2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 14-18, 2024. Abu Dhabi, UAE

Indoor Scene Change Understanding (SCU): Segment, Describe, and Revert Any Change

Mariia Khan[1], Yue Qiu[2], Yuren Cong[3], Bodo Rosenhahn[4], David Suter[5] and Jumana Abu-Khalaf[6]

***Abstract*— Understanding of scene changes is crucial for embodied AI applications, such as visual room rearrangement, where the agent must revert changes by restoring the objects to their original locations or states. Visual changes between two scenes, pre- and post-rearrangement, encompass two tasks: scene change detection (locating changes) and image difference captioning (describing changes). While previous methods, focused on sequential 2D images, have addressed these tasks separately, it is essential to emphasize the significance of their combination. Therefore, we propose a new Scene Change Understanding (SCU) task for simultaneous change detection and description. Moreover, we go beyond change language description generation and aim to generate rearrangement instructions for the robotic agent to revert changes. To solve this task, we propose a novel method - EmbSCU, which allows to compare instance-level change object masks (for 53 frequently-seen indoor object classes) before and after changes and generate rearrangement language instructions for the agent. EmbSCU is built on our Segment Any Object Model (SAOMv2) - a fine-tuned version of Segment Anything Model (SAM), adapted to obtain instance-level object masks for both foreground and background objects in indoor embodied environments. EmbSCU is evaluated on our own dataset of sequential 2D image pairs before and after changes, collected from the Ai2Thor simulator. The proposed framework achieves promising results in both change detection and change description. Moreover, EmbSCU demonstrates positive generalization results on real-world scenes without using any real-life data during training. The dataset and the code are available here.**

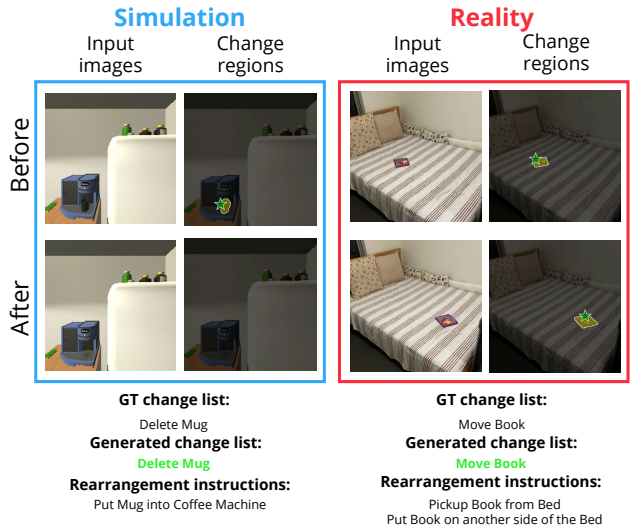

Fig. 1. EmbSCU performance for the SCU task. Given before-after image pairs of a simulated (left) or a real-life scene (right), EmbSCU, utilizing SAOMv2 - our fine-tuned SAM version, generates instance-level change object masks. It also provides change language descriptions and rearrangement instructions for the robotic agent to revert changes. EmbSCU demonstrates promising generalization results on the real-world scenes without using any real-life data during training. Best view in color and zoom in.

I. Introduction

Identification of temporal and spatial scene changes is an essential task for various real-life applications such as remote-sensing change detection (RSCD) [1]–[3], street scene change detection (SSCD) [4]–[6] and damage detection [7], [8]. Moreover, accurate recognition of scene changes is crucial for embodied AI tasks, where an environment is constantly changing due to the agent's actions over time. Information about changes in a scene is especially important for the visual room rearrangement task [9]. In this task, the aim for an agent is to revert the changes: restore locations or states of the changed objects.

Current works, related to scene change understanding, can be divided into scene change detection (SCD) and image difference captioning (IDC). The aim of SCD is to recognize pixel or point level changes in 2D images, or 3D point clouds. Existing SCD methods [1]–[6] only focus on identifying the "changed region", but do not provide any information about the change. To manipulate objects, the agent should recognize the change type (e.g. "open" or "add") and the changed object class (e.g. fridge or book). Thus, SCD methods are not sufficient to successfully tackle the rearrangement task's scene change detection stage. On the other hand, the existing methods for image difference captioning (IDC) generate only language descriptions for scene changes and do not identify the exact region of the change [10]–[13]. Usually centered on single-camera viewpoints, these methods are not suitable for the rearrangement task, where the robotic agent captures images from multiple viewpoints along its path.

To address the above-mentioned limitations, we propose a novel task of *Scene Change Understanding (SCU)*. The aim of SCU is to locate and describe the changes between two sets of sequential 2D images, taken along the path of the robotic agent. We work with 2D data instead of the 3D point clouds, as indoor real-life environments are usually equipped with simple surveillance cameras instead of LiDAR sensors. In SCU we focus on 3 main change aspects: change type, changed object class, and changed location (region) within the 2D images. We go beyond change language description generation (such as in the IDC task) and aim

[1,5,6]Edith Cowan University, School of Science, Centre of AI and Machine Learning, Australia mariia.khan@ecu.edu.au
[2]Artificial Intelligence Research Center, National Institute of Advanced Industrial Science and Technology, Japan qiu.yue@aist.go.jp
[3,4]Institute for Information Processing, Leibniz University of Hannover

to generate rearrangement instructions for the robotic agent. The instructions are expected to assist the agent to revert the changes by interacting with the environment and restoring changed object states and locations (Figure 1).

We also propose a novel method for the SCU task, suitable for embodied AI indoor scenes (EmbSCU). EmbSCU is based on the encoder-decoder transformer [14]. We compare instance-level segmentation masks before and after changes in the change encoder, while the decoder generates rearrangement language instructions for the agent (Figure 2).

We adapted a Segment Anything Model (SAM) [15] for semantic instance-level segmentation in the "everything" mode, where it is asked to provide a valid mask at each point in a predefined point grid on an image. Agents explore unseen environments and, therefore, operate in the "everything" mode without user input prompts (like points or boxes). According to [16], SAM tends to predict masks of the foreground objects better than for the background ones. In our previous research [17], we examined the performance of SAM across several simulators [18]–[21] in the "everything" mode and observed that SAM outputs part and sub-part segmentation masks, instead of the instance-level masks, especially for the background objects. In embodied AI tasks, as the same object may be visible from various viewpoints, recognizing instance-level object masks is crucial for objects in both the foreground and background. We reduce SAM's bias towards foreground object masks in embodied AI scenes by fine-tuning it with a larger point grid in SAOMv2 (64 points per side instead of 32), using the nearest neighbour assignment method from [17].

Given the resource-intensive nature (time, effort, manpower) of real-world data collection for embodied AI tasks, research related to various robotic tasks is typically performed using simulators. No open-world indoor robotics datasets currently exist for detecting changes in unconstrained real-life environments. To evaluate our approach, we created a new dataset using the Ai2Thor simulator [22]. Although the EmbSCU dataset is simulated, it is highly complex, incorporating 104 unique indoor Ai2Thor rooms with 53 real-looking object classes of 3 sizes: small (e.g. mug), medium (e.g. laptop) and large (e.g. fridge). Inspired by works using synthetic data for training a model and then testing its performance on real-image benchmarks [23], [24], we conduct additional sim-to-real experiments, where we directly implemented EmbSCU on the real-world images, without any training on real-world data. Experimental results on sim-to-real transfer demonstrate that EmbSCU has a promising generalization performance in real environments (Figures 1 and 6). While applying EmbSCU directly to real-life scenarios is outside our current scope, we highlight its potential for extension to such scenarios. Our **contributions** can be summarized as follows:

- A new Scene Change Understanding (SCU) task (Section IV) for embodied AI rearrangement scenarios. The task involves predicting a changed location, describing a change, and generating language instructions for the robotic agent to revert a change.
- SAOMv2 (Section V-A) - a fine-tuned version of SAM, adapted to mitigate SAM's bias towards selecting foreground object masks in the "everything" mode.
- EmbSCU (Section V) - a novel method for solving the SCU task, extracting instance-level, class-specific, and type-aware change object masks within SAOMv2's segmentation space. The model achieves promising generalization results on real-world scenes without using any real-life data during training (Section VI-D).
- A new dataset for EmbSCU evaluation, collected within Ai2Thor simulator (Section III), which is suitable for both change detection and change description tasks.

II. RELATED WORK

A. Change Understanding

Scene change detection (SCD) applications can be divided into remote-sensing change detection (RSCD) [1]–[3], [25] for change analysis on the earth's surface and street scene change detection (SSCD) [4]–[6]. Both RSCD and SSCD methods do not describe change contents, while the proposed by us method identifies the change content together with its location within the image.

The image difference captioning (IDC) task aims to generate change language descriptions from a pair of 2D images [10]–[13], sets of 2D sequential scene images [26], 3D point clouds, [27] or a combination of 2D and 3D data [28], [29]. However, [10]–[13] describe changes within a primitive scene setup with simple geometric shapes and solid color backgrounds. Although, [26], [28], [29] focus on changes within real-life scenes, they only provide the change language description, while our method further generates the rearrangement instructions for the robotic agent.

Recently, DyS2Change method [27] was proposed for simultaneous detection and description of changed objects. While DyS2Change works with 3D point clouds, we focus on comparison of 2D image sets. EmbSCU innovatively combines segmentation and language modules. Unlike existing approaches, it facilitates simultaneous change detection, description and language-based rearrangement instruction generation for the agent to revert changes.

B. Change Region Detection

In order to distinguish change regions in 2D image pairs, it is required to extract image features from images before and after changes. Some works [11], [13] compute feature-level differences between before-after image pairs. Others [10], [12], [30] divide an image into unified regions of small size, called patches, for further comparison.

We follow AnyChange [25] approach and propose to utilize object-level differences between before and after image pairs, utilizing image segmentation to achieve this. We adapted a Segment Anything Model (SAM) [15] for semantic instance-level segmentation, leveraging its robust generalization capabilities and extensive training on diverse image data. Both AnyChange and our EmbSCU are built on SAM, therefore can generate change object masks, but EmbSCU adds the semantic content to those masks as it

TABLE I
SIZE OF EMBSCU DATASET

	Train	*Val*	*Test*	*TOTAL*
Num of before-after image pairs	13184	3340	3920	20444
Num of changes	13410	3488	4107	21005

simultaneously provides change language descriptions and rearrangement instructions for the agent to revert changes.

III. DATASET

A. Dataset Setup

We follow the 1-phase room rearrangement task [9] setup to create the EmbSCU dataset. At each step, the agent is given two images before (goal state) and after changes at the same time. During the walkthrough stage [9], an agent navigates through the environment, explores its goal state, and records information about it. After that, we randomly change the positions or openness state for 1 to 5 objects in the room. During the scene change detection stage, the agent aims to identify the changed objects. The agent should reset changes and restore changed objects to their initial state during the unshuffle stage [9] of the rearrangement task.

The rearrangement task [9] involves "move", "open" and "close" change types for each scene. As we define changes for each image pair taken along the agent's path in the scene separately, we split the "move" change type into "move", "add" and "delete" changes. A "move" type change refers to an object altering its position within a given before-after image pair. While "add" and "delete" change types occur, when the relocated object is present in only one of the images within the image pair. Therefore, our dataset supports 5 change types: "add", "delete", "move", "open" and "close".

B. Data Generation

We collected the EmbSCU dataset using the Ai2Thor simulator [22]. For the initial scene set up, we extend the RoomR [9] dataset to include a larger amount of distinct rearrangement settings (from 6000 to 10845 scenarios) involving 53 different object types in 114 scenes. We record the agent's starting position and rotation, as well as start and target positions and rotations for all objects in the scene.

The simplest approach to observe a scene is to have the agent make a full turn around its initial position in the scene. We use the action RotateRight 4 times and a 90° rotation angle, to move the agent. For each robot step, a pair of RGB images before and after changes was collected. For each image pair, we also automatically extracted from Ai2Thor simulator the information about all changed objects within that image pair. Specifically, we recorded the change type and the changed object class (label). Additionally, we recorded both segmentation masks and 2D bounding boxes for each changed object.

C. Statistics

We use 67 train, 17 validation and 20 test scenes from Ai2Thor for the dataset generation. Overall, we provide a total of 21005 change descriptions for 20444 image pairs. Note that 9599 image pairs out of 20444 are without changes. Out of the remaining 10,845 image pairs, 4,961 have 1 change, 3,017 have 2 changes, 1,742 have 3 changes, 843 have 4 changes, and 280 have 5 changes per image pair. The dataset exhibits the following distribution of changes by type: 6660 "add", 6583 "delete", 5250 "move", 1556 "open" and 956 "close" changes.

In total, we used 53 frequently-seen indoor object classes: 11 openable, 25 pickupable objects and 18 receptacles (in or on which pickupable objects can be placed). Tables I and II present dataset size and change statistics for all pickupable and openable objects, respectively.

TABLE II
EMBSCU DATASET STATISTICS: CHANGES PER OBJECT TYPE

Type	Size	Class	Val	Test	Train	TOTAL
pickupable	Medium	Statue	189	348	877	1414
		Laptop	212	244	773	1229
		Vase	113	73	374	560
		TissueBox	66	182	239	487
		BaseballBat	129	79	258	466
		Pot	94	69	300	463
		Pan	86	72	297	455
		WateringCan	61	22	189	272
	Small	ToiletPaper	265	221	884	1370
		SoapBottle	255	217	869	1341
		Bowl	100	262	778	1140
		Mug	233	213	620	1066
		Box	122	161	540	823
		Book	117	181	493	791
		DishSponge	148	163	463	774
		SprayBottle	140	110	517	767
		AlarmClock	117	157	492	766
		Plate	140	157	467	764
		Cup	132	161	401	694
		ScrubBrush	102	93	407	602
		SoapBar	101	97	355	553
		Plunger	66	96	368	530
		Newspaper	58	144	276	478
		PaperTowelRoll	25	17	268	310
		BasketBall	0	73	214	287
openable	medium and large	Drawer	180	237	729	1146
		Cabinet	152	114	539	805
		Toilet	25	35	135	195
		Blinds	15	57	62	134
		ShowerDoor	10	13	62	85
		ShowerCurtain	9	15	49	73
		Fridge	17	11	43	71
		Microwave	8	13	39	60
		LaundryHamper	0	0	17	17
		Safe	1	0	16	17

IV. TASK

We propose a Scene Change Understanding (SCU) task for indoor embodied AI scenes. Given a scene $s \in \mathbb{S}$, an agent's initial position p, and rotation r, the agent explores the environment twice using m actions from set A defined by policy P: before k changes $c_1, ..., c_k \in \mathbb{C}$ and after (where $k \in [1;5]$). On each exploration, the agent collects n RGB images. Therefore, we have for the comparison 2 sets of images: $i^1_{bef}, ..., i^n_{bef} \in \mathbb{I}^{H \times W \times C}_{bef}$ and $i^1_{aft}, ..., i^n_{aft} \in \mathbb{I}^{H \times W \times C}_{aft}$, where H, W, and C denote height, width, and channel

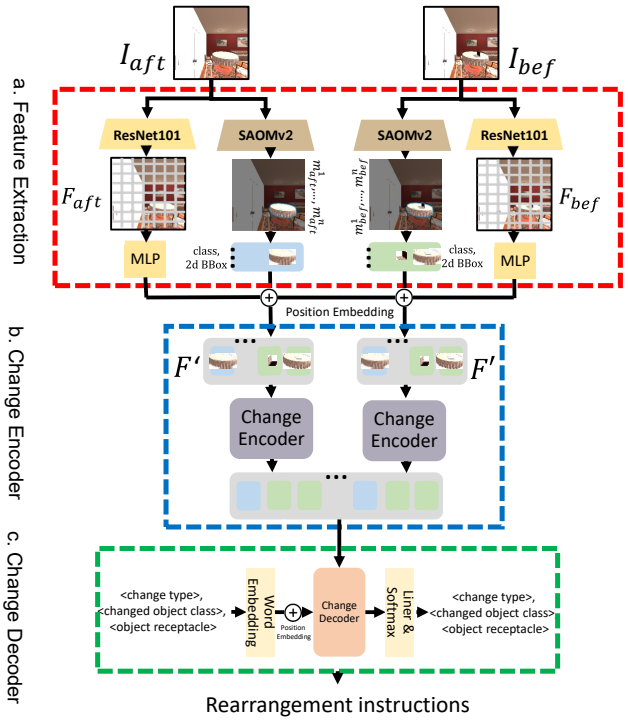

Fig. 2. EmbSCU's framework: (a) Image features extraction from before-after image pairs using ResNet101, concatenated with object masks from SAOMv2. (b) Change Encoder to capture relationships between object masks in the image pairs. (c) Representations from the encoder are used to generate change lists and rearrangement instructions through the decoder.

numbers, respectively. There are 5 possible change types $\mathbb{T} = \{$"*add*","*delete*","*move*","*open*","*close*"$\}$. The changes can occur with the objects $o_1,...,o_k \in \mathbb{O}$, where $k \in [1;5]$.

The SCU task targets simultaneous change detection (changed object mask), description (change type, changed object class) and language instruction generation (to revert changes) based on the predicted change data (Figure 1).

V. METHOD

This paper presents EmbSCU, an innovative method for solving the SCU task (Section IV), allowing concurrent change detection and description, together with the rearrangement instructions generation. Fig. 2 illustrates EmbSCU's overall pipeline. We further provide the details of the feature extraction stage (Section V-A), the change encoder-decoder structure (Section V-B) and mechanism of the rearrangement instructions generation (Section V-C).

A. Feature Extraction

In the feature extraction stage, we propose to utilize object-level differences between before-after image pairs and, therefore, reduce change detection to semantic segmentation. We fine-tune a SAM model [15] for semantic instance-level segmentation, in the "everything" mode using the nearest neighbour assignment method from [17]. We further reduce SAM's bias towards selection of the foreground object masks in embodied AI scenes, using a point grid with 64 × 64 points in SAOMv2 instead of 32 × 32 as in the original SAM. On top of SAM, we add a classifier layer to predict labels for each segmentation mask.

For the two sets of before-after scene images, using SAOMv2 we predict $2 \times n$ sets of object masks, alone with object labels and object bounding boxes. Processing one image pair (i^n_{bef} and i^n_{aft}) at a time, we extract features using the ResNet101 model [31]. Focusing on positive regions where objects exist, we derive SAOMv2's object mask features $f^1_{bef},...,f^p_{bef} \in \mathbb{F}_{bef}$ and $f^1_{aft},...,f^p_{aft} \in \mathbb{F}_{aft}$. Employing an MLP layer for each detected object, we reduce feature dimensions. We concatenate object features with class and bounding box information extracted by SAOMv2. Finally, we feed the extracted mask features into the transformer encoder-decoder model [14].

B. Change Encoder-Decoder

We adopt the Multi-Change Captioning Transformers-Single [10] encoder (MCCFomers-S) architecture to capture SAOMv2's mask relations in before-after image pairs. Using masks from before-after image pairs (extracted by our SAOMv2) as position embeddings, we enhance the model's understanding of spatial relationships between objects. These position embeddings are then concatenated with mask features before and after changes: $\mathbb{F}_{bef}$ and $\mathbb{F}_{aft}$.

The next stage in EmbSCU involves a change transformer decoder similar to [10]. It establishes connections between individual words and the extracted object masks. The change decoder facilitates the generation of multiple change descriptions. In contrast with [10], our aim is not the generation of change captions. For EmbSCU, the output of the encoder-decoder transformer is the change list in the format: "change type", "changed object class", "change region".

C. Generation of Rearrangement Instructions

We further assist the robotic agent to revert changes by generating language instructions based on the transformer encoder-decoder output. To achieve this, we reverse the order of predicted changes during the post-processing stage of the EmbSCU method. For example, if the initial prediction indicates an "add" change, the corresponding instruction advises the robot to "pickup" the object. Similarly, for a "delete" change, the instruction transforms it into a "put" action. A 'move' change type is converted into a sequence of "pickup" and "put" instructions. Regarding openable objects, "close" changes become "open" actions, and vice versa.

D. Loss Function

Similarly to [27], EmbSCU simultaneously performs change detection, change object recognition and change captioning. We adopt standard cross-entropy loss L_{cap} for change captioning [10], where T is the length of a generated caption and $(w_1,\ldots,w_T)$ is the target word sequence:

$$L_{\text{cap}} = \sum_{t=1}^{T} -\log(p(w_t|w_1,\ldots,w_{t-1};I_{bef},I_{aft})) \qquad (1)$$

The SAOMv2's fine-tuning loss consists of the linear combination of the focal loss and the dice loss (for changed

Fig. 3. Comparison between vanilla SAM, Semantic SAM and our SAOMv2 on images from Ai2Thor simulator in the "everything" mode. SAOMv2 predicts the instance-level object masks for both the background and the foreground objects, while SAM tends to divide background objects on parts and sub-parts.

TABLE III
EVALUATION OF OBJECT LOCATIONS (MASKS) PREDICTION

Objects	Num	SAM		SemSAM		SAOMv2	
		IoU ↑	Acc ↑	IoU ↑	Acc ↑	IoU ↑	Acc ↑
Toilet	570	2.94	50.10	8.04	46.73	50.66	65.14
Drawer	351	10.03	65.87	5.24	46.41	19.17	44.67
Cabinet	297	14.87	49.13	11.69	43.11	13.16	26.40
Fridge	34	58.12	58.21	39.80	49.09	65.33	67.36
ToiletPap	520	1.00	44.63	0.90	41.11	40.69	71.07
SoapBot	321	4.88	50.54	3.00	41.89	64.38	80.23
Laptop	266	21.58	56.18	9.41	47.49	77.26	84.91
Mug	264	2.28	66.75	1.29	45.79	39.82	88.10
Box	255	19.00	28.51	10.85	44.05	77.42	83.34

object mask detection) in a 20:1 ratio of focal loss to dice loss, following [15]. On top of that, we add the standard cross entropy loss for changed object class recognition.

VI. EXPERIMENTS

A. Implementation Details

We used the Ai2Thor [22] simulator in our experiments. It offers a rich collection of various sized interactable objects, and 4 different scene types: kitchens, bathrooms, bedrooms and living rooms. We collected RGB images before and after changes (with 224 × 224 resolution, to reduce the computational complexity). For SAOMv2's fine-tuning process, we adopt the pre-trained SAM, with a ViT-B [32] image encoder, to increase the training speed. SAOMv2 was trained for 200 epochs using the 64×64 point grid in the "everything" mode. We set the initial learning rate as 10^{-3} and adopt the AdamW [33] optimizer with a cosine scheduler. Similar to [10], for both the encoder and the decoder we implemented the transformer architecture using two layers and four heads. We set the learning rate to 10^{-3}, and trained the EmbSCU model for 20 epochs with the Adam optimizer [34].

B. Change Detection Evaluation

The evaluation of EmbSCU's ability to predict object locations within a 2D image involves assessing SAOMv2's object segmentation masks. We used personalized evaluation approach from [35] to assess SAOMv2's ability to segment changed objects in any unseen poses or scenes without manual prompting (in the "everything" mode). Similar to [35], we evaluate SAOMv2's segmentation masks by calculating Intersection over Union (IoU) and accuracy scores (example in Figure 4), then combining those metrics per class. The comparison of our SAOMv2, SemSAM [36] and the original SAM is present in Table III for openable (top) and pickupable (bottom) objects (objects with the highest occurrence frequency in the validation set). We chose SemSAM for comparison with our SAOMv2, as it is a SAM's fine-tuned version suitable for instance level masks generation in the "everything" mode. For all 34 changed object classes, SAOMv2 has an increase in mIoU from 17.56 to 55.40 and in mAcc from 59.9 to 77.15 in comparison with SAM. Table III shows a notable increase in IoU scores for all pickupable and most openable objects in comparison with the vanilla SAM and the fine-tuned SemSAM. However, SAOMv2 exhibits lower segmentation accuracy for certain openable objects, such as cabinets or drawers. This is due to the occlusion by various pickupable objects, resulting in partial visibility and posing challenges for boundary prediction.

The qualitative results in the "everything" mode are presented in Fig. 3 for our SAOMv2, SemSAM and the vanilla SAM. SAOMv2 tends to predict instance-level object masks in the "everything" mode and mitigate the bias of SAM towards selecting foreground object masks.

C. Change Description Evaluation

We used the following metrics to evaluate EmbSCU's generated change list (change descriptions):

- Overall change accuracy: percentage of with-change image pairs, where the number of changes, change types, and change object classes were simultaneously predicted correctly.
- Number of changes, change type and changed object classification accuracy: the count of with-change image pairs, the number of changes, recalled change type and recalled change object class were predicted correctly.

We assess the efficiency of the newly introduced mask-feature extraction mechanism in EmbSCU for the generation of change lists. In a series of experiments (Table IV), we replaced the EmbSCU feature extractor (Figure 2 (a)), which includes a combination of ResNet101 [31] and SAOMv2, with the standalone ResNet101 and CLIP [37] networks. We use MCCFormers-S and -D transformers [10] for change content prediction. Furthermore, we implemented EmbSCU (obj) method, which utilizes only SAOMv2 output for the feature extraction (without the use of the ResNet101 features).

EmbSCU's change region detection and change list prediction examples are present in Figures 1 (left) and 5. The comparison of our EmbSCU and the baseline models is presented in Table IV. Among all baselines, EmbSCU has the highest overall change accuracy of 46.6 percent. It means that EmbSCU can predict image pairs with and without changes with high confidence. The combination of ResNet and SAOMv2 features enhances EmbSCU's object detection

TABLE IV

CHANGE DESCRIPTION EVALUATION USING EMBSCU DATASET

Img Encoder	Transformer	Epoch	Overall, % ↑	Change, % ↑		
				Number	Type	Obj Class
Resnet101	MCC-D	20	45	30.7	50.9	12.7
Resnet101	MCC-S	20	45.8	41.4	58.5	17.5
CLIP	MCC-S	20	29.8	36.2	49	20.2
EmbSCU (obj)	MCC-S	20	28.6	22.6	54.2	29.4
EmbSCU	MCC-S	20	46.6	41.7	62.9	22.2

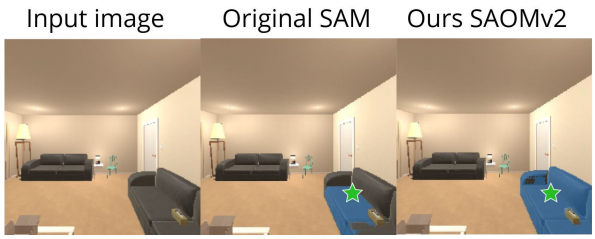

Fig. 4. Comparison between vanilla SAM and our SAOMv2 on images from Ai2Thor simulator in the "everything" mode. A green star represents the foreground object point (from SAOMv2's 64 x 64 point grid) chosen for the evaluation of the object. Best view in color and zoom in.

efficiency, by leveraging ResNet101's robustness to viewpoint changes, and SAOMv2's objects knowledge. EmbSCU is also the best method in predicting the number of changes in with-change image pairs (41.7 score), and significantly outperforms all baselines in change type accuracy (62.9 score). EmbSCU (obj) has the highest score for the object change accuracy, due to the strong connection to the object in EmbSCU (obj).

D. Sim-to-Real Inference Stage

We created an additional sim-to-real test set of before-after image pairs in real life environments, containing objects from 53 frequently-seen indoor object classes in diverse contexts (in different poses and scene types). Our sim-to-real images were captured at varying times of the day and from slightly different viewpoints. The EmbSCU's change region detection and rearrangement instructions are shown in Figures 1 (right) and 6. Our experiments show that EmbSCU generalizes well to the real-world data. It predicts the change region, the changed object type and the change type in real-life indoor scenes, despite training without any real-world data samples. SAOMv2 is robust for slight viewpoint changes, but may require modifications with significant viewpoint shifts.

VII. CONCLUSIONS

The two main contributions of this paper include introducing a novel Scene Change Understanding (SCU) task for indoor environments and proposing the EmbSCU method to address it. EmbSCU is not only able to detect changes but also provides change descriptions and generates language rearrangement instructions for the robotic agents to revert the changes. EmbSCU demonstrates promising generalization results on real-world scenes without using any real-life data during training. The enhanced capability of robots to understand and respond to changes through the generated language instructions, opens up possibilities for improved human-robot interaction, rearrangement, and problem-solving in indoor embodied AI environments.

In the future work, we will merge before-after image sets into two panoramas, improving the generation of a comprehensive change object list. This minimizes issues like missing or misidentified changes. We plan to generate more natural rearrangement instructions using Large Language Models and involve the user into the dialog with the agent to revert changes. We also plan to work with both temporal and spatial changes in embodied AI scenes.

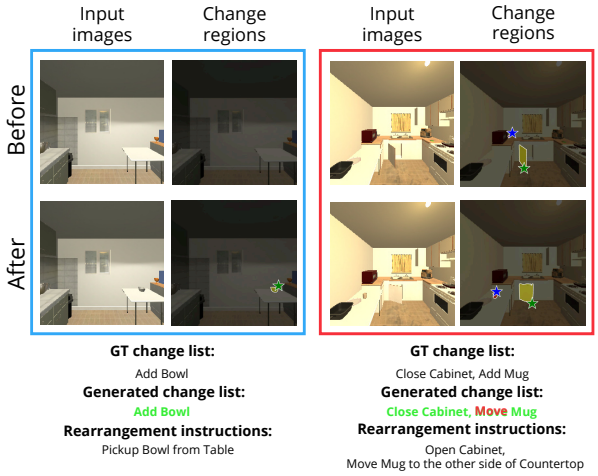

Fig. 5. SAOMv2's generated change masks and EmbSCU's change list prediction on images collected from Ai2Thor. Stars represent points from the 64x64 point grid, used by SAOMv2 to generate change object masks. Depending on the change type, we generate rearrangement language instructions for the robotic agent. Best view in color and zoom in.

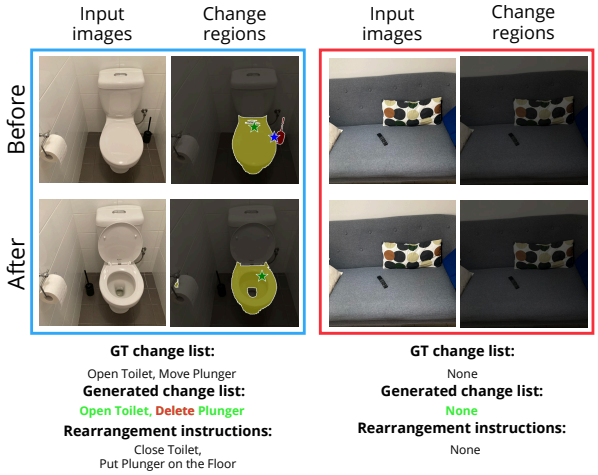

Fig. 6. SAOMv2's generated change masks and EmbSCU's change list prediction on our real-to-sim test set. Stars represent points from the 64x64 point grid, used by SAOMv2 to generate change object masks. Note that sim-to-real images were captured at varying times of the day and from slightly different viewpoints. EmbSCU can identify image pairs with (left) and without changes (right) equally well. Best view in color and zoom in.

REFERENCES

[1] H. Chen and Z. Shi, "A spatial-temporal attention-based method and a new dataset for remote sensing image change detection," *Remote Sensing*, vol. 12, no. 10, p. 1662, 2020.

[2] H. Zheng, M. Gong, T. Liu, F. Jiang, T. Zhan, D. Lu, and M. Zhang, "Hfa-net: High frequency attention siamese network for building change detection in vhr remote sensing images," *Pattern Recognition*, vol. 129, p. 108717, 2022.

[3] Y. Sun, L. Lei, D. Guan, J. Wu, and G. Kuang, "Iterative structure transformation and conditional random field based method for unsupervised multimodal change detection," *Pattern Recognition*, vol. 131, p. 108845, 2022.

[4] J.-M. Park, J.-H. Jang, S.-M. Yoo, S.-K. Lee, U.-H. Kim, and J.-H. Kim, "Changesim: Towards end-to-end online scene change detection in industrial indoor environments," 2021.

[5] Y. Lei, D. Peng, P. Zhang, Q. Ke, and H. Li, "Hierarchical paired channel fusion network for street scene change detection," *IEEE Transactions on Image Processing*, vol. 30, pp. 55–67, 2020.

[6] S. Chen, K. Yang, and R. Stiefelhagen, "Dr-tanet: Dynamic receptive temporal attention network for street scene change detection," 2021.

[7] A. Ismail and M. Awad, "Bldnet: A semi-supervised change detection building damage framework using graph convolutional networks and urban domain knowledge," *arXiv preprint arXiv:2201.10389*, 2022.

[8] V. Oludare, L. Kezebou, O. Jinadu, K. Panetta, and S. Agaian, "Attention-based two-stream high-resolution networks for building damage assessment from satellite imagery," in *Multimodal Image Exploitation and Learning 2022*, vol. 12100. SPIE, 2022, pp. 224–239.

[9] L. Weihs, M. Deitke, A. Kembhavi, and R. Mottaghi, "Visual room rearrangement," 2021.

[10] Y. Qiu, S. Yamamoto, K. Nakashima, R. Suzuki, K. Iwata, H. Kataoka, and Y. Satoh, "Describing and localizing multiple changes with transformers," 2021.

[11] H. Kim, J. Kim, H. Lee, H. Park, and G. Kim, "Viewpoint-agnostic change captioning with cycle consistency," in *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*. Los Alamitos, CA, USA: IEEE Computer Society, oct 2021, pp. 2075–2084. [Online]. Available: https://doi.ieeecomputersociety.org/10.1109/ICCV48922.2021.00210

[12] Z. Guo, T.-J. J. Wang, and J. Laaksonen, "Clip4idc: Clip for image difference captioning," 2022.

[13] D. H. Park, T. Darrell, and A. Rohrbach, "Robust change captioning," 2019.

[14] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," 2023.

[15] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, P. Dollár, and R. Girshick, "Segment anything," 2023.

[16] W. Ji, J. Li, Q. Bi, T. Liu, W. Li, and L. Cheng, "Segment anything is not always perfect: An investigation of sam on different real-world applications," 2023.

[17] M. Khan, Y. Qiu, Y. Cong, J. Abu-Khalaf, D. Suter, and B. Rosenhahn, "Segment any object model (saom): Real-to-simulation fine-tuning strategy for multi-class multi-instance segmentation," *IEEE International Conference on Image Processing (ICIP 2024)*, 2024 (accepted).

[18] E. Kolve, R. Mottaghi, W. Han, E. VanderBilt, L. Weihs, A. Herrasti, M. Deitke, K. Ehsani, D. Gordon, Y. Zhu, *et al.*, "Ai2-thor: An interactive 3d environment for visual ai," *arXiv preprint arXiv:1712.05474*, 2017.

[19] C. Yan, D. Misra, A. Bennnett, A. Walsman, Y. Bisk, and Y. Artzi, "Chalet: Cornell house agent learning environment," *arXiv preprint arXiv:1801.07357*, 2018.

[20] X. Gao, R. Gong, T. Shu, X. Xie, S. Wang, and S.-C. Zhu, "Vrkitchen: an interactive 3d virtual environment for task-oriented learning," *arXiv preprint arXiv:1903.05757*, 2019.

[21] C. Li, F. Xia, R. Martín-Martín, M. Lingelbach, S. Srivastava, B. Shen, K. Vainio, C. Gokmen, G. Dharan, T. Jain, *et al.*, "igibson 2.0: Object-centric simulation for robot learning of everyday household tasks," *arXiv preprint arXiv:2108.03272*, 2021.

[22] E. Kolve, R. Mottaghi, W. Han, E. VanderBilt, L. Weihs, A. Herrasti, M. Deitke, K. Ehsani, D. Gordon, Y. Zhu, A. Kembhavi, A. Gupta, and A. Farhadi, "Ai2-thor: An interactive 3d environment for visual ai," 2022.

[23] M. J. Black, P. Patel, J. Tesch, and J. Yang, "Bedlam: A synthetic dataset of bodies exhibiting detailed lifelike animated motion," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 8726–8737.

[24] N. Liu, Y. Cai, T. Lu, R. Wang, and S. Wang, "Real–sim–real transfer for real-world robot control policy learning with deep reinforcement learning," *Applied Sciences*, vol. 10, no. 5, p. 1555, 2020.

[25] Z. Zheng, Y. Zhong, L. Zhang, and S. Ermon, "Segment any change," *arXiv preprint arXiv:2402.01188*, 2024.

[26] Y. Qiu, Y. Satoh, R. Suzuki, K. Iwata, and H. Kataoka, "3d-aware scene change captioning from multiview images," *IEEE Robotics and Automation Letters*, vol. 5, no. 3, pp. 4743–4750, 2020.

[27] Y. Qiu, S. Yamamoto, R. Yamada, R. Suzuki, H. Kataoka, K. Iwata, and Y. Satoh, "3d change localization and captioning from dynamic scans of indoor scenes," in *2023 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 2023, pp. 1176–1185.

[28] Y. Qiu, Y. Satoh, R. Suzuki, K. Iwata, and H. Kataoka, "Indoor scene change captioning based on multimodality data," *Sensors*, vol. 20, no. 17, p. 4761, 2020.

[29] Y. Qiu, K. Nakashima, Y. Satoh, R. Suzuki, K. Iwata, and H. Kataoka, "Scene change captioning in real scenarios," in *Artificial Intelligence in HCI*, H. Degen and S. Ntoa, Eds. Cham: Springer International Publishing, 2022, pp. 405–419.

[30] Y. Cong, W. Liao, B. Rosenhahn, and M. Y. Yang, "Learning similarity between scene graphs and images with transformers," *arXiv preprint arXiv:2304.00590*, 2023.

[31] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," 2015.

[32] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," 2021.

[33] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," 2019.

[34] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2017.

[35] R. Zhang, Z. Jiang, Z. Guo, S. Yan, J. Pan, H. Dong, P. Gao, and H. Li, "Personalize segment anything model with one shot," *arXiv preprint arXiv:2305.03048*, 2023.

[36] F. Li, H. Zhang, P. Sun, X. Zou, S. Liu, J. Yang, C. Li, L. Zhang, and J. Gao, "Semantic-sam: Segment and recognize anything at any granularity," *arXiv preprint arXiv:2307.04767*, 2023.

[37] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever, "Learning transferable visual models from natural language supervision," 2021.