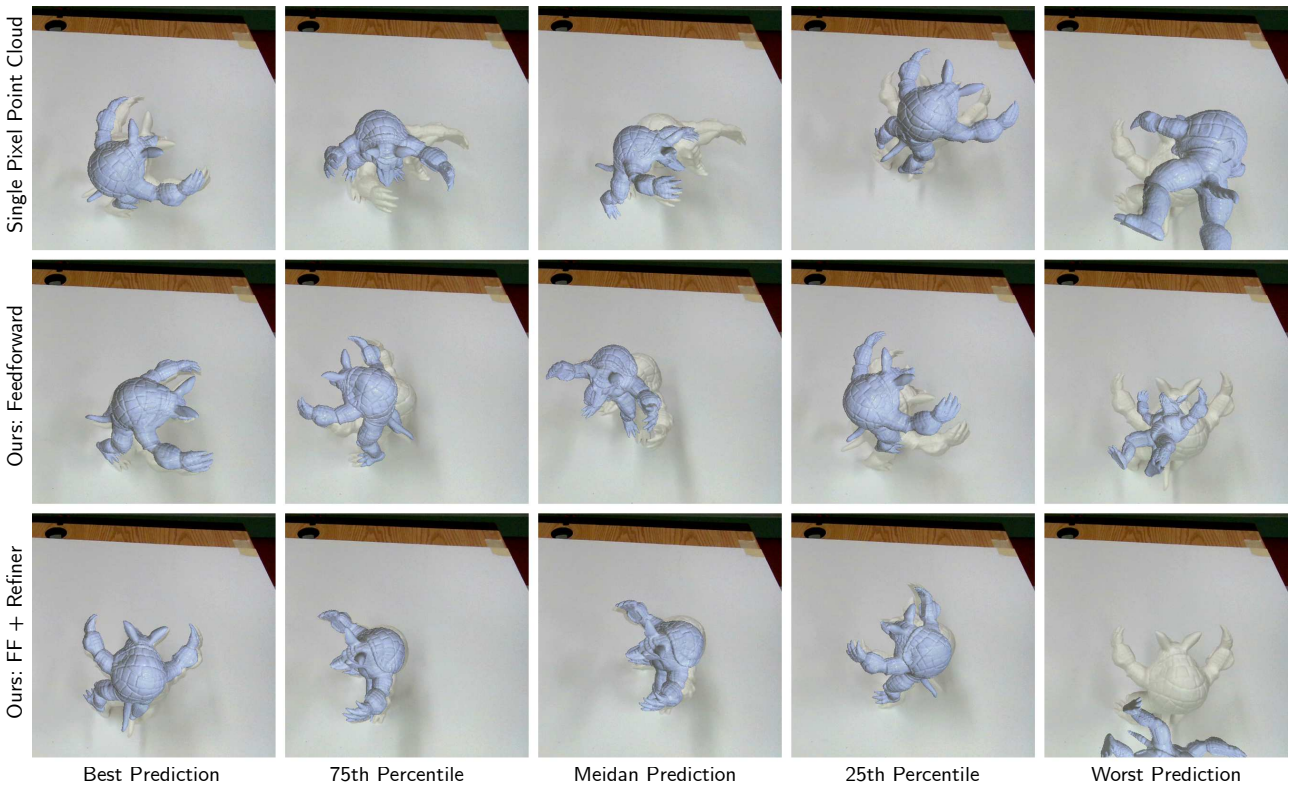


Figure N. Visualization of results on the 3D printed "armadillo" object.

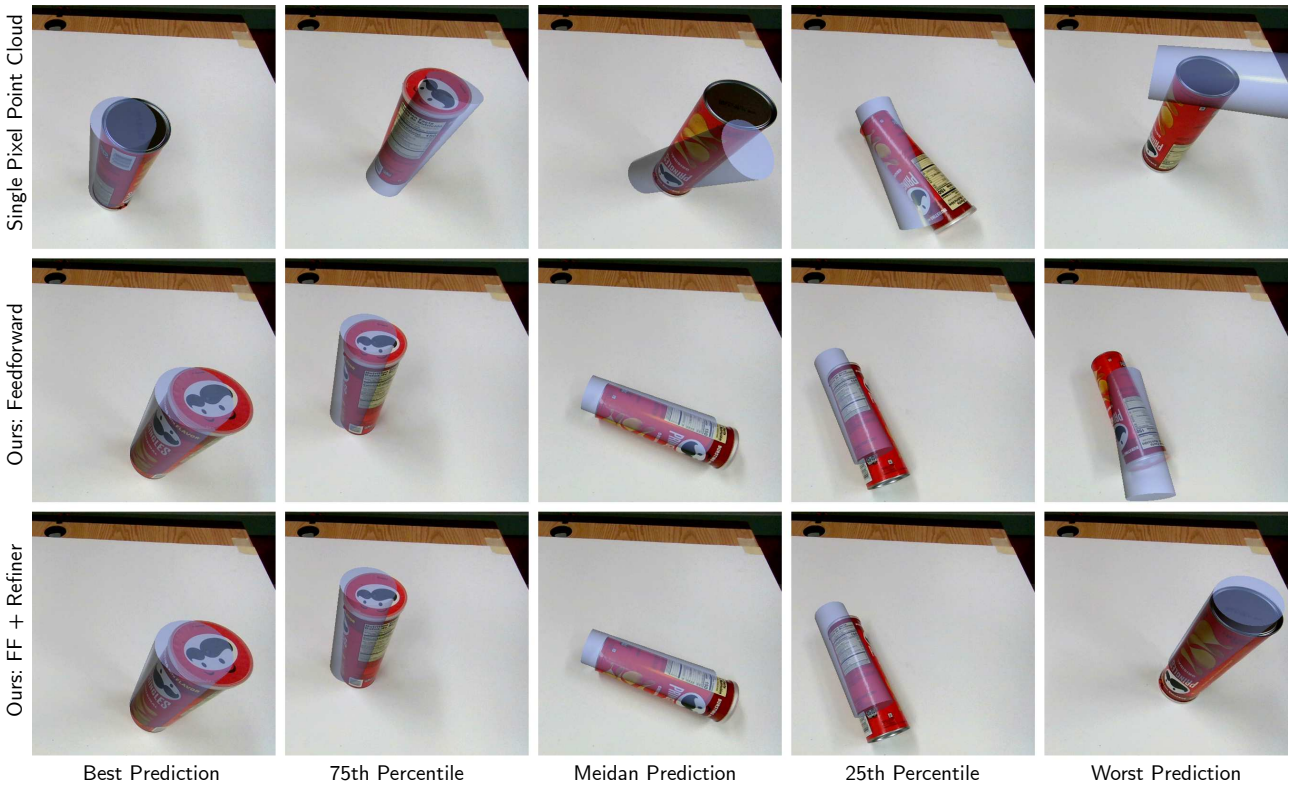


Figure O. Visualization of results on the "chips" object from the YCB dataset.

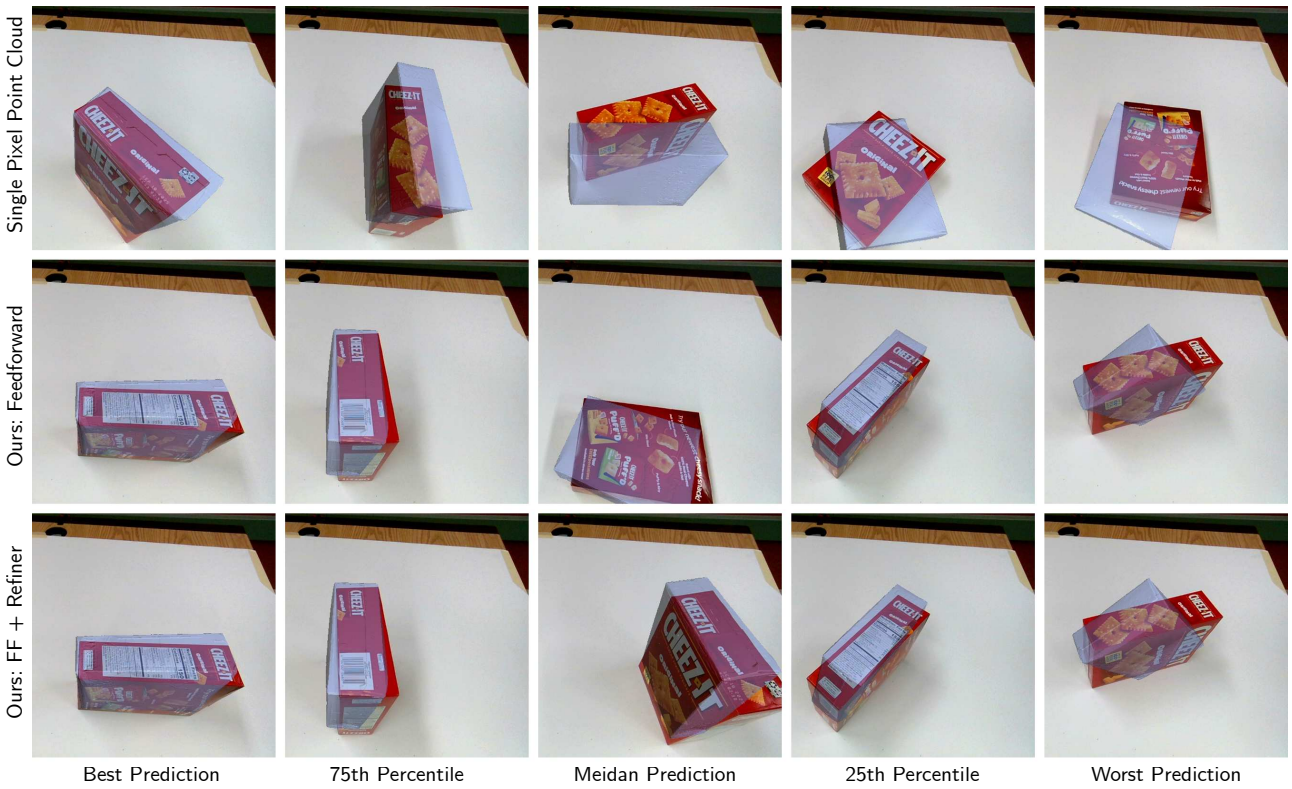


Figure P. Visualization of results on the "crackers" object from the YCB dataset.

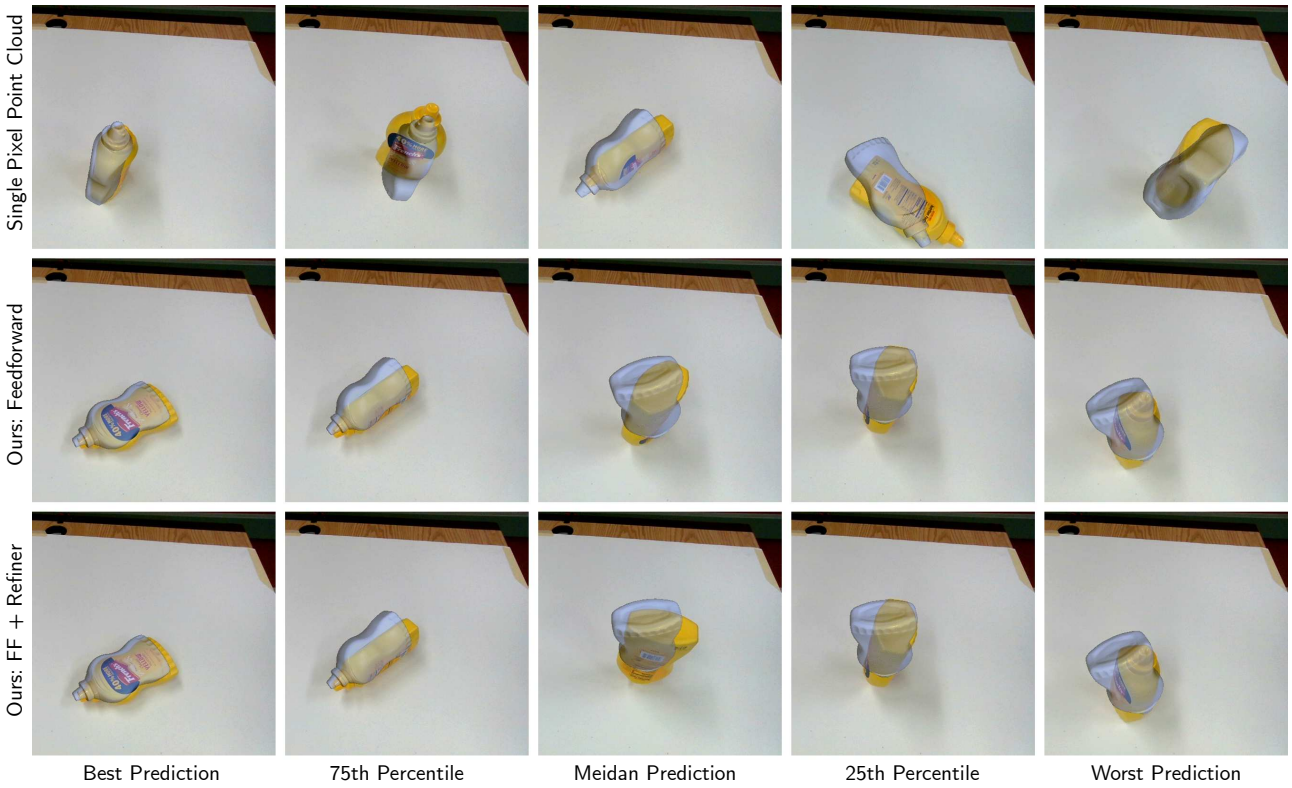


Figure Q. Visualization of results on the "mustard" object from the YCB dataset.

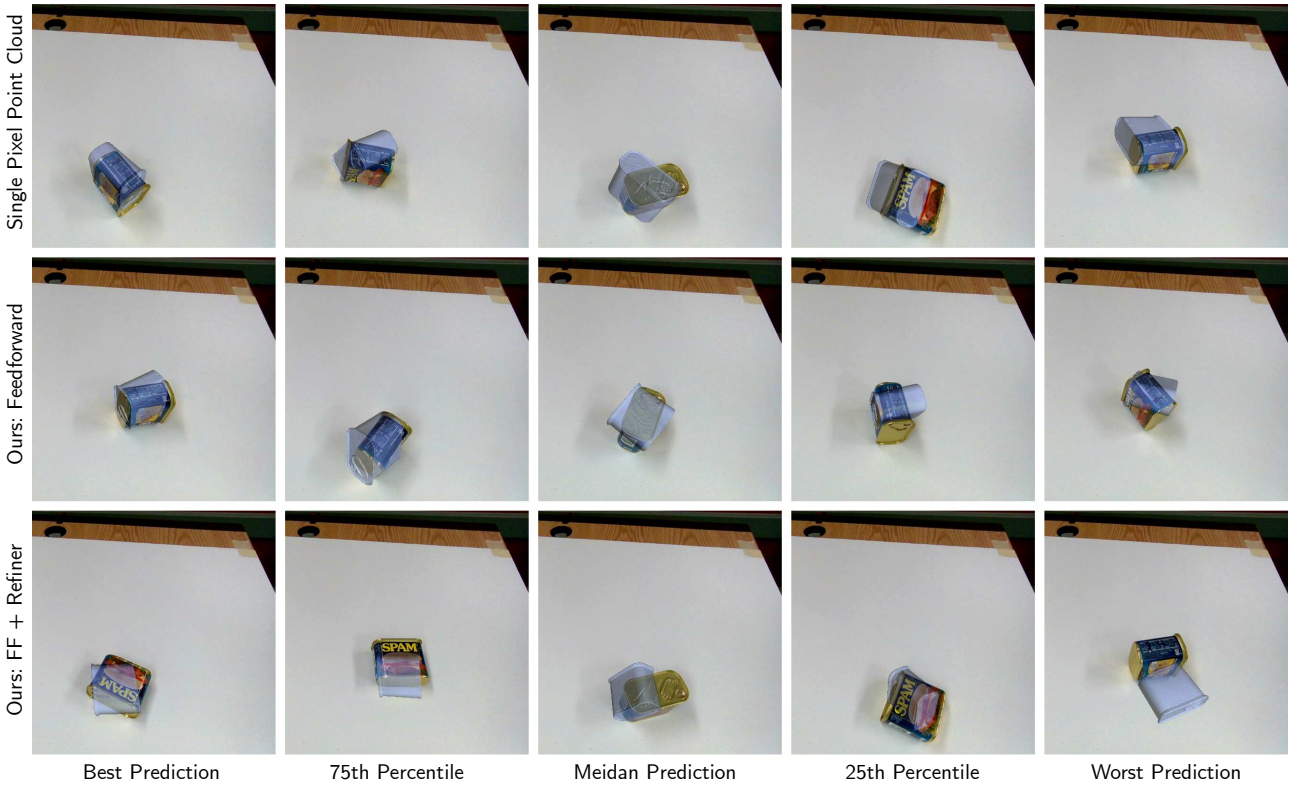


Figure R. Visualization of results on "SPAM" object from the YCB dataset. The SPAM is a failure case for our method due to its specular surface, small size, and many near-symmetries which make optimization difficult.