DiVa-360: The Dynamic Visual Dataset for Immersive Neural Fields

Supplementary Material

1. Design of Brown Interaction Capture System (BRICS)

Aluminum Frame: To capture table-scale scenes, we chose a refrigerator-sized aluminum frame (Figure 3 (a) of the main paper) that houses a 1 m^3 capture volume mounted on wheels for mobility. Each of the 6 side walls of the capture volume is composed of a 3×3 grid with dual polycarbonate panels on each grid square (total of 54 squares). Two of the walls are doors that allow quick access to the capture volume. The height of the system allows an average person to easily reach into the volume for interaction capture. A transparent polycarbonate shelf in the capture volume allows bottom cameras to still see objects to provide a 360° view. A shelf in the bottom houses power supplies, network switches, and a control workstation.

Sensor/Illumination Panels: For 53 of the 54 grid squares (we leave one out for easy access) on the side walls, we installed translucent polycarbonate panels on the interior consisting of cameras, microphones, and LEDs. Each panel can support up to 3 RGB cameras, 3 microphones, and a fully programmable RGB light strip with 72 individual LEDs. This panel naturally diffuses the LED lights enabling uniform lighting of the volume. In our current setup, each of the 53 panels has an LED strip and 1 off-the-shelf RGB camera capturing at 1280×720 @ 120 FPS. We install microphones on 6 panels, one on each side wall of the capture cube.

Communication Panels: The sensor panels collectively generate more than 13.25 GB/s (0.25 GB/s per panel) of uncompressed data – well beyond the bandwidth of common wired communication technologies like USB or ethernet. To enable the capture and storage of such amounts of data, we built our own communication system. Briefly, this system consists of single-board computers (SBCs) that connect to sensors via USB and are responsible for compressing the data before sending it to a control station over gigabit ethernet. With this setup, we are able to simultaneously transmit large amounts of data with low latency.

Control Workstation: We use a workstation with 52 CPU cores to simultaneously uncompress, store, and transmit all the data. To ensure high throughput, we use a 10 Gigabit ethernet uplink to the SBCs, a PCI solid state drive, and 200 GB RAM for caching.

Panels: BRICS panels are designed to be modular, to allow for quick customization for different research endeavors. BRICS consists of 42 panels in total across six sides. Each side has six square panels of size (9.75 in x 9.75 in) and a single rectangular middle panel of size (32.25 in x 9.75 in) that can be changed to consist of three square panels based on research tasks. The panels inside are white translucent panels made of TAP plastics Satinice White Acrylic to encourage light dispersion towards the inside. Outer panels are white and opaque made of TAP plastics KOMATEX foamed PVC Sheets.

The inner panels allow the mounting of three different cameras or other accessories. Although the panel currently consists of an RGB camera of size (71.5 mm x 71.5 mm) mounted at the center, future plans include attaching depth and infrared cameras. All panels are 1/8th inch in thickness.

Mounts: We utilize custom-designed mounts to attach cameras to the panels. We use custom-designed ball bearing mounts, that are rotatable to allow for changing the camera orientation.

Lighting: We use BTF-Lighting WS2812-B individually addressable RGB lighting strips. This allows for highly customized lighting conditions and environment maps. Moreover, using LED strips allows us to add additional lighting quickly.

Each panel has 70 LED's placed in between the inner panel and outer panel. These LED's are powered individually and sequentially connected for data. The LED's are all controlled with six Raspberry Pi 3 Model B+ computers, one for each side. To control the LED's we used the standard NeoPixel python library. Furthermore, each side allows for individual brightness control.

Cameras: We used off-the-shelf USB 2.0 cameras that can capture 1280×720 @ 120 FPS. Specifically, we used the ELP-SUSB1080P01-LC1100 from ELP Cameras.

Single Board Computers (SBCs): We need the single board computers to have enough processing power and USB ports to support up to 3 cameras and 1 microphone each. For this reason and easy market availability, we chose the Odroid N2+ 4 GB, which was sufficient for our purpose.

2. Dynamic Dataset Benchmark Comparison

Pre-processing: To do a benchmark, we pre-process the raw data captured through BRICS following Section 5.

Baselines Training: Per-Frame I-NGP (PF I-NGP) sequentially learns a model for each time step. Each I-NGP [5] is initialized from the model of the previous time step. The PF I-NGP allows us to fit the dynamic video efficiently while not considering the motion between frames. In addition, the streamable training feature also allows us to optimize the camera pose and lens distortion individually for each frame. We train each I-NGP for 5000 iterations. The average training time for each I-NGP is 48.7 seconds with a standard deviation of 4.4 seconds. The smaller standard deviation

Types	Baseline	PSNR↑	SSIM↑	LPIPS↓	JOD↑
Dynamic objects	PF I-NGP [5]	29.62 / 4.42	0.95 / 0.02	0.06 / 0.02	7.53 / 1.28
	MixVoxels [6]	28.43 / 2.87	0.95 / 0.02	0.06 / 0.03	7.48 / 1.31
	K-Planes [2]	26.62 / 4.41	0.91 / 0.04	0.21 / 0.09	6.66 / 1.47
Interactions	PF I-NGP	26.89 / 1.76	0.94 / 0.03	0.09 / 0.04	7.68 / 0.51
	MixVoxels	26.45 / 1.91	0.93 / 0.03	0.10 / 0.04	7.50 / 0.70
	K-Planes	26.18 / 2.08	0.93 / 0.03	0.17 / 0.06	7.55 / 0.49
Long-duration Sequences	PF I-NGP	29.35 / 0.90	0.93 / 0.03	0.10 / 0.03	7.60 / 0.57
	MixVoxels	29.62 / 0.75	0.93 / 0.02	0.11 / 0.02	7.87 / 0.29
	K-Planes	26.45 / 1.86	0.92 / 0.02	0.21 / 0.04	7.27 / 0.68

Table 1. Rendering quality (mean / standard deviation) of dynamic objects, interactions, and long duration sequences respectively using PF I-NGP[5], MixVoxels.[6], and K-Planes [2]. Although the PSNR of PF I-NGP is much better than MixVoxels in terms of dynamic objects, the PSNR of PF I-NGP is similar to the MixVoxels for interactions and even underperforms MixVoxels for long-duration sequences. This indicates that it is hard to capture the scene occluded by hands and maintain the temporal consistency, especially for static parts such as the chessboard, without utilizing the temporal information. Unlike PF I-NGP and MixVoxels, the performance of K-Planes is more consistent across the three types.

is due to the fact that PF I-NGP does not consider motion. However, this will lead to temporal inconsistency, which is especially obvious at static parts when comparing PF I-NGP with MixVoxels [6] and K-Planes [2]. In Figure 21, we concatenate the pixels from the line across frames in the same view and demonstrate that PF I-NGP contains much white noise and is less temporal smooth.

MixVoxels [6] is trained to capture the dynamic video every 150 frames. We train each MixVoxels for 25000 iterations. We lower the dynamic threshold to capture more dynamic samples for scenes with drastic motion. Although MixVoxels is trained and benchmarked with black backgrounds, we render it with white backgrounds for all figures in the paper. This will make the floaters of MixVoxels darker in the figures. Unlike PF I-NGP, MixVoxels allows us to learn a dynamic NeRF with motion. In addition, MixVoxels trained with multiple frames also encourage temporal consistency (see Figure 21), which is absent in PF I-NGP. However, MixVoxels struggles to capture dynamic parts with complex motion and small dynamic objects. Hence, in Figure 17, we can observe many floaters and noise around the hand of Chess, and cannot see the small train in the Music Box and holes of knitted fabric in the Crochet. We assume the noise and floaters are caused by the insufficient capacity of the dynamic branch when receiving too many dynamic samples. The small objects and fine-grained details are missed because the variation field fails to decompose the scene correctly. For each time step, the average training time is 57.55 seconds, with a standard deviation of 6.96 seconds. The large standard deviation is because MixVoxels will sample more dynamic points for its dynamic branch when learning a scene with complex motion.

For a fair comparison, we also train a K-Planes every 150 frames. We train each K-Planes for 90000 iterations. Similar to the MixVoxels, K-Planes is also more temporally consistent. This is especially obvious in Chess and Put Fruit of Figure 21. K-Planes sometimes fails to reconstruct static parts and tends to generate many floaters, especially for objects with slow motion such as Music Box. Additionally, it misses the fine-grained details of the knitted crochet in Figure 17. This could be the result of overfitting and contamination from the dynamic planes due to weaker decomposition. For each time step, the average training time is 47.59 seconds with a standard deviation of 5.13 seconds. Although the training time of K-Planes is faster than PF I-NGP and MixVoxels without considering standard deviation, K-Planes shows the slowest rendering speed among them.

Table 1 separately shows quantitative results of dynamic objects, interactions, and long-duration sequences. PF I-NGP, MixVoxels, and K-Planes perform better with dynamic objects than interactions. One reason is that models trained on dynamic objects do not need to handle occlusion caused by the hands. Another reason is that hand motion is more complex. The PSNR gap of PF I-NGP is 2.73 dB, the PSNR gap of MixVoxels is 1.98 dB, and the PSNR gap of K-Planes is 0.44 dB. The performance gap between dynamic objects and interactions is more obvious with PF I-NGP because it does not utilize temporal information and, therefore, cannot handle occlusion well. A similar situation happens to PF I-NGP when comparing dynamic objects with long-duration sequences. However, this situation does not occur with MixVoxels when comparing dynamic objects with long-duration sequences, which also capture hand-object interaction scenes but longer. This is because the scenes

Baseline	Motion	PSNR↑	SSIM↑	LPIPS↓	JOD↑
PF I-NGP [5]	Slow	29.76 / 2.18	0.96 / 0.02	0.05 / 0.03	7.13 / 2.57
	Fast	30.06 / 5.15	0.95 / 0.01	0.06 / 0.02	7.60 / 0.60
	Detailed	33.11 / 7.81	0.96 / 0.02	0.07 / 0.03	6.74 / 2.35
	Repetitive	28.76 / 2.16	0.96 / 0.02	0.06 / 0.02	7.75 / 0.71
	Random	31.59 / 8.95	0.95 / 0.01	0.07 / 0.02	6.74 / 2.34
MixVoxels [6]	Slow	29.64 / 2.14	0.96 / 0.02	0.05 / 0.03	7.58 / 2.52
	Fast	28.39 / 3.09	0.95 / 0.02	0.07 / 0.02	7.41 / 0.73
	Detailed	30.25 / 3.95	0.96 / 0.02	0.06 / 0.03	6.59 / 2.57
	Repetitive	28.37 / 2.35	0.94 / 0.02	0.09 / 0.02	6.94 / 0.72
	Random*	27.89 / 5.75	0.95 / 0.02	0.07 / 0.03	5.91 / 1.99
K-Planes [2]	Slow	25.18 / 2.05	0.88 / 0.03	0.31 / 0.03	6.10 / 1.03
	Fast	26.99 / 5.23	0.92 / 0.05	0.18 / 0.09	6.73 / 1.63
	Detailed	27.91 / 8.54	0.88 / 0.05	0.26 / 0.10	5.68 / 1.54
	Repetitive	25.43 / 2.78	0.91 / 0.05	0.23 / 0.10	6.47 / 1.60
	Random	30.86 / 6.02	0.92 / 0.04	0.19 / 0.07	6.65 / 1.38

Table 2. Rendering quality (mean / standard deviation) of dynamic objects using PF I-NGP[5], MixVoxels[6] and K-Planes[2] in terms of different types of motion. * indicates that the results do not include the Plasma Ball Clip sequence because the codebase cannot handle that scene. PF I-NGP serves as a baseline without considering temporal information. Compared with PF I-NGP, the PSNR of MixVoxels decreases by 0.12 dB, 1.67 dB, 2.86 dB, 0.39 dB, and 3.7 dB for slow, fast, detailed, repetitive, and random motion, respectively. MixVoxels perform better on scenes with slow and repetitive motion while worse on scenes with drastic motion. Compared with PF I-NGP, the PSNR of K-Planes decreases by 4.58 dB, 3.07 dB, 5.2 dB, 3.33 dB, and 0.73 dB for slow, fast, detailed, repetitive, and random motion, respectively. K-Planes perform better on scenes with fast and random motion while worse on scenes with less drastic motion.

of long-duration sequences often contain a large portion of static parts, such as the chessboard in the Chess Long (see Figure 10). MixVoxels can largely benefit from this kind of sequence because the independent static branch of the MixVoxels can handle static parts smoothly across frames. Interestingly, this allows MixVoxels to outperform PF I-NGP, which is less temporally smooth across these frames (see Figure 21). Unlike PF I-NGP and MixVoxels, the performance of K-Planes is robust across three types because of the relatively soft static dynamic decomposition.

Dynamic Object Results: Table 2 shows the performance of PF I-NGP, MixVoxels, and K-Planes in different motion types. We manually classified the 21 dynamic object sequences into five overlapping categories: slow, fast, detailed, repetitive, and random (Table 3). Although PF I-NGP does not consider motion, it serves as a baseline for dynamic models by fitting each frame separately.

All baselines perform slightly better on detailed motions. However, a gap exists between MixVoxels and PF I-NGP's quantitative results, suggesting that MixVoxels can capture the static background but struggles to capture detailed motion (e.g. the second hand of the Clock in Figure 3 disappears). Unlike MixVoxels, K-Planes partially reconstructs the second hand but fails to reconstruct the static part well (e.g., the clock's body). This leads to a larger performance gap between K-Planes and PF I-NGP. Slow and repetitive motions are the two categories that MixVoxels' results are the closest to PF I-NGP's while the performance gaps are larger in fast and random motions. In other words, MixVoxels can successfully capture dynamic information when the motion is continuous and gradual but cannot generalize well to drastic motion. For example, the Horse in Figure 1 and the Penguin in Figure 4 are clean, while the Blue Car in Figure 2 and the Wolf in Figure 5 miss fine-grained details or parts. Fast and random motions are the two categories that K-Planes' results are the closest to PF I-NGP's while the performance gaps are larger in slow and repetitive motions. In other words, K-Planes can construct scenes of drastic motion with fewer artifacts but may produce more artifacts, such as floaters for mild motion. For instance, the Dog in Figure 2 and the Wolf in Figure 5 contain fewer floaters and artifacts than the Horse in Figure 1 and the Stirling Engine in Figure 4. Together with the quantitative results (Table 4), the visualization results suggest that PF I-NGP can successfully capture most of the scene, whereas Mixvoxels and K-Planes struggle to generalize to different types of motions and may need hyperparameter tuning to fit scenes with different motion types.

Interaction Results: Our interaction scenes, which cover several human daily activities such as flipping a book in

Scene	Slow	Fast	Detailed	Repetitive	Random	Size L×W×H (cm)
blue car	✗	✓	✗	✓	✗	23.20 × 9.50 × 5.20
bunny	✗	✓	✗	✓	✗	33.02 × 27.94 × 19.05
clock	✓	✗	✓	✗	✗	30.48 × 30.48 × 4.57
dog	✗	✓	✗	✗	✓	27.00 × 11.99 × 27.00
horse	✓	✗	✗	✓	✗	11.50 × 7.60 × 13.50
hourglass	✓	✗	✓	✗	✓	7.00 × 7.00 × 17.00
k1 double punch	✗	✓	✗	✓	✗	9.80 × 17.20 × 35.00
k1 handstand	✗	✓	✗	✗	✗	9.80 × 17.20 × 35.00
k1 push up	✗	✓	✗	✓	✗	9.80 × 17.20 × 35.00
music box	✓	✗	✗	✓	✗	10.80 × 10.80 × 12.60
penguin	✗	✗	✗	✓	✗	24.69 × 27.00 × 13.77
plasma ball	✗	✓	✓	✗	✓	15.00 × 15.00 × 24.10
plasma ball clip	✗	✓	✓	✗	✓	15.00 × 15.00 × 24.10
red car	✗	✓	✗	✓	✗	22.50 × 9.20 × 5.20
stirling	✗	✓	✓	✓	✗	9.00 × 9.00 × 13.00
tornado	✗	✓	✗	✓	✗	11.00 × 11.00 × 27.50
trex	✗	✓	✗	✗	✗	30.50 × 7.50 × 20.50
truck	✗	✓	✗	✗	✗	17.80 × 7.10 × 7.30
wall-e	✗	✓	✗	✗	✗	35.56 × 20.32 × 27.94
wolf	✗	✗	✗	✓	✓	17.78 × 12.70 × 35.56
world globe	✓	✗	✗	✓	✗	13.97 × 13.97 × 19.81

Table 3. Motion types and object size (cm) of each dynamic object. Objects are split into 5 overlapping categories: slow, fast, detailed, repetitive, and random motion.

Figure 7, require the model to capture a sequence of realistic motions. For example, the Battery scene in Figure 6 contains the motion of using a screwdriver to open the toy's battery cover, putting in the batteries, assembling the cover back, and turning on the toy. We consider the interaction data as more challenging cases for neural radiance fields because of the occlusion and complex motion from the hands. Table 5 demonstrates all results of PF I-NGP, MixVoxels, and K-Planes on interaction scenes. Overall, the performances of these three baselines are similar across interaction scenes. PF I-NGP performs robustly across most of the scenes while failing to totally reconstruct the cover of the book in Figure 7 and produces scratches on the hand in Figure 8. MixVoxels reconstructs blurrier texts in Figure 7, and a few floaters around the hand in Figure 9. K-Planes reconstructs many floaters around the hand and objects in Figures 6, 8 and 9, misses the text in Figure 7, distorts the hand in Figure 6, and produces white scratches on the hand in Figure 8. We hope that our interaction dynamic dataset can open a new direction and provide a new understanding for hand object interaction tasks [1] in the future. Notably, visualization results show black artifacts in Figure 6 and Figure 9 in MixVoxels renderings. This effect is not observable in numerical results because all models except K-Planes are trained and evaluated with black backgrounds. We used white backgrounds for rendering to better visualize the results.

Long duration Results: Although our object and interaction datasets contain several long videos, we propose a long-duration dataset that only includes videos that are at least 2 minutes long (see Figure 22). Table 6 details the rendering quality of PF I-NGP, MixVoxels, and K-Planes on each scene of the long-duration data. Surprisingly, MixVoxls outperforms PF I-NGP in many scenes of the dataset. This is because many scenes of the dataset contain large static parts such as a chessboard in the Chess Long, and the tray in the Jenga Long, Legos, Origami, Painting, and Puzzle. MixVoxels is good at maintaining the consistency of the static parts across frames, while PF I-NGP cannot (see Figure 21). Figure 11 demonstrates that PF I-NGP still provides more fine-grained details, such as the holes of fabrics in Crochet, than MixVoxels and K-Planes. Figure 10 shows that MixVoxels performs better at static parts like the chessboard while K-Planes performs better at dynamic parts such as the hand.

3. Dynamic Dataset Experiments

In this section, we discuss the visualization results for each experiment in Section 5.2 of the paper.

Temporal Information: Figure 12 demonstrates the visu-

Scene	PF I-NGP / MixVoxels / K-Planes PSNR↑	PF I-NGP / MixVoxels / K-Planes SSIM↑	PF I-NGP / MixVoxels / K-Planes LPIPS↓	PF I-NGP / MixVoxels / K-Planes JOD↑
blue car	29.833 / 29.485 / 28.304	0.957 / 0.955 / 0.962	0.047 / 0.049 / 0.059	7.951 / 8.097 / 8.581
bunny	26.491 / 24.983 / 27.244	0.941 / 0.928 / 0.935	0.085 / 0.098 / 0.176	7.905 / 7.407 / 7.790
clock	28.943 / 28.682 / 21.972	0.935 / 0.934 / 0.868	0.108 / 0.100 / 0.299	7.810 / 9.026 / 6.833
dog	25.463 / 23.233 / 29.483	0.949 / 0.929 / 0.949	0.085 / 0.100 / 0.143	7.754 / 6.350 / 8.405
horse	31.869 / 31.724 / 25.940	0.982 / 0.980 / 0.878	0.023 / 0.023 / 0.333	8.736 / 8.910 / 5.816
hourglass	27.244 / 27.468 / 27.559	0.970 / 0.976 / 0.930	0.054 / 0.027 / 0.261	2.572 / 3.100 / 7.477
k1 double punch	27.422 / 27.208 / 23.190	0.938 / 0.938 / 0.917	0.070 / 0.068 / 0.202	6.733 / 6.421 / 6.832
k1 handstand	27.631 / 26.795 / 23.178	0.936 / 0.930 / 0.906	0.072 / 0.078 / 0.180	7.233 / 7.377 / 5.945
k1 push up	27.393 / 27.326 / 22.345	0.936 / 0.935 / 0.889	0.072 / 0.072 / 0.230	6.886 / 7.439 / 5.006
music box	32.225 / 32.129 / 24.862	0.980 / 0.979 / 0.871	0.031 / 0.028 / 0.329	8.444 / 8.636 / 5.004
penguin	27.034 / 26.675 / 28.504	0.950 / 0.950 / 0.949	0.074 / 0.068 / 0.165	8.182 / 8.291 / 8.338
plasma ball	33.422 / 36.102 / 29.145	0.944 / 0.955 / 0.857	0.072 / 0.060 / 0.273	7.368 / 6.372 / 5.048
plasma ball clip	46.476 / — / 41.437	0.968 / — / 0.940	0.049 / — / 0.108	8.139 / — / 5.502
red car	30.844 / 30.342 / 28.626	0.961 / 0.960 / 0.962	0.046 / 0.050 / 0.111	8.020 / 8.161 / 8.352
stirling	29.473 / 28.744 / 19.415	0.966 / 0.963 / 0.810	0.045 / 0.037 / 0.372	7.808 / 7.868 / 3.550
tornado	28.629 / 28.825 / 24.427	0.965 / 0.966 / 0.886	0.045 / 0.043 / 0.309	6.398 / 6.815 / 6.164
trex	28.496 / 28.057 / 24.976	0.948 / 0.947 / 0.935	0.056 / 0.058 / 0.148	8.257 / 8.172 / 6.266
truck	31.033 / 30.654 / 30.431	0.969 / 0.967 / 0.973	0.038 / 0.044 / 0.064	8.418 / 8.412 / 8.849
wall-e	28.164 / 27.263 / 25.694	0.934 / 0.923 / 0.936	0.085 / 0.114 / 0.134	7.597 / 7.399 / 7.933
wolf	25.341 / 24.764 / 26.683	0.940 / 0.935 / 0.932	0.089 / 0.094 / 0.183	7.855 / 7.810 / 6.800
world globe	28.529 / 28.175 / 25.560	0.953 / 0.955 / 0.874	0.052 / 0.047 / 0.322	8.063 / 8.243 / 5.372

Table 4. Rendering quality of all dynamic objects using PF I-NGP [5], MixVoxels [6] and K-Planes [2]. PF I-NGP and MixVoxels are evaluated with black backgrounds. K-Planes is evaluated with white backgrounds.

alization results of MixVoxels and K-Planes when trained with different numbers of chunks. Notably, more chunks indicate a shorter temporal length per model and less temporal information. Visualization results of MixVoxels show less noise when the number of chunks increases, while K-Planes' results show the opposite. Hence, the visualization results support that MixVoxels prefers a shorter temporal length per model, while K-Planes prefers a longer temporal length per model.

Spatial Information: From Figure 13, firstly, we notice that the spatial interpolation ability of the neural radiance field is impressive. The visualization of PF I-NGP and MixVoxels are almost the same across three resolutions, but they start missing fine-grained details, such as the stripes of the Bunny's clothes in the lowest resolution setting. This indicates that the performance drop won't happen if the resolution is acceptable for neural radiance fields. Secondly, we found that the reconstruction results of MixVoxels and K-Planes do not miss any parts of the object but are similarly blurry across all three settings. This suggests that current dynamic NeRFs may be biased towards capturing the shape of moving objects, potentially sacrificing the ability to capture details. Finally, the floaters generated from K-Planes start disappearing when the resolution gets lower. This is because K-Planes can revisit the same training samples frequently under the same training setting.

Spatial and Temporal Information: In Figure 14, the reconstruction results of MixVoxels are blurrier (the hand in Scissor) and incomplete (the bear's ears in Horse) on the lowest resolution and more temporal information settings. Although the reconstruction results of K-Planes are slightly blurrier (the hand in Scissor), the reconstructions are complete (the Horse's head) and have fewer floaters with the lowest resolution and more temporal information setting. These match our previous experiment results where we only control one of these factors at a time.

4. Dataset Justification

In this section, we discuss the visualization results of BRICS (*All-view*), BRICS with just two panels (*Forward*), and BRICS with fewer cameras (*Multi-view*) to support that multi-view 360° setting with enough cameras can lead to the most completed reconstruction.

Multi-view 360° Capture System: In Figures 15 and 16, PF I-NGP, MixVoxels, and K-Planes can reconstruct World Globe and Wolf better with *All-view* than *Multi-view*. This indicates that the number of cameras covering the bounding space is significant. PF I-NGP, MixVoxels, and K-Planes can reconstruct World Globe and Wolf better with *All-view* and *Multi-view* than *Forward* regarding the occluded view. Hence, 360° setting is necessary to reconstruct the objects with the correct color completely. Through the visualization results of *Forward*, we observe that MixVoxels tends to render unknown occluded parts with a black background color, while PF I-NGP and K-Planes show better novel view synthesis ability for occluded view.

5. Foreground-Background Segmentation Method

As mentioned in Section 4 of the paper, we use I-NGP for foreground-background segmentation and compare the I-

Scene	PF I-NGP / Mixvoxels / K-Planes PSNR↑	PF I-NGP / Mixvoxels /K-Planes SSIM↑	PF I-NGP / Mixvoxels / K-Planes LPIPS↓	PF I-NGP / Mixvoxels / K-Planes JOD↑
battery	26.828 / 25.805 / 24.351	0.931 / 0.902 / 0.923	0.088 / 0.128 / 0.169	7.483 / 6.729 / 7.404
chess	22.945 / 20.799 / 22.274	0.821 / 0.807 / 0.865	0.215 / 0.233 / 0.267	6.756 / 5.932 / 7.691
drum	24.662 / 23.784 / 22.299	0.903 / 0.894 / 0.880	0.136 / 0.148 / 0.253	7.703 / 6.663 / 7.199
flip book	26.303 / 24.367 / 25.826	0.928 / 0.904 / 0.920	0.120 / 0.150 / 0.198	7.347 / 5.750 / 8.093
jenga	29.683 / 28.884 / 30.992	0.972 / 0.962 / 0.973	0.046 / 0.063 / 0.087	8.583 / 8.281 / 8.284
keyboard mouse	29.635 / 29.153 / 24.734	0.926 / 0.926 / 0.915	0.102 / 0.101 / 0.204	7.808 / 7.777 / 6.910
kindle	29.556 / 28.585 / 25.796	0.958 / 0.951 / 0.943	0.069 / 0.087 / 0.169	8.237 / 7.969 / 6.816
maracas	26.083 / 26.300 / 27.352	0.953 / 0.949 / 0.960	0.072 / 0.081 / 0.085	7.659 / 7.575 / 8.312
pan	27.392 / 26.759 / 24.803	0.935 / 0.918 / 0.937	0.094 / 0.116 / 0.153	7.003 / 6.996 / 7.532
peel apple	27.270 / 27.205 / 26.108	0.939 / 0.942 / 0.934	0.086 / 0.089 / 0.184	7.843 / 7.728 / 7.595
piano	26.824 / 26.097 / 23.879	0.929 / 0.925 / 0.886	0.104 / 0.099 / 0.257	7.719 / 8.438 / 7.013
poker	27.786 / 27.736 / 29.137	0.958 / 0.954 / 0.960	0.062 / 0.068 / 0.095	8.298 / 8.143 / 8.409
pour salt	25.845 / 25.729 / 24.702	0.919 / 0.919 / 0.903	0.104 / 0.111 / 0.225	7.714 / 7.311 / 7.622
pour tea	26.071 / 25.775 / 27.626	0.946 / 0.941 / 0.950	0.089 / 0.094 / 0.144	7.447 / 6.951 / 7.607
put candy	28.189 / 27.360 / 25.870	0.950 / 0.943 / 0.940	0.071 / 0.078 / 0.162	7.906 / 7.836 / 6.737
put fruit	27.129 / 26.614 / 26.546	0.935 / 0.931 / 0.938	0.089 / 0.097 / 0.141	7.815 / 7.389 / 7.456
scissor	25.346 / 25.090 / 25.883	0.944 / 0.937 / 0.936	0.076 / 0.086 / 0.168	7.854 / 7.563 / 7.685
slice apple	26.026 / 25.014 / 26.692	0.951 / 0.940 / 0.939	0.106 / 0.118 / 0.218	7.829 / 7.515 / 7.948
soda	28.780 / 28.727 / 27.115	0.964 / 0.959 / 0.936	0.059 / 0.067 / 0.184	8.322 / 8.091 / 7.704
tambourine	27.985 / 27.624 / 27.482	0.972 / 0.965 / 0.968	0.044 / 0.049 / 0.100	7.634 / 7.455 / 7.227
tea	27.410 / 26.808 / 28.202	0.956 / 0.946 / 0.962	0.073 / 0.088 / 0.099	7.723 / 7.268 / 7.946
unlock	28.649 / 29.275 / 29.967	0.971 / 0.966 / 0.974	0.045 / 0.058 / 0.070	8.158 / 8.243 / 8.441
writing 1	24.086 / 25.145 / 25.630	0.916 / 0.926 / 0.926	0.158 / 0.163 / 0.236	6.725 / 7.789 / 7.222
writing 2	24.345 / 25.613 / 25.782	0.930 / 0.934 / 0.926	0.146 / 0.148 / 0.242	6.490 / 7.983 / 7.604
xylophone	27.334 / 26.951 / 25.449	0.950 / 0.948 / 0.916	0.068 / 0.066 / 0.206	7.950 / 8.103 / 7.094

Table 5. Rendering quality of all dynamic interaction scenes using PF I-NGP[5], MixVoxels[6] and K-Planes[2]. PF I-NGP and MixVoxels are evaluated with black backgrounds. K-Planes is evaluated with white backgrounds.

NGP method to the Segment Anything (SAM) [3]. In this section, we show the strength of the I-NGP segmentation method and its failure case through visualization.

I-NGP Seg. vs SAM: We show multi-view inconsistencies of SAM in Figure 18 and Figure 19. Specifically, we can notice that different views of the same sequence may contain different artifacts, remove certain objects, or miss the boundary of certain objects. For one time step of the Pour Salt sequence, one view contains part of the background, one misses the boundary of the spoon, and one completely removes the spoon. Likewise, in the Replace Battery sequence, one view misses the boundary of the hand, one completely removes part of the object, and a few views include the camera in the background.

Failure Cases: We show failure cases of I-NGP segmentation in Figure 20. Generally, the method misses sections where the object is very small (e.g. the strand of yarn in the Crochet sequence), transparent (e.g. the clear water bottle in the Pour Tea Sequence), highly reflective (e.g. the key in the Unlock sequence), or white (e.g. the paper in the Writing sequence). This is likely because the objects are too similar to the background, making it hard to distinguish between the two. It is also important to note that the lower arm in Pour Tea is missed because we shrink the bounding box slightly smaller than the BRICS machine. This is acceptable because the lower arm is cropped in the 3D space, so it still maintains the multi-view consistency, and we focus on the interaction between objects and hands. This also causes the performance of I-NGP segmentation to be underestimated because we label the whole arms for ground truth. Of course, the segmentation model in our pipeline can always be replaced by another much better method in the future.

6. Dynamic Dataset Distribution

In this section, we elaborate on the dataset distribution of DiVa-360.

Temporal Length: Figure 22 demonstrates the dataset distribution of all data, object data, interaction data, and long-duration data. Overall, DiVa-360 contains dynamic sequences ranging from 5 to 200 seconds. Among them, 37 sequences are 5-29 seconds long, 5 sequences are 30-59 seconds long, 1 sequence is 60-119 seconds long, and 11 sequences are 120-200 seconds long. Our object and interaction dataset contains several long sequences longer than 30 seconds, and our long-duration dataset contains 8 sequences longer than 119 seconds.

7. Other metadata

How can a video play without any audio and captions? Hence, our sequences are accompanied by audio captured through microphones and text descriptions labeled by humans for a better viewing experience. Currently, the audio and text descriptions are only used for a better immersive

Scene	PF I-NGP / Mixvoxels / K-Plane PSNR↑	PF I-NGP / Mixvoxels /K-Planes SSIM↑	PF I-NGP / Mixvoxels / K-Planes LPIPS↓	PF I-NGP / Mixvoxels / K-Planes JOD↑
chess long	28.555 / 29.972 / 23.216	0.901 / 0.930 / 0.885	0.138 / 0.132 / 0.251	7.062 / 7.558 / 6.820
crochet	29.719 / 28.672 / 26.609	0.964 / 0.946 / 0.948	0.065 / 0.095 / 0.178	8.146 / 7.930 / 6.390
jenga long	30.052 / 30.548 / 29.061	0.928 / 0.939 / 0.931	0.085 / 0.083 / 0.165	8.135 / 8.435 / 8.437
legos	27.927 / 28.436 / 25.342	0.887 / 0.901 / 0.890	0.132 / 0.122 / 0.212	6.514 / 7.492 / 6.611
origami	29.694 / 30.191 / 28.028	0.933 / 0.936 / 0.932	0.095 / 0.094 / 0.175	7.796 / 7.955 / 7.740
painting	29.937 / 30.151 / 27.881	0.921 / 0.933 / 0.922	0.118 / 0.112 / 0.292	7.557 / 7.851 / 7.316
puzzle	28.463 / 29.343 / 25.756	0.927 / 0.942 / 0.925	0.115 / 0.110 / 0.204	7.536 / 8.000 / 7.751
rubiks cube	30.436 / 29.649 / 25.668	0.969 / 0.951 / 0.937	0.064 / 0.093 / 0.209	8.024 / 7.776 / 7.112

Table 6. Rendering quality of all long-duration scenes using PF I-NGP[5], MixVoxels[6] and K-Planes[2]. It is surprising that MixVoxels outperforms PF I-NGP in most scenes, with the exception of Crochet and Rubik's Cube. This is because PF I-NGP cannot maintain the temporal consistency for static parts, such as the large chessboard in the Chess Long scene and tray in Jenga Long, Legos, Origami, Painting, and Puzzle scenes, while MixVoxels is good at them.

experience when viewing the sequences. However, we hope this can be extended to other multimodal NeRFs [4] in the future.

Audio Data: BRICS can also act as a multimodality capture system that captures visual and sound data simultaneously. The object sequences and hand-object interaction sequences of the dynamic dataset are accompanied by synchronized spatial audio. The 6 microphones located throughout the capture system allow for 360°audio which provides both loud (e.g. microphones located at the top) and more subtle sounds (e.g. microphones located at the bottom) of the motion. We do not currently have synchronized audio for the long-duration dynamic sequences.

Text Description: The object sequences and the hand-object interaction sequences of the dynamic dataset are accompanied by natural language descriptions at 3 levels of detail. These descriptions are generated entirely by a human annotator without the assistance of any automated tools. The coarsest level aims to capture a broad summary of the scene ("putting candy into a mug"), while finer levels increasingly describe appearance ("...the pieces are in pink, green, orange, and black wrappers..."), relative position ("...candy scattered around a black mug..."), number of hands, audio, and temporal progression. Across the dynamic scenes, the average length of the descriptions is 6.1, 18.4, and 38.7 words for the 3 levels of detail, amounting to a total of 2907 words. We do not currently have natural language descriptions for the long-duration dynamic sequences. To prevent the bias of labeling, we plan to have more people label text descriptions in the future.

8. Visualization Quality

We encourage readers to download the uncompressed data from our website for better visualization quality since some blurriness in images is due to the object's small size relative to the image. For instance, the horse is a hand-size object while the dog is two times larger. This accounts for differences in observable detail between the sequences. We report object sizes in the Table. 3.

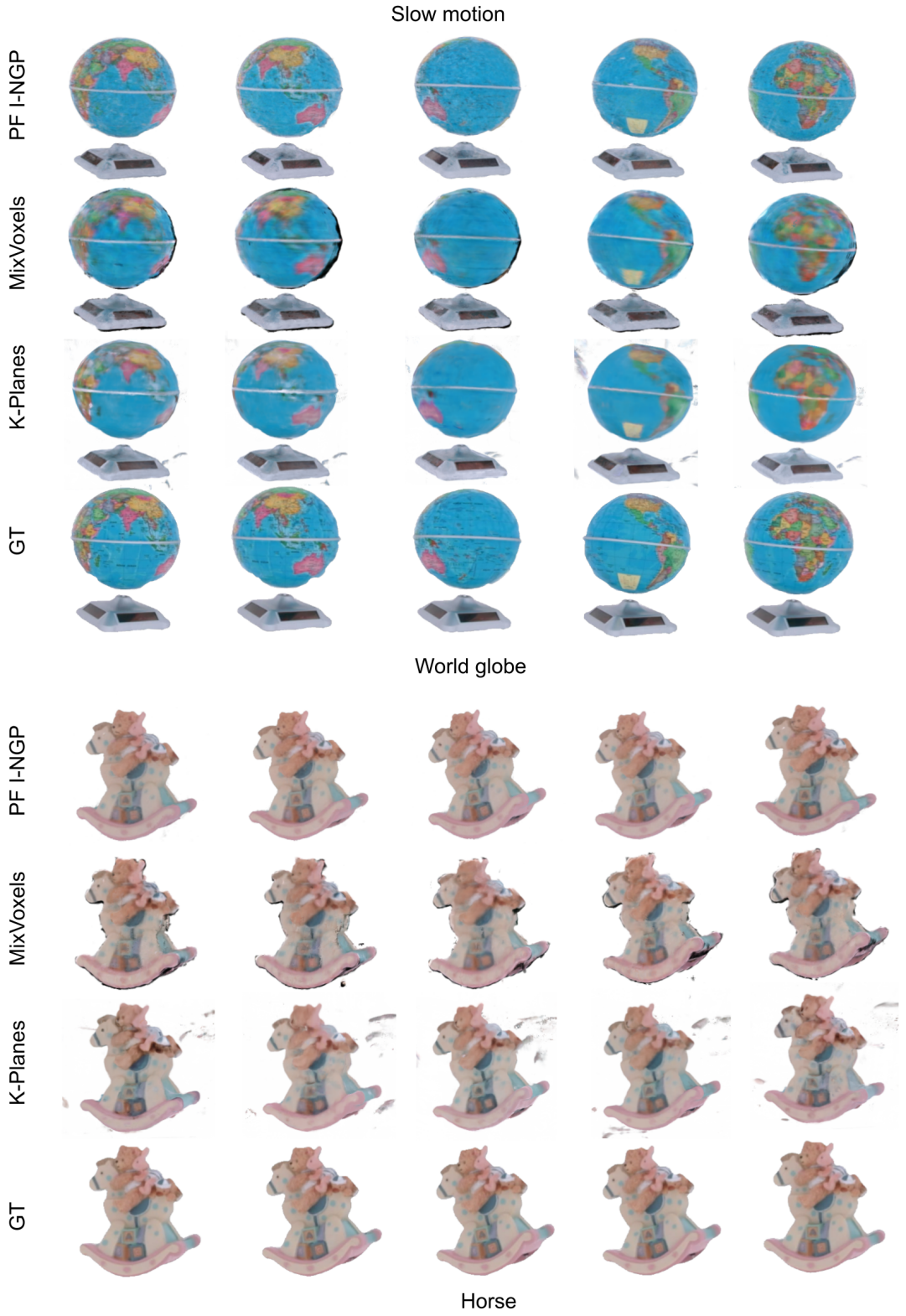

Figure 1. Test view reconstruction and ground truth of the two dynamic objects with slow motion: world globe and horse. Top: Although MixVoxels cannot capture the high-frequency details, the object's shape is correctly reconstructed. K-Planes cannot capture the high-frequency details and construct many floaters around the object. Bottom: The rendering results of MixVoxels are close to the ground truth. K-Planes still suffers from a bunch of floaters.

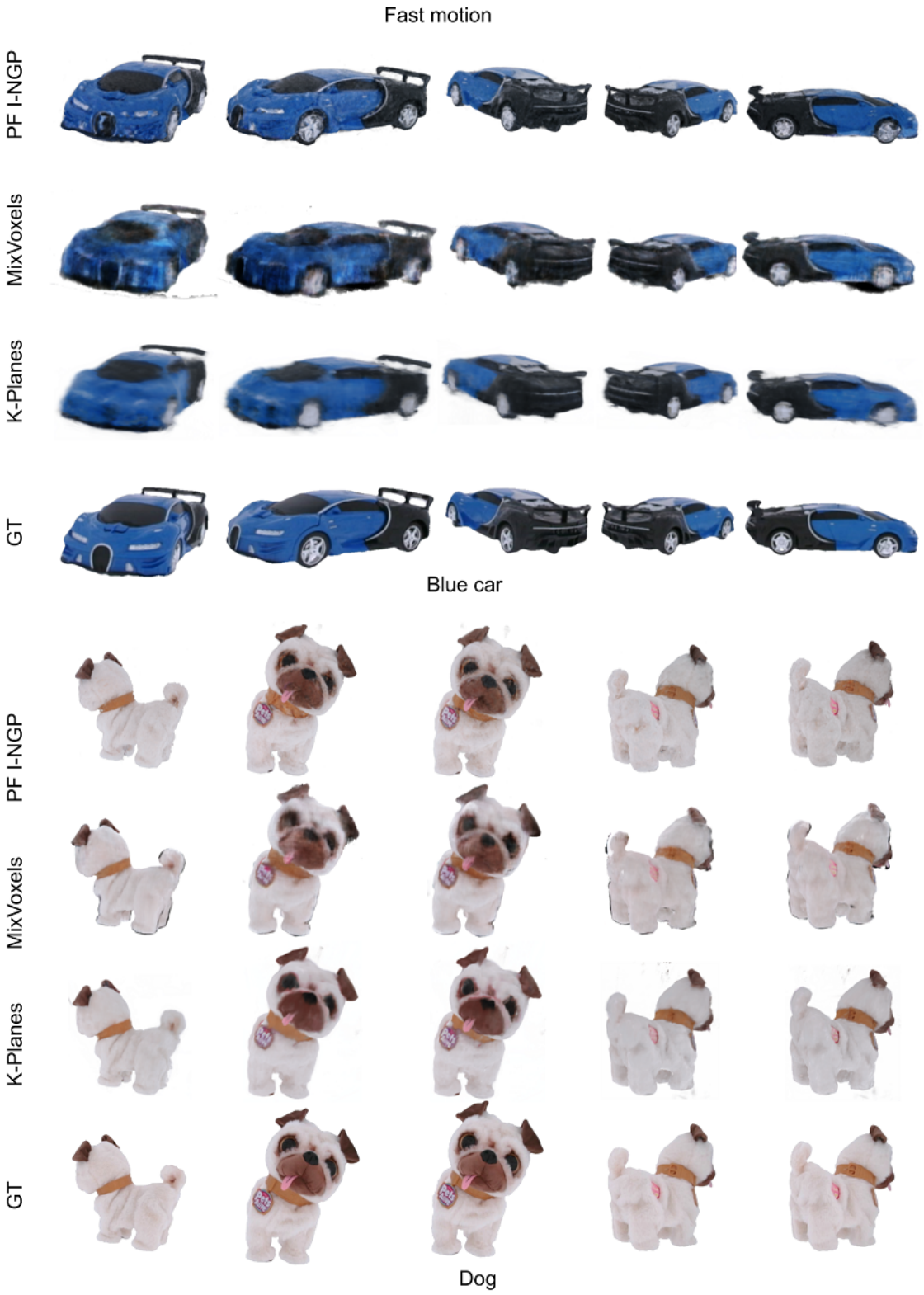

Figure 2. Test view reconstruction and ground truth of two fast motion objects: blue car and dog. Top: The reconstruction results of MixVoxels and K-Planes are blurry. Bottom: The reconstruction results of MixVoxels, such as the dog's tongue, are slightly blurry. K-Planes looks similar to the ground truth.

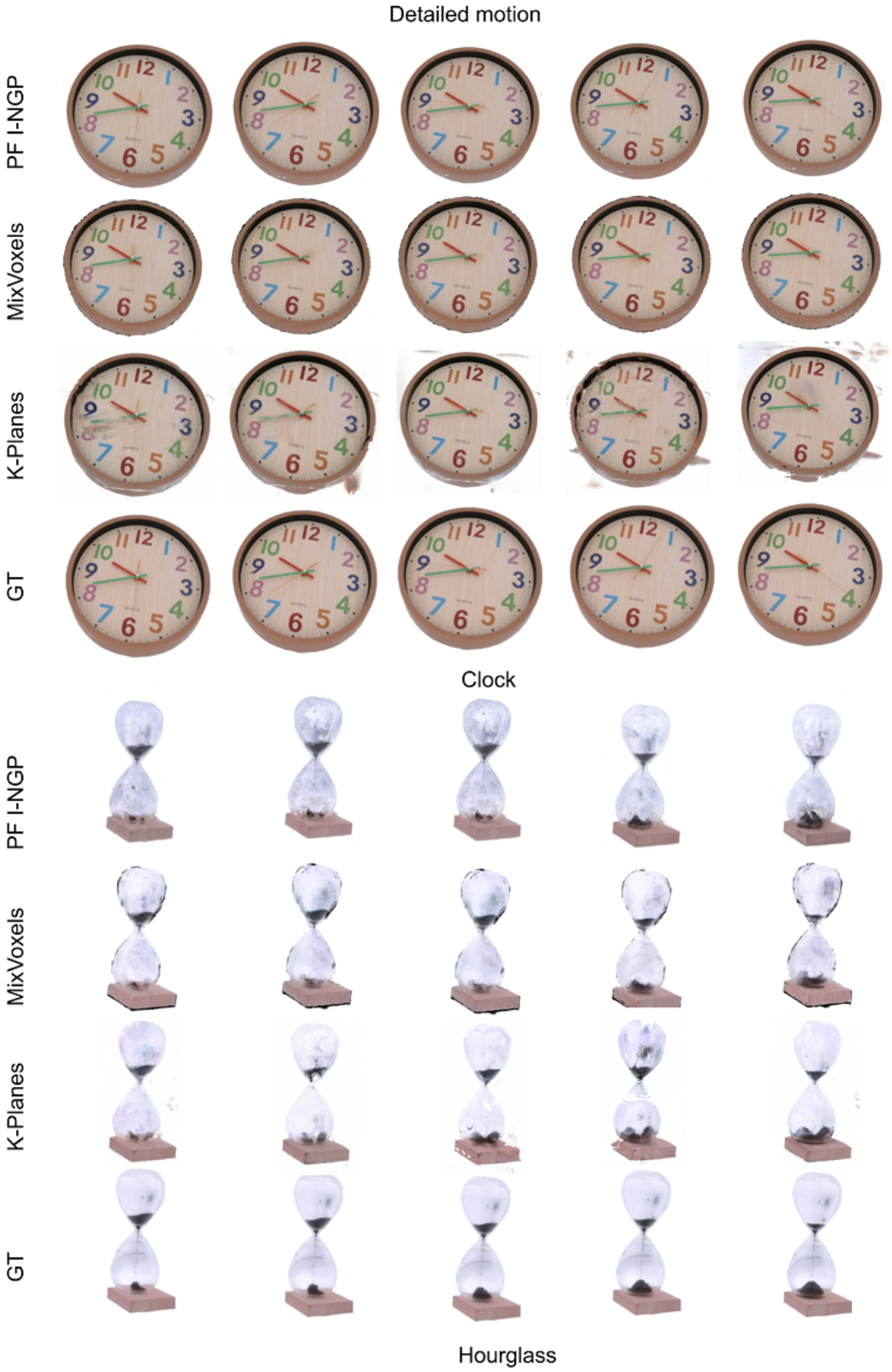

Figure 3. Test view reconstruction and ground truth of two detailed motion objects: clock and hourglass. Top: MixVoxels captures the clock's body accurately but cannot reconstruct the second hand. By contrast, K-Planes struggle for the static part, the clock's body, but the second hand is partially reconstructed. Bottom: All baselines cannot reconstruct the hourglass well due to transparency and highly detailed motion.

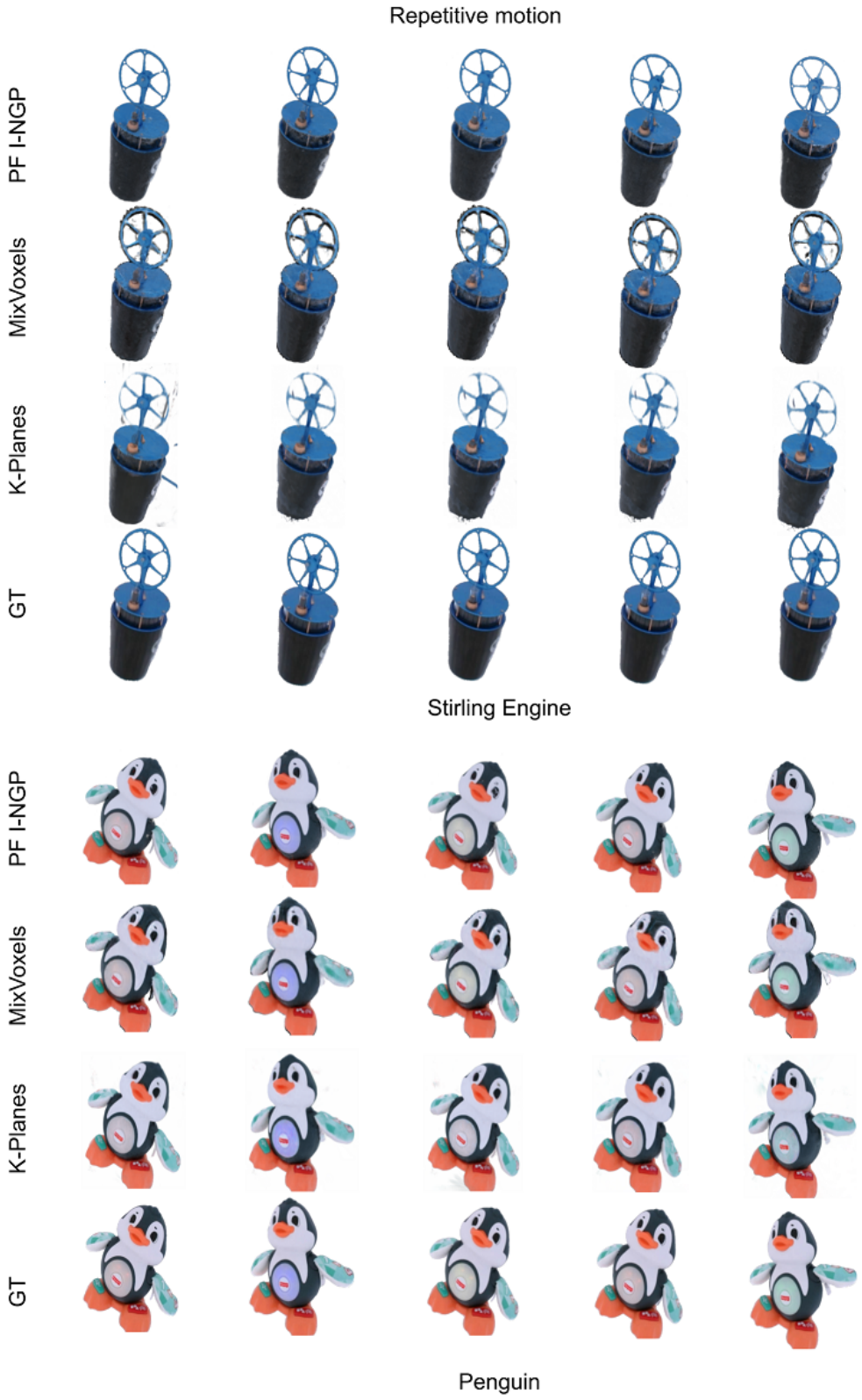

Figure 4. Test view reconstruction and ground truth of two objects with repetitive motions: stirling engine and toy penguin. Top: MixVoxels reconstructs the stirling engine well, while K-Planes fails to capture the rotating part correctly. Bottom: All baselines almost faithfully capture the toy penguin.

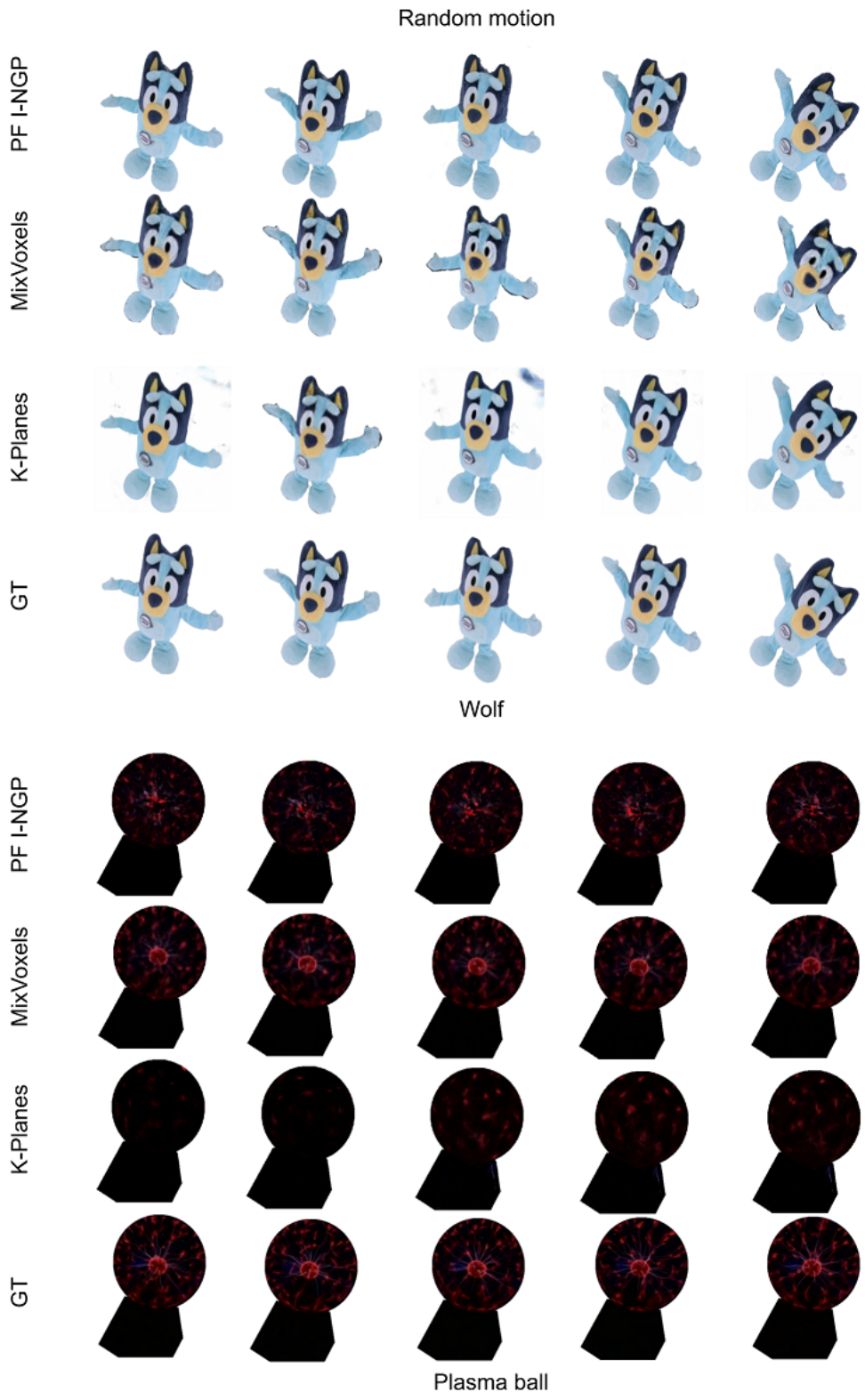

Figure 5. Test view reconstruction and ground truth of two random motion objects: toy wolf and plasma ball. Top: Both MixVoxels and K-Planes capture the toy's motion, but MixVoxels generates some artifacts and sometimes fails to capture parts of the ear and foot of the wolf, and K-Planes contains a few floaters in some frames. Bottom: MixVoxels captures the currents in the plasma ball surprisingly well, while K-Planes and PF I-NGP completely fail. Notably, the plasma ball is captured from a darker environment, so we use the ground truth mask to turn the background into white color for visually pleasing purposes.

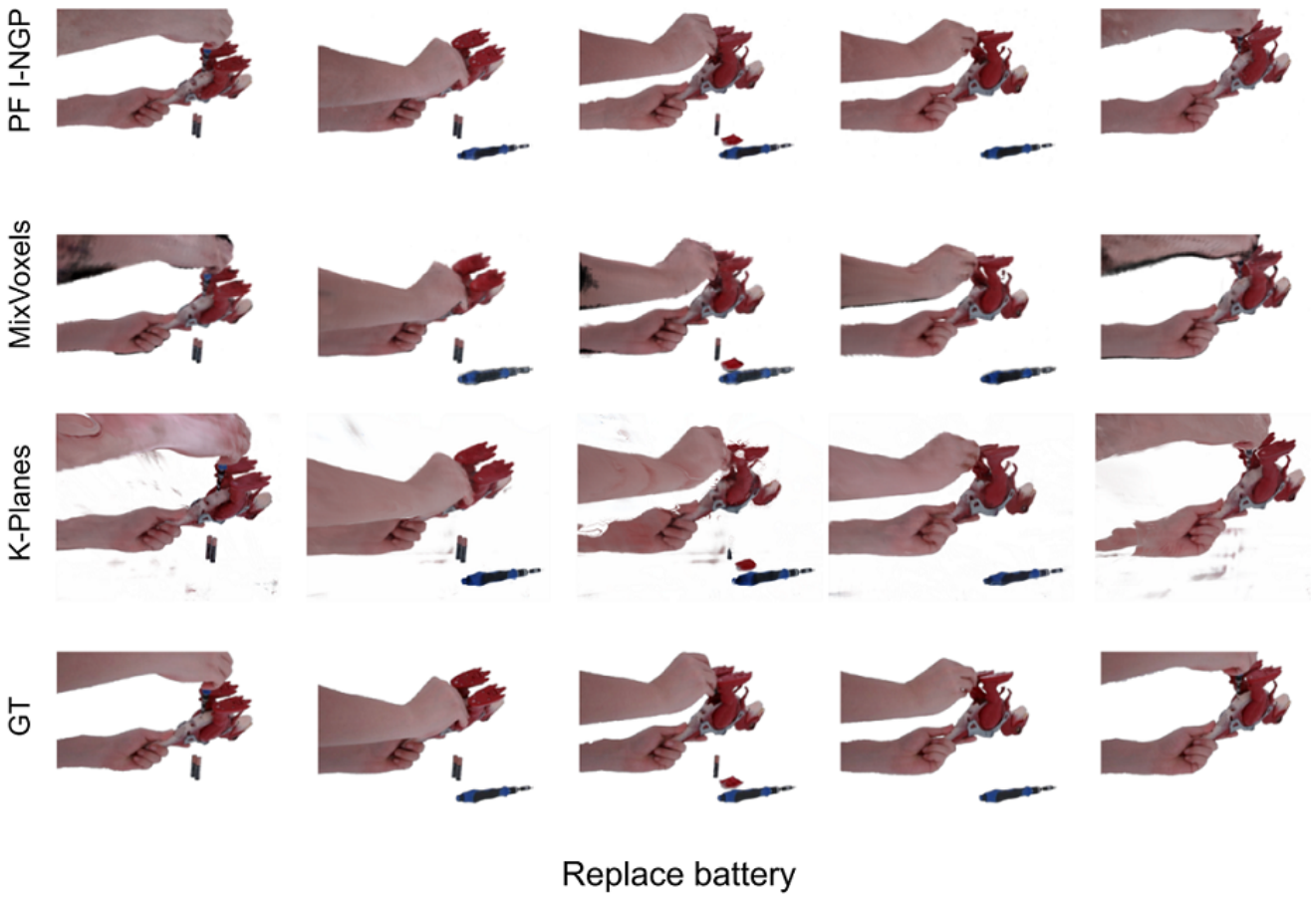

Figure 6. Interaction scene showing the motion of replacing a toy Trex's battery. The sequence contains a series of realistic motions. Both PF I-NGP and MixVoxels correctly reconstruct the scene. K-Planes distorts the hand in the first column and generates many floaters.

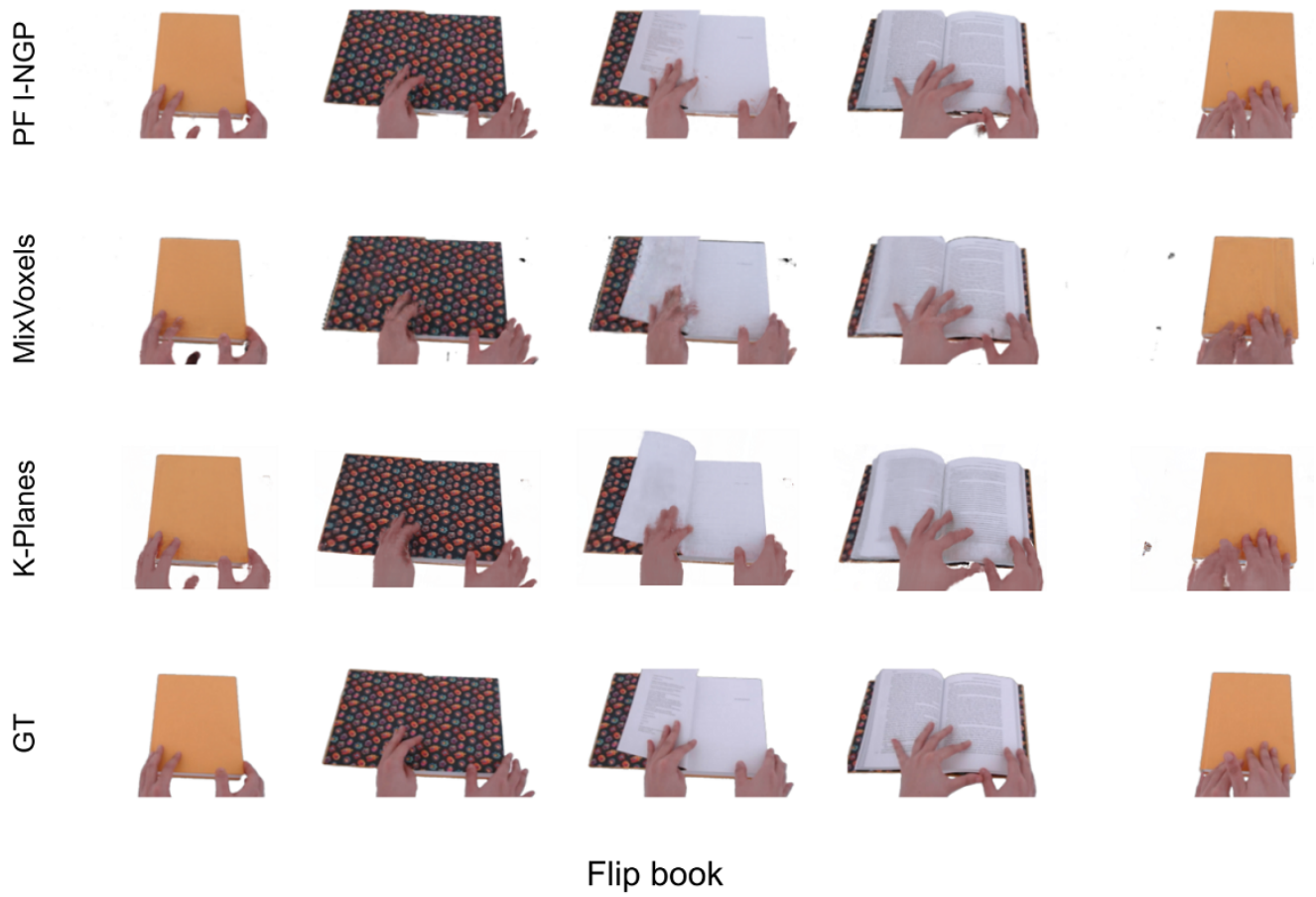

Figure 7. Interaction scene showing the motion of flipping through a book. PF I-NGP generates some artifacts around the bottom of the book in the second and fourth columns. MixVoxels generates blurry book pages in the third column. K-Planes totally misses the texts in the third column.

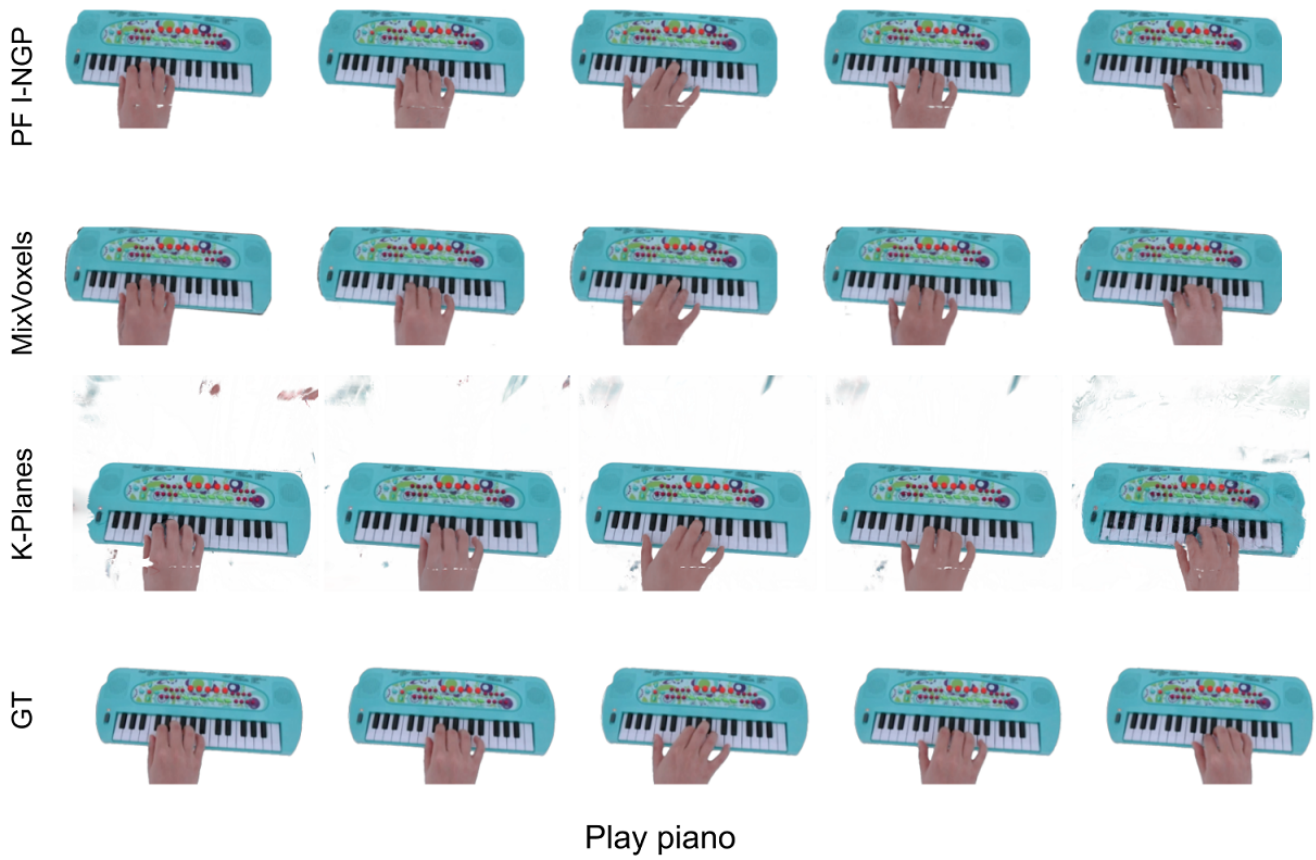

Figure 8. Interaction scene showing a hand playing a toy piano. Both PF I-NGP and MixVoxels capture the motion successfully, but PF I-NGP produces some white scratches on the back of the hand. K-Planes produces white scratches on the back of the hand and floaters in the background.

Figure 9. Interaction scene showing the process of opening a can of soda. MixVoxels generates a few floaters around the hand in the first column. K-Planes generates many floaters in the background.

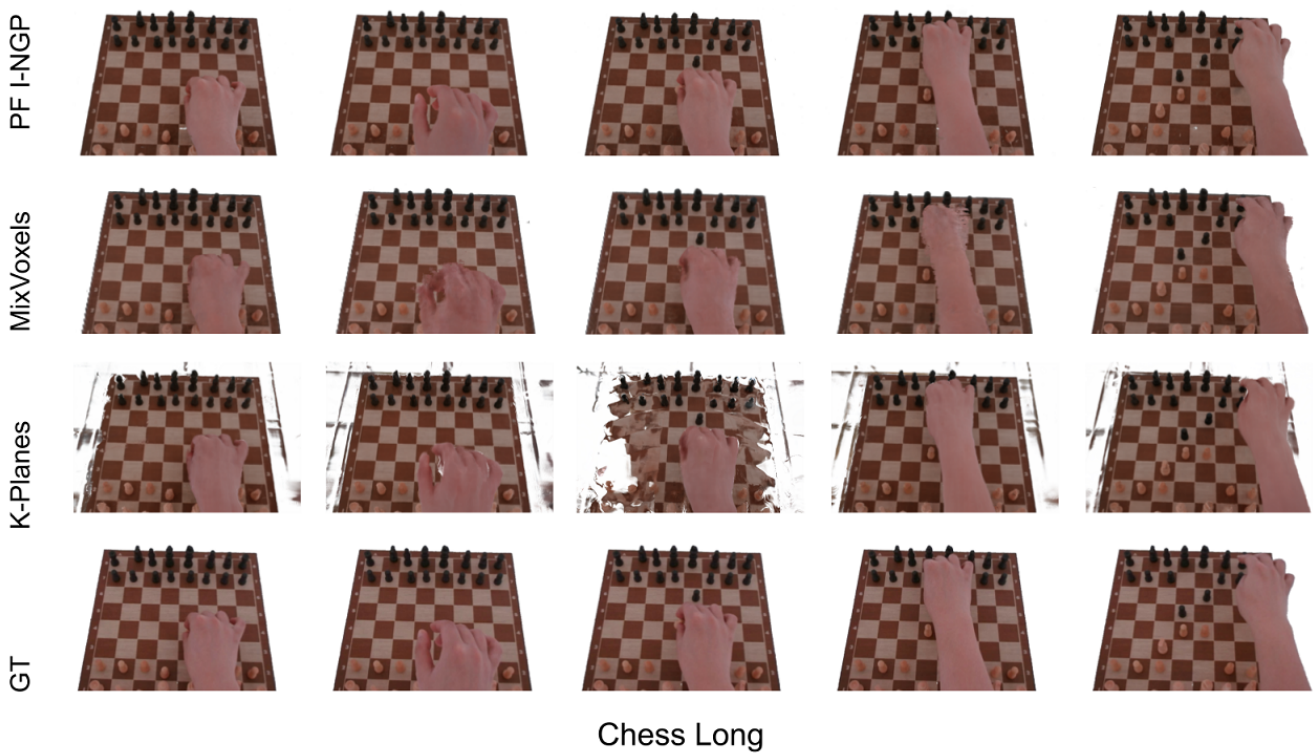

Figure 10. Long-duration scene showing people playing chess. Both PF I-NGP and MixVoxels can reconstruct the chessboard and chess well, while K-Planes struggles at these static parts. For dynamic parts, both PF I-NGP and K-Planes can capture the hand clearly, while MixVoxels cannot.

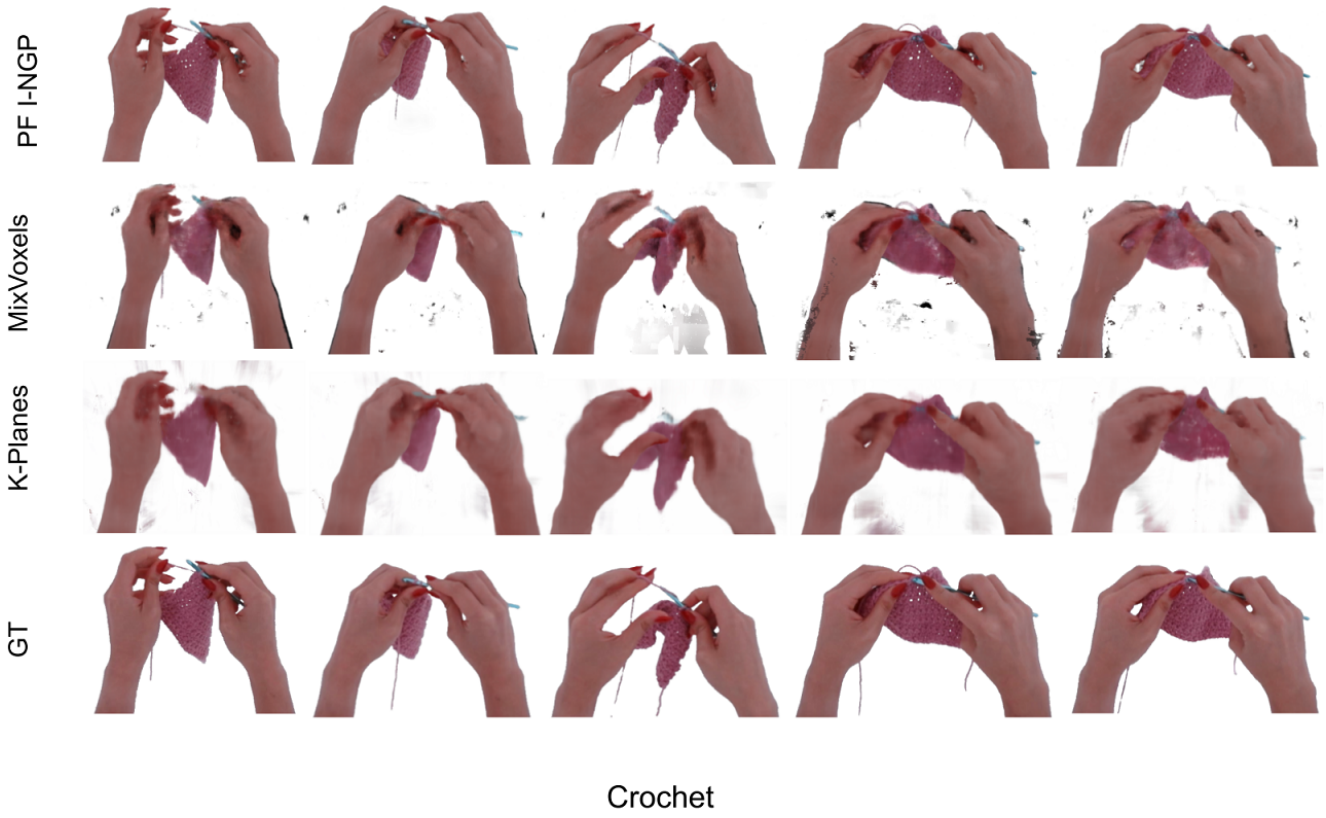

Figure 11. Long-duration scene showing a person crocheting. PF I-NGP can capture the holes in the fabrics, while MixVoxels and K-Planes smooth the fabrics.

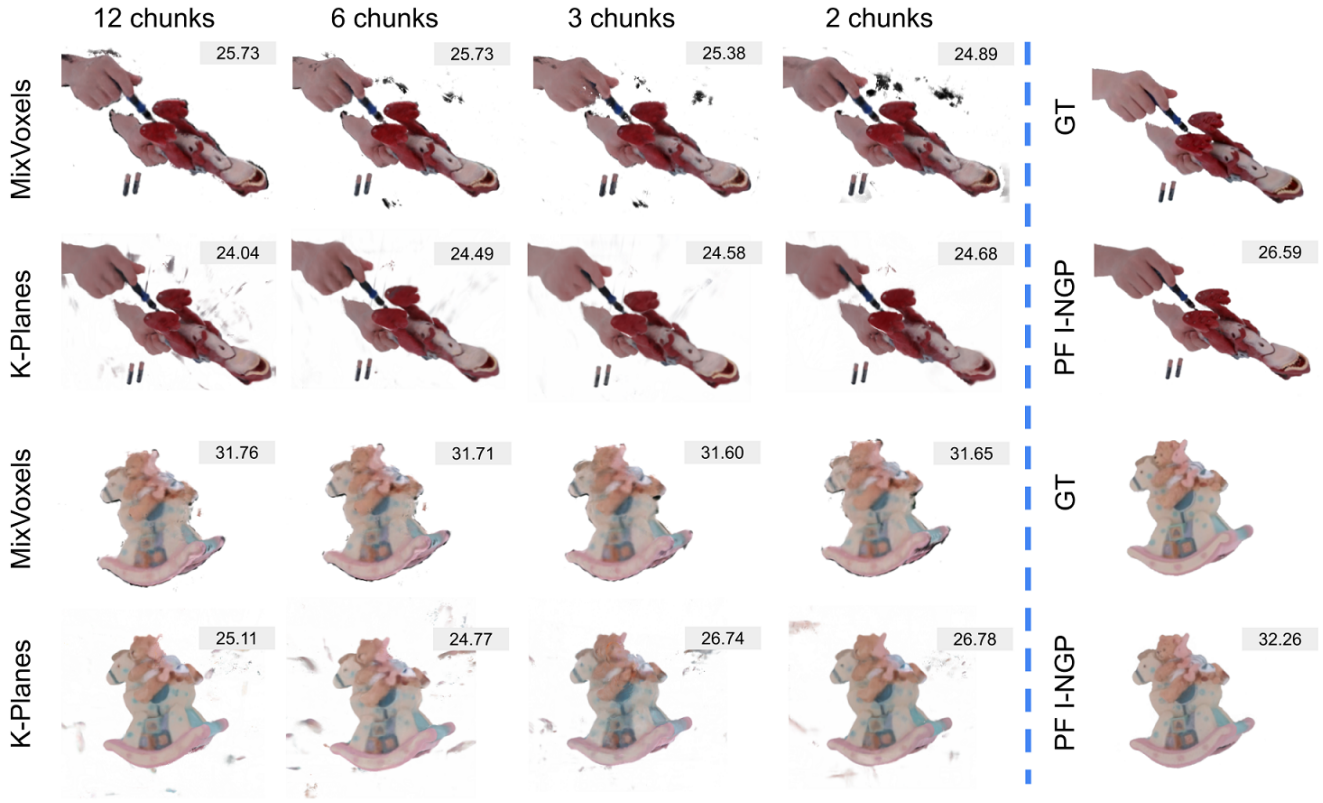

Figure 12. Visualization results of three baselines trained with sequences that split into 12, 6, 3, and 2 chunks. More chunks indicate a shorter temporal length per model and less temporal information. MixVoxels is less noise in Replace Battery, and the bear's eyes in Horse are more apparent when the amount of chunks increases. Unlike MixVoxels, K-Planes generates fewer floaters around the object when the amount of chunks decreases.

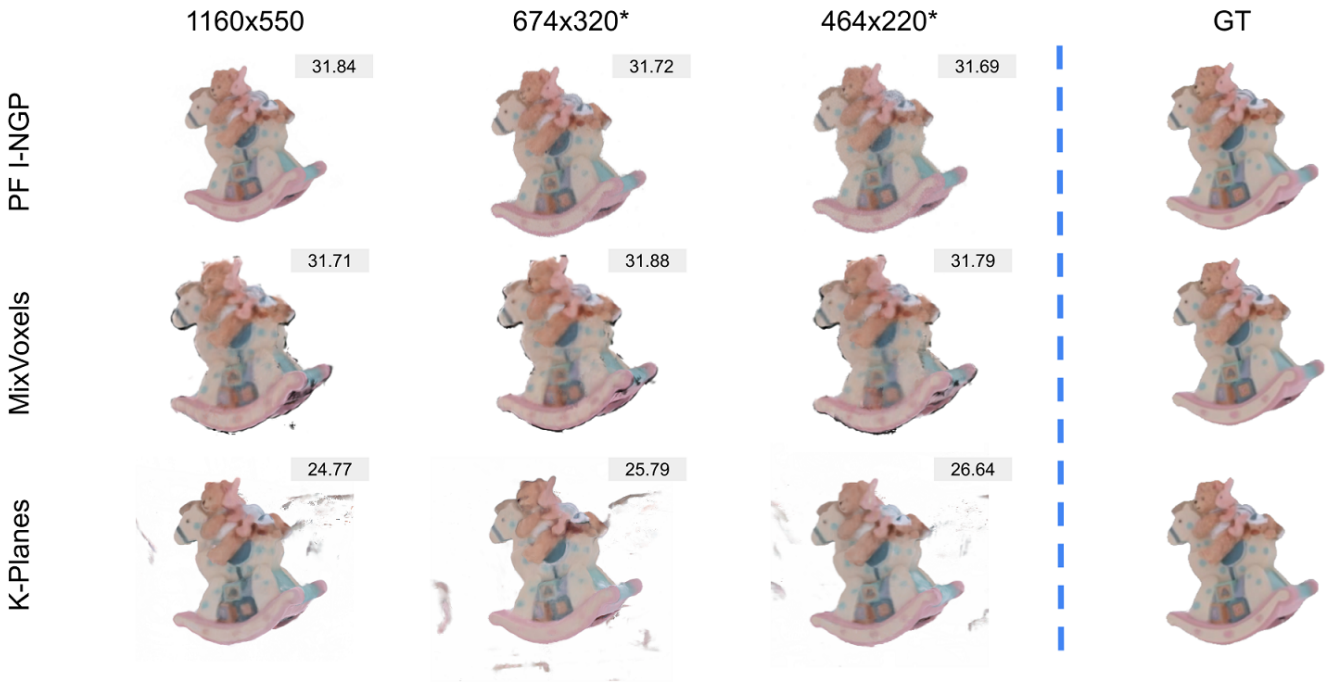

Figure 13. Visualization results of PF I-NGP, MixVoxels, and K-Planes trained with images in different resolutions. * indicates that we spatially interpolate the rendering results to 1160×550 during testing. The visualization results of PF I-NGP and MixVoxels are similar across three settings in Horse, but the stripes of the bunny's clothes are not well reconstructed in 464×200 (better zoom in to see the details). The visualization results of K-Planes contain fewer floaters when the resolution decreases.

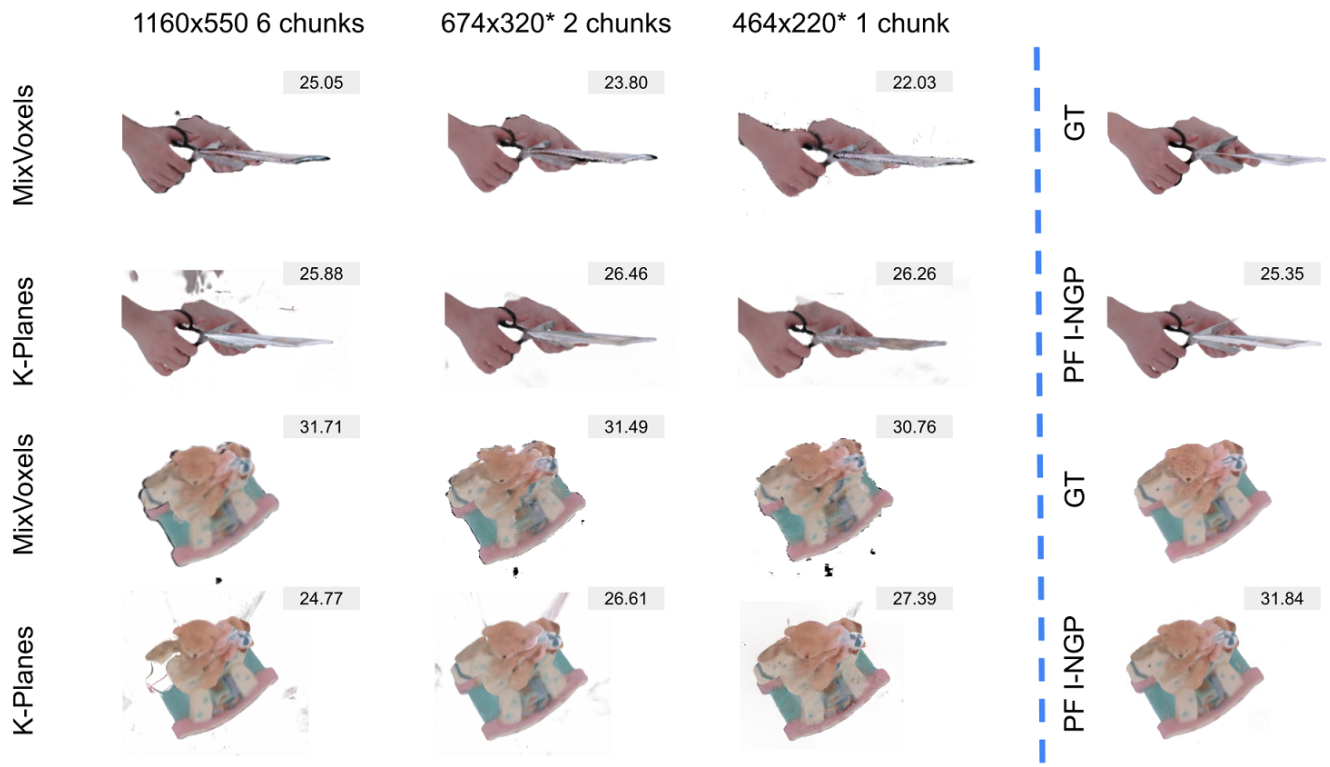

Figure 14. Controlling the spatial resolution and amount of chunks simultaneously does not break the property of MixVoxels and K-Planes.

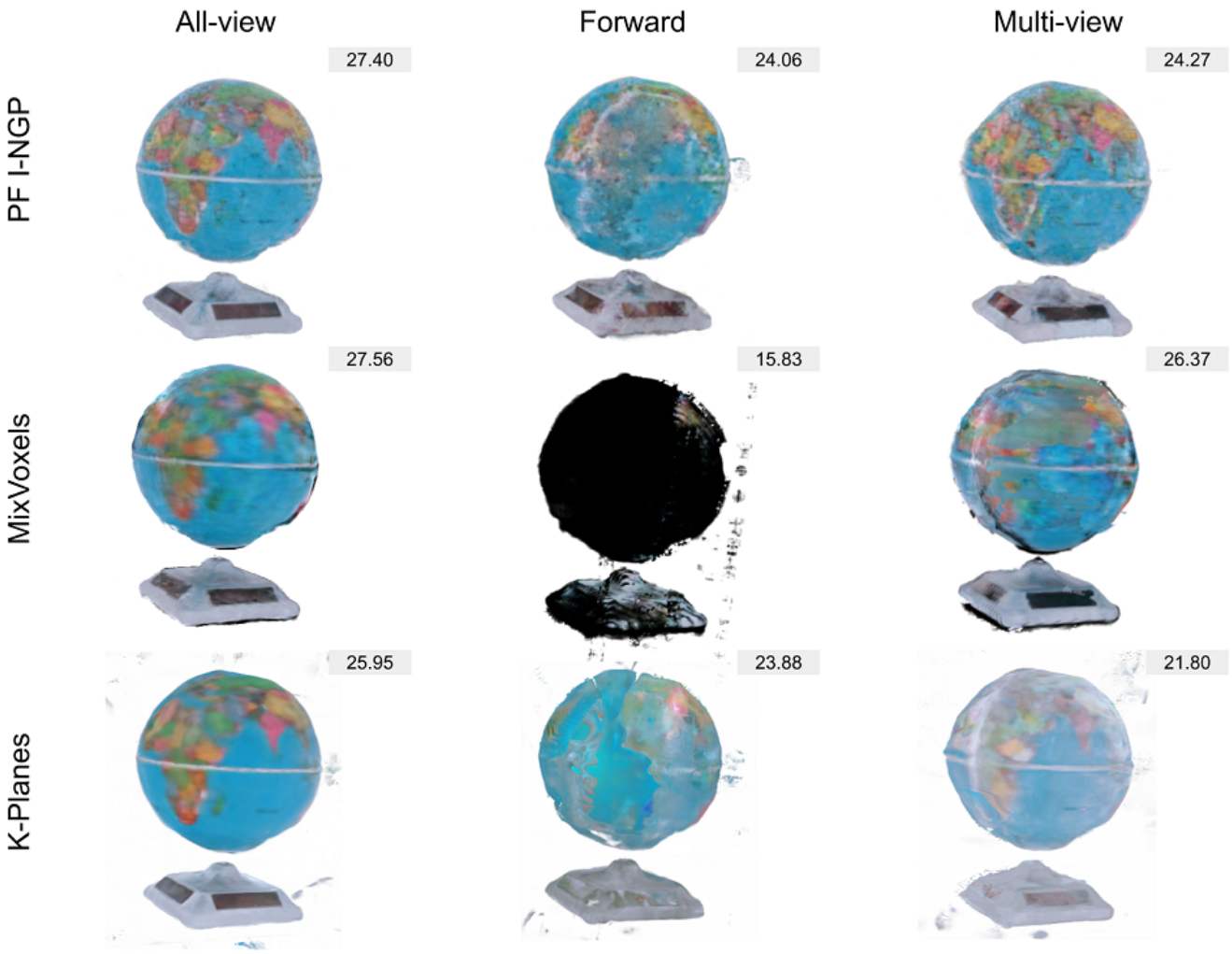

Figure 15. The visualization results of PF I-NGP, MixVoxels, and K-Planes trained with World Globe captured from BRICS (*All-view*), two panels of BRICS (*Forward*), and BRICS with fewer cameras (*Multi-view*). We render the occluded view of *Forward* to demonstrate that the Multi-view 360° setting with enough cameras can provide the most comprehensive reconstruction. Both PF I-NGP and K-Planes can render the occluded view with roughly similar RGB colors, while MixVoxels renders the occluded view with black background color.

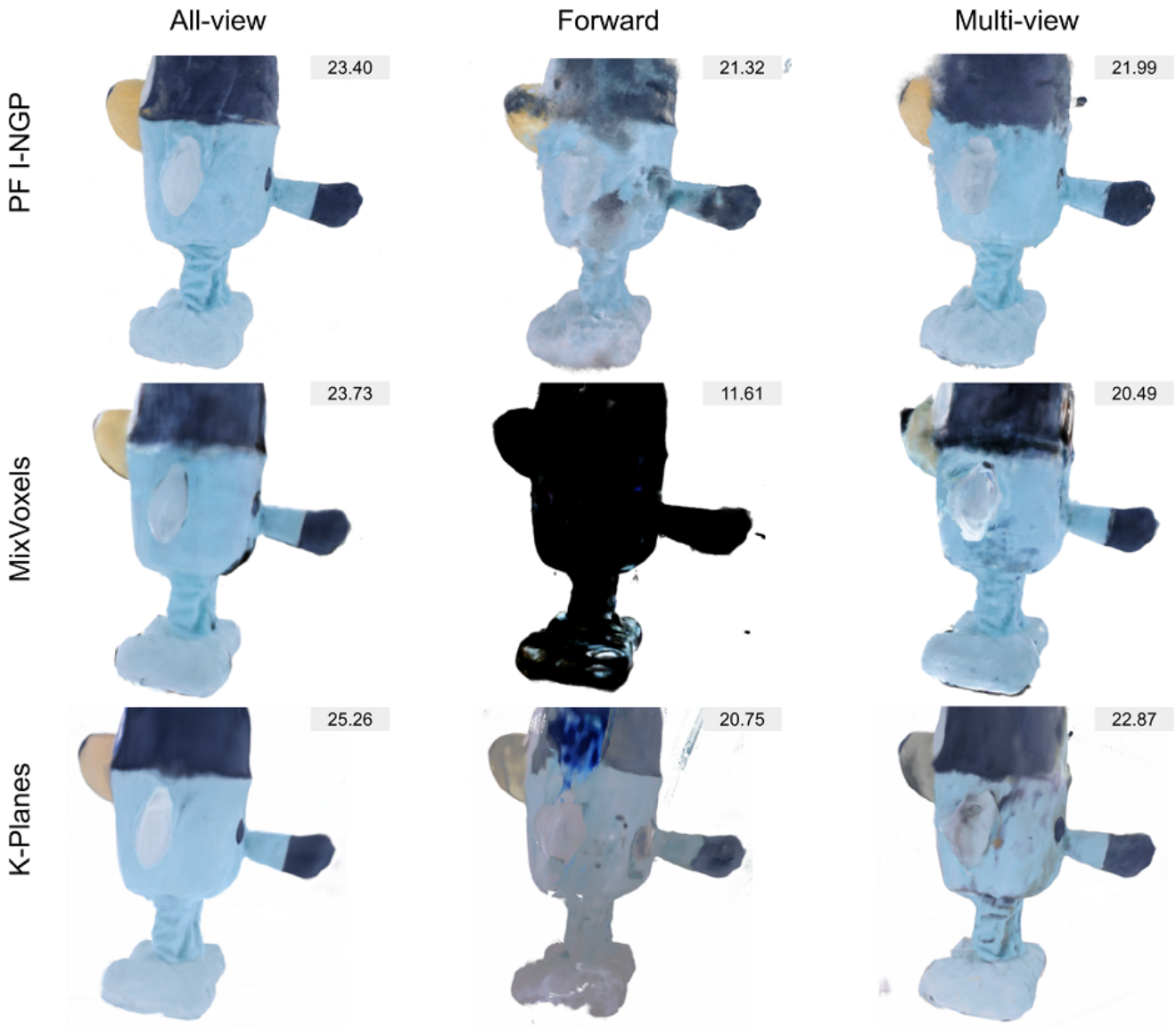

Figure 16. The visualization results of PF I-NGP, MixVoxels, and K-Planes trained with Wolf captured from BRICS (*All-view*), two panels of BRICS (*Forward*), and BRICS with fewer cameras (*Multi-view*). We render the occluded view of *Forward* to demonstrate that the Multi-view 360° setting with enough cameras can provide the most comprehensive reconstruction. Both PF I-NGP and K-Planes can render the occluded view with roughly similar RGB colors, while MixVoxels renders the occluded view with black background color.

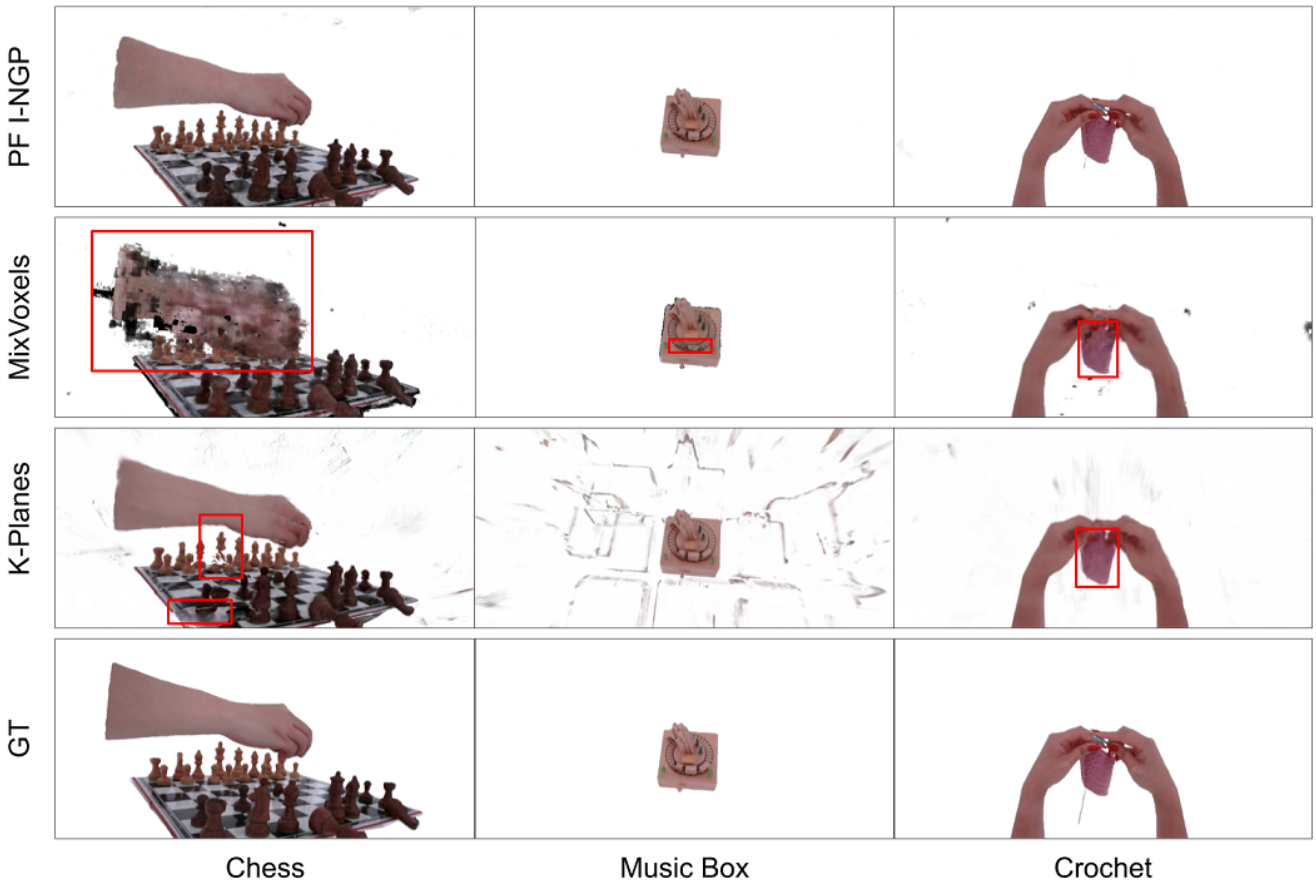

Figure 17. Failure cases of MixVoxels and K-Planes. MixVoxels constructs many floaters around the hand in Chess, fails to reconstruct the small train in Music Box, and the holes of knitted fabric in Crochet. Hence, MixVoxels struggles to capture dynamic parts with complex motion and fine-grained details. K-Planes misses some static parts of Chess, constructs many floaters around objects, especially for slow motion objects such as Music Box, and fills the holes of knitted fabric in Crochet. Therefore, K-Planes struggles to capture static parts and fine-grained details.

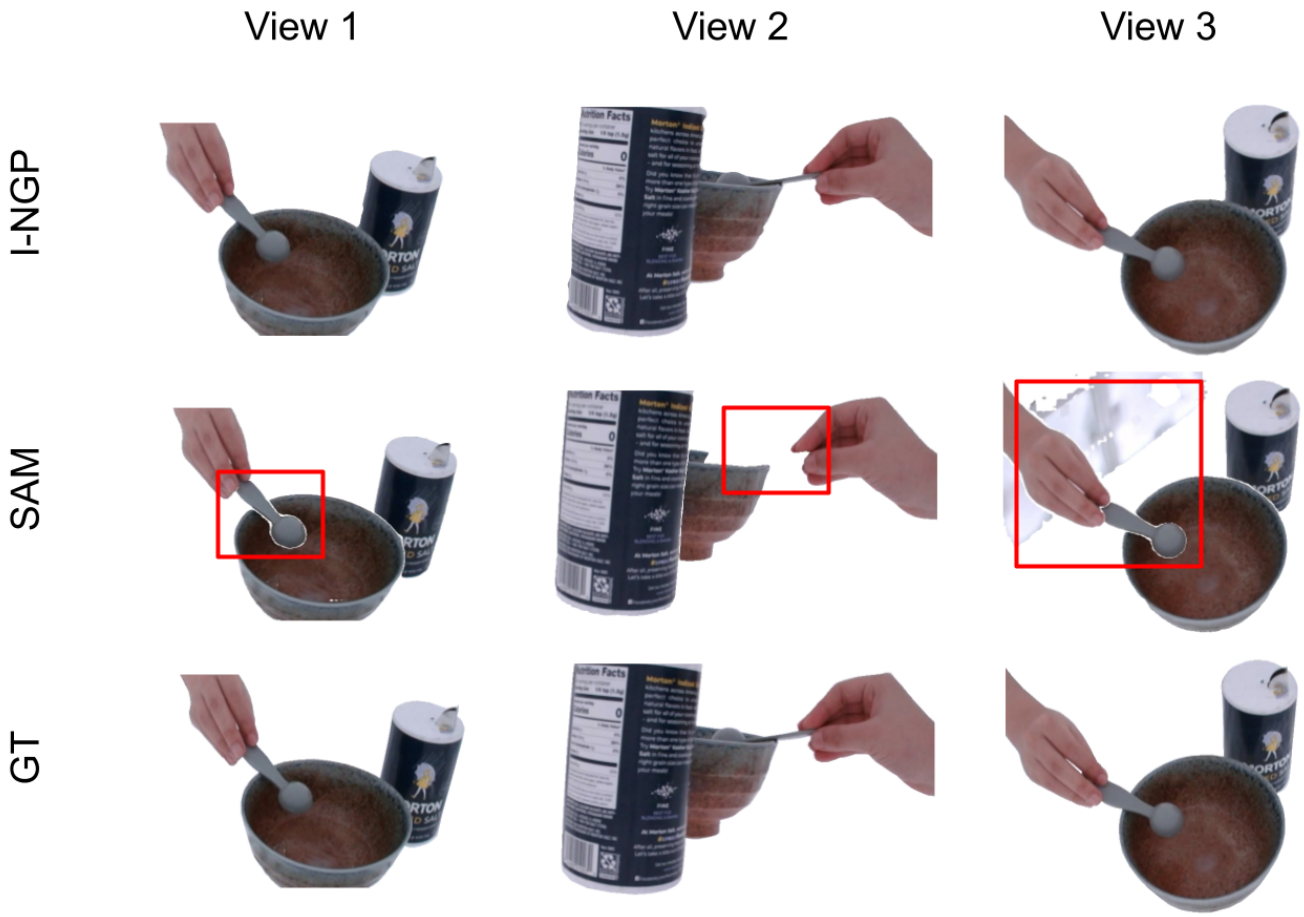

Figure 18. The segmented images of I-NGP and SAM from Pour Salt. SAM cannot maintain multiview consistency, so it contains different artifacts across views. SAM misses the boundary of the spoon in the first view, removes the whole spoon in the second view, and keeps the background in the third view.

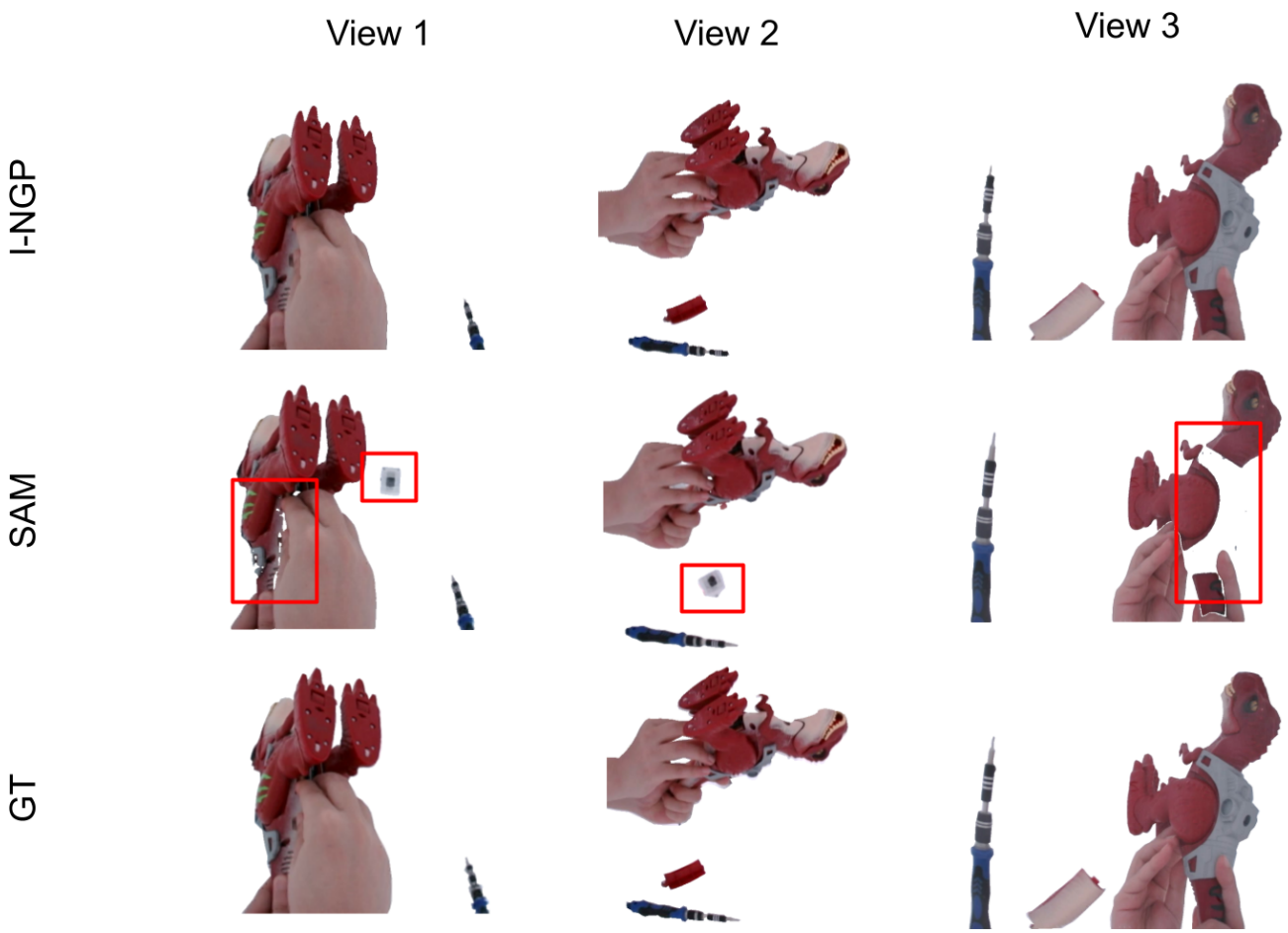

Figure 19. The segmented images of I-NGP and SAM from Replace Battery. SAM cannot maintain multiview consistency, so it contains different artifacts across views. SAM misses the boundary of the hand in the first view, keeps the background in the first and second view, and removes the saddle in the third view.

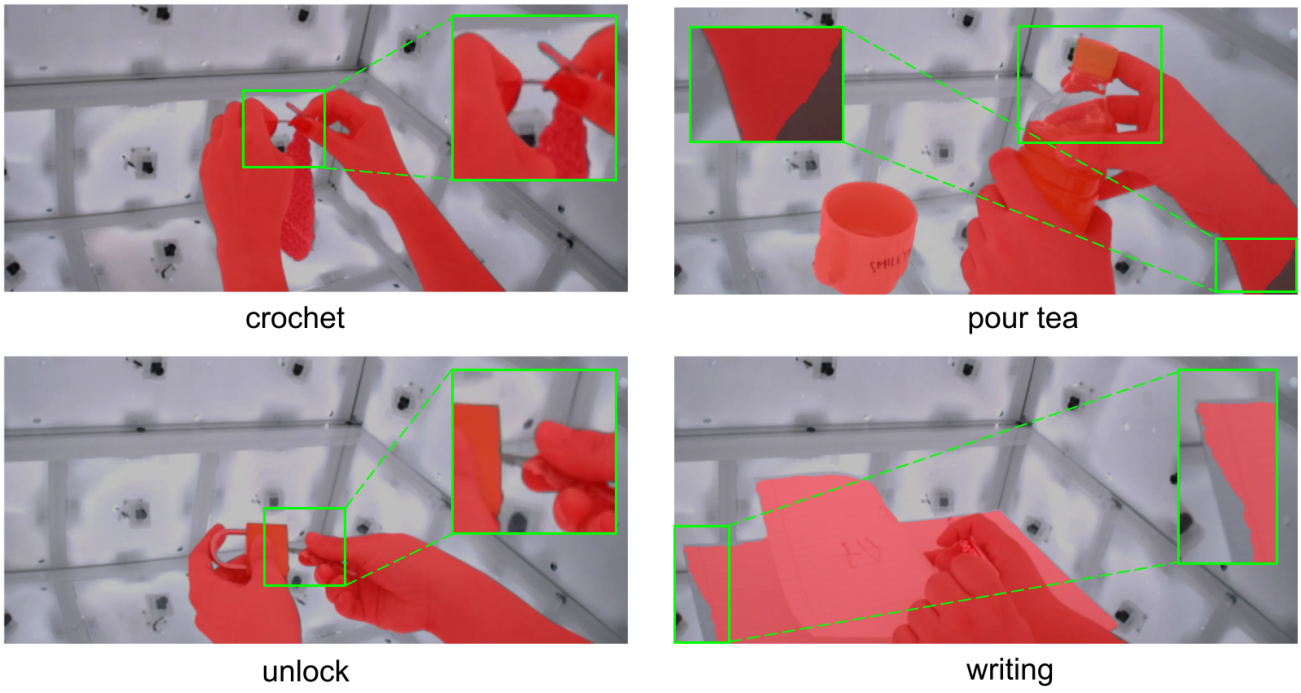

Figure 20. Failure cases of I-NGP segmentation. The performance of I-NGP segmentation is less robust with small thin objects (the yarn in Crochet), transparent objects (the bottle in Pour Tea), high reflection objects (the key in the Unlock), and white objects (the paper in Writing). A part of the lower arm is cut by the bounding box of I-NGP.

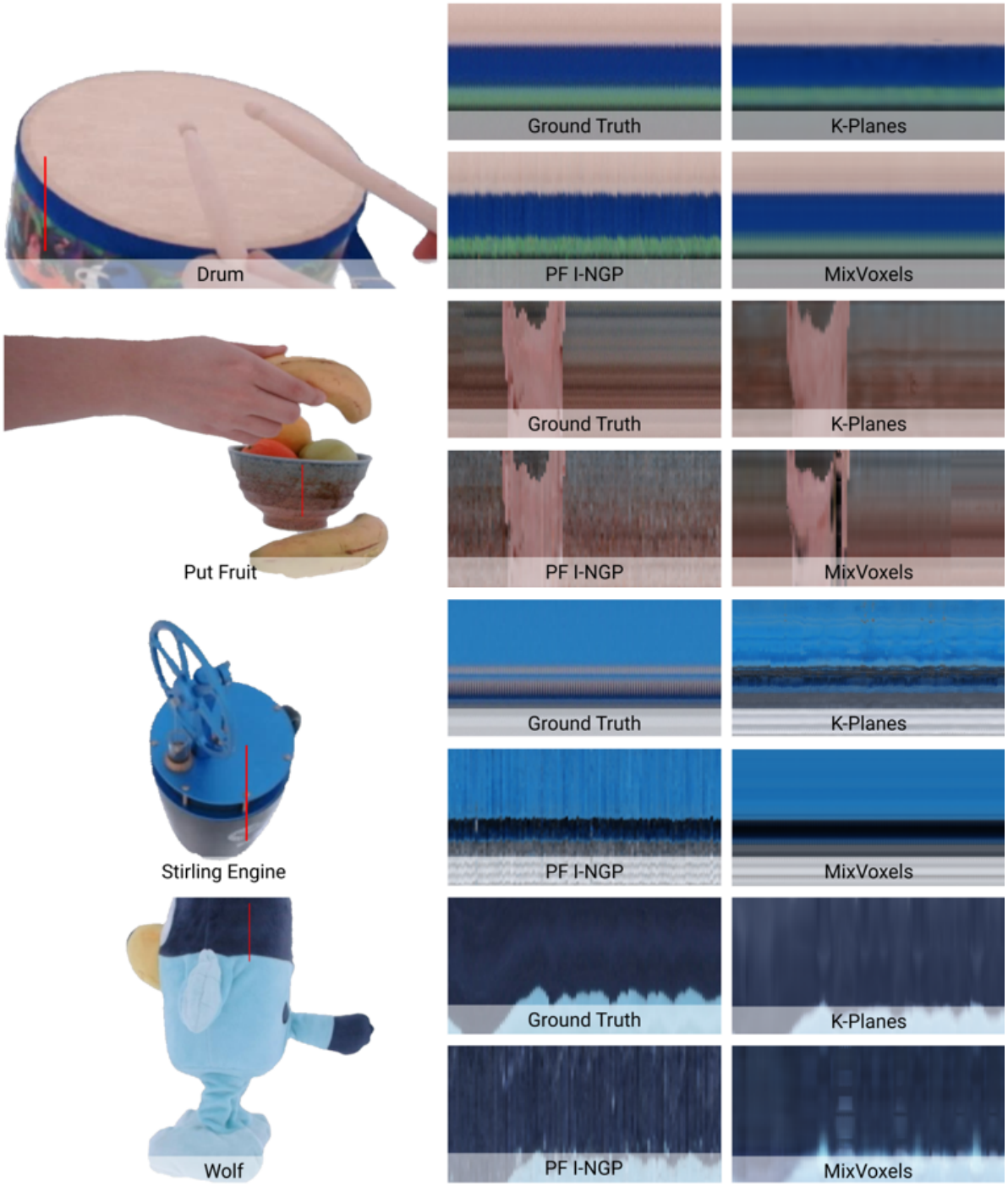

Figure 21. Visualization of temporal consistency by concatenating a line of pixels across frames from the same view. PF I-NGP contains white noise in all four objects, so PF I-NGP is less temporal consistent. Although K-Planes's rendering result is whiter than the ground truth in Put Fruit, Stirling Engine, and Wolf, the noise is smooth across frames. The rendering result of MixVoxels is pretty smooth in Chess and Stirling Engine, but MixVoxels shows black noise on the hand of the Put Fruit and white blocks in Wolf.

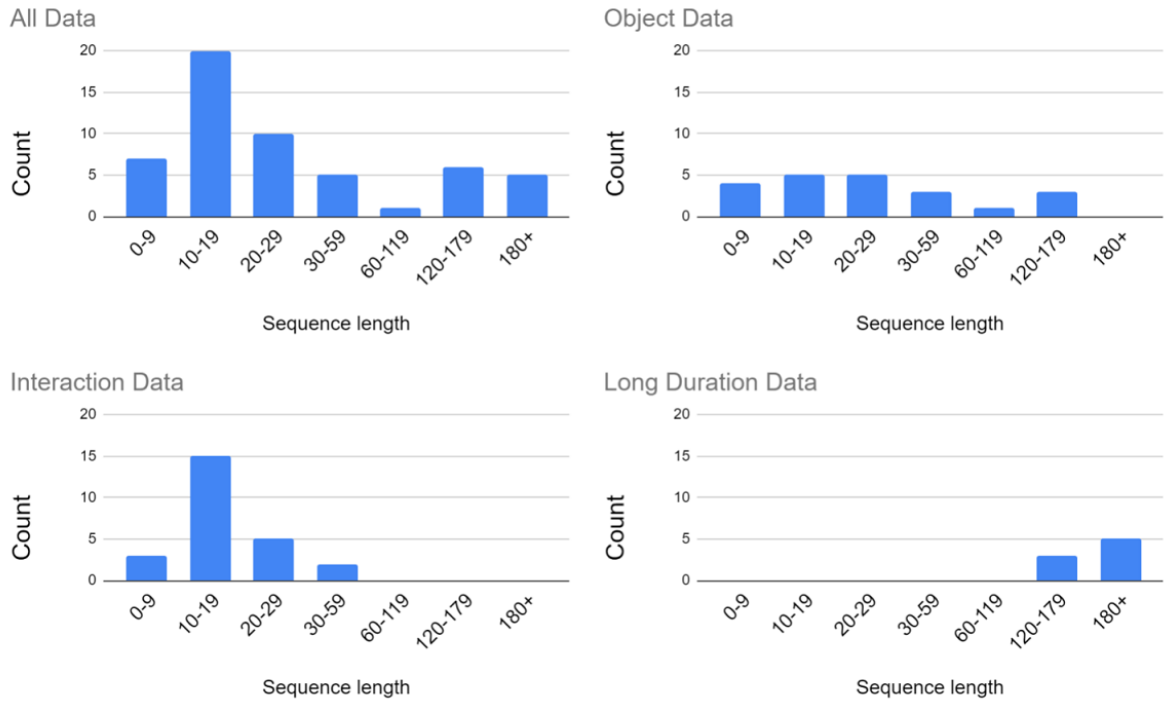

Figure 22. Data distribution in terms of sequence length. Overall, the sequence length of DiVa-360 ranges from 5 to 200 seconds. Both object and interaction datasets have included several long sequences longer than 10 seconds. Our long-duration dataset provides sequences that are at least 120 seconds long.

References

[1] Valts Blukis, Taeyeop Lee, Jonathan Tremblay, Bowen Wen, In So Kweon, Kuk-Jin Yoon, Dieter Fox, and Stan Birchfield. Neural fields for robotic object manipulation from a single image. *arXiv preprint arXiv:2210.12126*, 2022. 4

[2] Sara Fridovich-Keil, Giacomo Meanti, Frederik Rahbæk Warburg, Benjamin Recht, and Angjoo Kanazawa. K-planes: Explicit radiance fields in space, time, and appearance. In *CVPR*, 2023. 2, 3, 5, 6, 7

[3] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick. Segment anything. *arXiv:2304.02643*, 2023. 6

[4] Susan Liang, Chao Huang, Yapeng Tian, Anurag Kumar, and Chenliang Xu. Av-nerf: Learning neural fields for real-world audio-visual scene synthesis. *arXiv preprint arXiv:2302.02088*, 2023. 7

[5] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Trans. Graph.*, 41(4):102:1–102:15, 2022. 1, 2, 3, 5, 6, 7

[6] Feng Wang, Sinan Tan, Xinghang Li, Zeyue Tian, Yafei Song, and Huaping Liu. Mixed neural voxels for fast multi-view video synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 19706–19716, 2023. 2, 3, 5, 6, 7