

## A Supplementary Material

This supplementary material provides (i) additional benchmarking results, (ii) detailed per-predicate performance, (iii) visualizations of scene complexity, and (iv) a detailed explanation of dataset structure and organization, including data format, handling of missing data, and annotation statistics.

Table 4: Ablation on the contribution of egocentric and exocentric inputs to the scene graph generation performance. We report macro F1 scores for two surgical procedures: UI and MISS. Used modalities such as egocentric (Ego) and exocentric (Exo) RGB images, ultrasound screen (Ultra.), audio, point cloud (PC), gaze and hand pose (Hand) are indicated with a checkmark.

Model	Ego	Exo	Ultra.	Audio	PC	Gaze	Hand	UI	MISS	Overall
Ego Only	✓		✓	✓		✓	✓	0.71	0.67	0.68
Exo Only		✓	✓	✓	✓			0.45	0.41	0.42
<b>EgoExOR (Ours)</b>	✓	✓	✓	✓	✓	✓	✓	<b>0.79</b>	<b>0.68</b>	<b>0.72</b>

Table 5: Per-predicate F1-scores for the EgoExOR model. Predicates with zero support in the test split are excluded.

Predicate	anaesthetising	applying	aspirating	looking	closeTo	controlling	cutting	disinfection	dropping	entering	holding	injecting	inserting	lyingon	manipulating	removing	scanning	touching	Macro Avg.
<b>F1-Score</b>	0.02	0.81	0.87	0.81	0.96	0.95	0.30	0.71	0.83	0.60	0.79	0.77	0.77	0.99	0.55	0.78	0.95	0.88	<b>0.72</b>

### A.1 Additional Results

**Ego-Exo Ablation.** In Table 4, we ablate the individual contributions of egocentric and exocentric inputs to surgical scene graph generation. To this end, we train and test two variants of the EgoExOR model, one using only egocentric inputs and one using only exocentric. The results show that models using only exocentric input perform significantly worse, particularly struggling with predicates like *inserting* and *injecting*, which require precise spatial understanding, highlighting the limitations of third-person views alone. While using only egocentric input performances much better, the best performance is achieved when all inputs are used together. These results underscore the

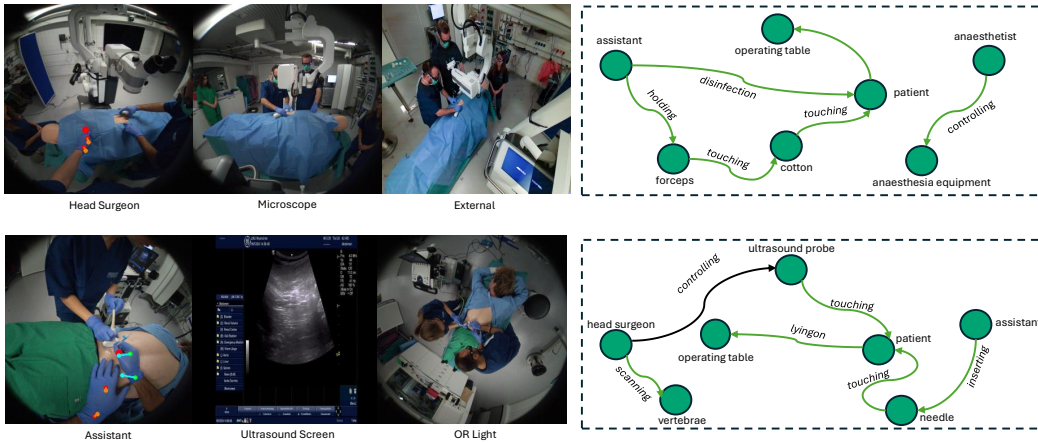


Figure 4: Additional qualitative examples from the EgoExOR model. Correctly predicted entities and predicates are highlighted in green, wrong ones are left white. The "closeTo" predicate is not visualized for brevity.

complementary nature of egocentric and exocentric modalities, with the dual-branch EgoExOR model effectively leveraging both to achieve robust scene understanding.

**Per-Predicate Performance.** In Table 5 we provide detailed per-predicate macro F1 scores for the EgoExOR model, which achieved the highest overall performance (0.72 macro F1). The results reveal significant variability in the EgoExOR model’s performance across different predicates, where the model excels in detecting high-frequency predicates such as *lyingon*, *closeto*, *touching*, and *scanning*, likely due to their prevalence in the dataset and clear visual or contextual cues from combined egocentric and exocentric inputs. Fine-grained actions like *aspirating*, *injecting*, and *controlling* also show strong performance (F1-scores above 0.87), benefiting from the integration of hand pose and gaze data in the egocentric branch, which captures precise tool-hand interactions. However, the model struggles with low-frequency predicates like *anaesthetising* and *cutting*, suggesting challenges in capturing rare events with limited training examples.

**More Qualitative Examples.** In Figure 4 we present two additional qualitative results, showcasing the performance of the EgoExOR model.

## A.2 Dataset Construction and Organization

**From sessions to takes.** EgoExOR comprises 41 fully-processed takes obtained from two emulated operating-room procedures: Ultrasound-Guided Injection (UI) and Minimally Invasive Spine Surgery (MISS). Data acquisition was *continuous*: sensors ran without interruption throughout each emulated procedure, producing nine raw recording sessions. During curation every session was (i) manually segmented at clinically meaningful phase boundaries, (ii) temporally synchronized across all modalities, and (iii) passed through the multimodal-alignment pipeline. The resulting segments are the 41 “takes” referenced in the main paper. The distinction between raw recordings and takes is introduced here for completeness but is not relevant for the dataset usage.

**HDF5 Files.** Each raw session is packaged into a dedicated HDF5 container, totaling nine files, exposed in the public repository. This layout:

- keeps all sensor-specific metadata in the same file as the data they govern.
- offers convenient download granularity: researchers can fetch only the sessions relevant to their study.

To support workflows that prefer working with a consolidated view of the dataset, we provide a helper script that merges the nine session files along with the corresponding split definitions into a single unified HDF5 archive. Additionally, we include dataloaders that facilitate loading and handling of both individual session and unified HDF5 datasets.

Dataset	Shape	Description (dtype)
rgb	[F,S,H,W,3]	Video tensor (uint8). <i>F</i> : frames, <i>S</i> : sources, <i>H</i> : height, <i>W</i> : width.
eye_gaze	[F,E,3]	Gaze array storing [cam_id, x, y] (float32), where E is the number of egocentric sources.
eye_gaze_depth	[F,E]	Gaze depth in meters (float32).
hand_tracking	[F,E,17]	Hand keypoints (8 per hand + camera flag, float32, NaN-padded).
audio/waveform	[samples,2]	Global stereo audio (float32).
audio/snippets	[F,rate,2]	1-second audio snippets per frame (float32, rate=48,000 Hz).
pc/coordinates	[F,P,3]	Point cloud XYZ coordinates (float32). <i>P</i> =2,500 points.
pc/colors	[F,P,3]	Point cloud RGB colors (0-1 range, float32).
annotations	[N,3]	Tokenized triplets in <i>scene_graph</i> (subject, relation, object) (int32). <i>N</i> : annotations per frame. Also available as text in <i>rel_annotations</i> (byte string).

Table 6: Shape conventions for datasets in the HDF5 hierarchy, relative to `data/<surgery_type>/<procedure_id>/takes/<take_id>/`.

Table 7: Annotation statistics for EgoExOR scene graphs, showing counts for entity and relation classes. Entities reflect clinical roles, tools, and objects; relations include actions and spatial interactions.

Entity Class	Count	Relation Class	Count
operating_table	243,664	closeTo	174,044
head_surgeon	198,211	lyingOn	81,918
patient	160,759	touching	79,036
assistant	149,927	holding	55,912
ultrasound_probe	71,757	looking	50,864
needle	45,912	controlling	42,056
anaesthetist	40,158	scanning	35,736
vertebrae	35,656	manipulating	13,430
ultrasound_screen	25,249	injecting	8,137
syringe	22,326	applying	5,576
circulator	19,035	inserting	5,208
instrument_table	18,384	removing	5,180
forceps	16,979	entering	3,125
anesthesia_equipment	12,594	disinfection	2,132
cotton	9,152	dressing	1,342
tissue_paper	6,709	aspirating	1,246
curette	6,502	anaesthetising	1,126
ultrasound_machine	5,195	wearing	544
health_monitor	4,803	cutting	512
microscope_eye	4,548	dropping	478
ultrasound_gel	4,293	positioning	390
antiseptic	4,283	preparing	243
microscope	3,657		
microscope_controller	3,486		
dressing_material	3,212		
operating_room	3,127		
scalpel	2,975		
gloves	2,900		
herbal_disk	2,750		
body_marker	2,219		
scissors	1,868		
instruments	1,724		
microscope_screen	1,084		
bin	789		
tissue_mark	379		
unsterile_instruments	204		

Each session file adheres to a hierarchical structure organized under two primary root groups:

- **metadata** Contains global dataset information, such as vocabulary definitions, camera-to-source mappings, and dataset-level attributes (e.g., version, creation timestamp, and description).
- **data** Contains all sensor data, grouped hierarchically by `surgery_type`, `procedure_id`, and `take_id`. This structure ensures that all time-synchronized data from a single take—across multiple modalities—reside under a unified branch, facilitating efficient indexed access and multimodal alignment.

In the merged dataset, an additional **splits** group is included, containing pre-defined `train`, `val`, and `test` splits. Each split is represented as a structured dataset, with each row specifying a single annotated frame using a 4-tuple identifier: `surgery_type`, `procedure_id`, `take_id`, and `frame_id`. Table 6 summarizes the data keys, their tensor shapes, and datatype conventions used throughout the dataset.

**Missing-data handling.** Sensor streams rarely have identical temporal coverage, e.g. the circulator may enter the OR after the assistant and surgeon, or hand-tracking may be unavailable in some frames adopt strict conventions so that downstream code can distinguish absent signals:

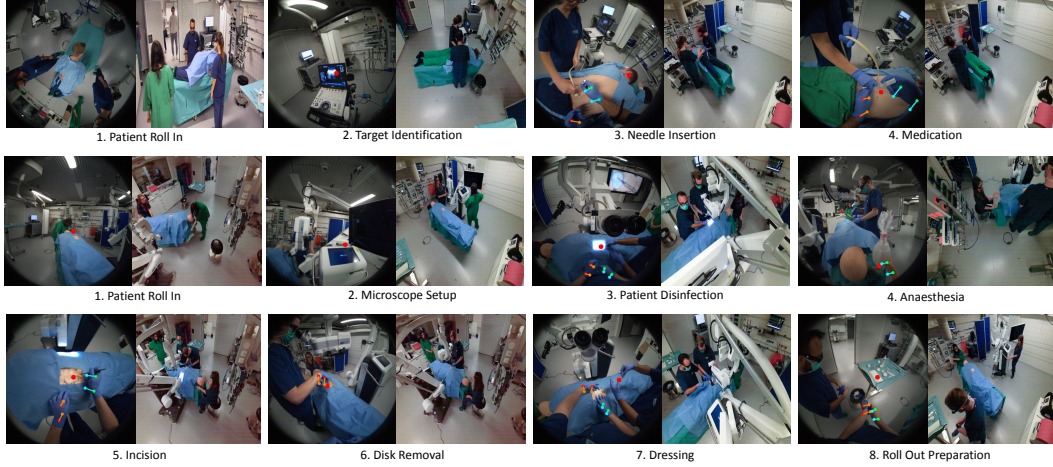


Figure 5: Representative frames illustrating the key procedural steps captured in our two surgical procedures. Each pair of images shows one synchronized egocentric view (left) alongside one exocentric room camera view (right).

- **RGB frames:** Missing frames is zero-filled.
- **Eye gaze:** Invalid points are encoded as  $(x, y) = (-1, -1)$ , which also facilitates filtering of gaze depth values.
- **Hand tracking:** Key-point coordinates are stored as NaN when confidence  $< 0.1$ .
- **Audio:** Missing audio samples are zero-filled.
- **Point Cloud:** Missing point cloud data is represented as an empty point cloud.

### A.3 Annotation Statistics

In Table 7, we provide exact counts for each entity and predicate class in the dataset, revealing the dataset’s complexity and class imbalance. For example, frequent entities like *head\_surgeon* and relations like *closeTo* dominate, while rare entities *tissue\_mark* and relations *checking pose* detection challenges. Additionally, in Figure 5 we visualize some of the key steps in the dataset, to provide a better visual overview. Finally, we also include a short video to better showcase different aspects of EgoExOR.