Graph Representation for Order-aware Visual Transformation

Yue Qiu, Yanjun Sun, Fumiya Matsuzawa, Kenji Iwata, Hirokatsu Kataoka
National Institute of Advanced Industrial Science and Technology (AIST)
{qiu.yue, yanjun.Son, fumi8.matsuzawa, kenji.iwata, hirokatsu.kataoka}@aist.go.jp

Abstract

This paper proposes a new visual reasoning formulation that aims at discovering changes between image pairs and their temporal orders. Recognizing scene dynamics and their chronological orders is a fundamental aspect of human cognition. The aforementioned abilities make it possible to follow step-by-step instructions, reason about and analyze events, recognize abnormal dynamics, and restore scenes to their previous states. However, it remains unclear how well current AI systems perform in these capabilities. Although a series of studies have focused on identifying and describing changes from image pairs, they mainly consider those changes that occur synchronously, thus neglecting potential orders within those changes. To address the above issue, we first propose a visual transformation graph structure for conveying order-aware changes. Then, we benchmarked previous methods on our newly generated dataset and identified the issues of existing methods for change order recognition. Finally, we show a significant improvement in order-aware change recognition by introducing a new model that explicitly associates different changes and then identifies changes and their orders in a graph representation.

1. Introduction

The Only Constant in Life Is Change.

- Heraclitus

Humans conduct numerous reasoning processes beyond object and motion recognition. Through these processes, we can capture a wide range of information with just a glimpse of a scenario. To achieve human-level visual understanding, various studies have recently focused on different aspects of visual reasoning, such as compositional [1–4], causal [5, 6], abstract [7–9], abductive [10, 11], and commonsense visual reasoning [12, 13]. Due to the ever-changing visual surrounding, perceiving and reasoning over scene dynamics are essential. However, most existing visual reasoning studies focus on scenes in fixed periods of time. Therefore, this study focuses on a new formulation of visual reasoning for identifying scene dynamics.

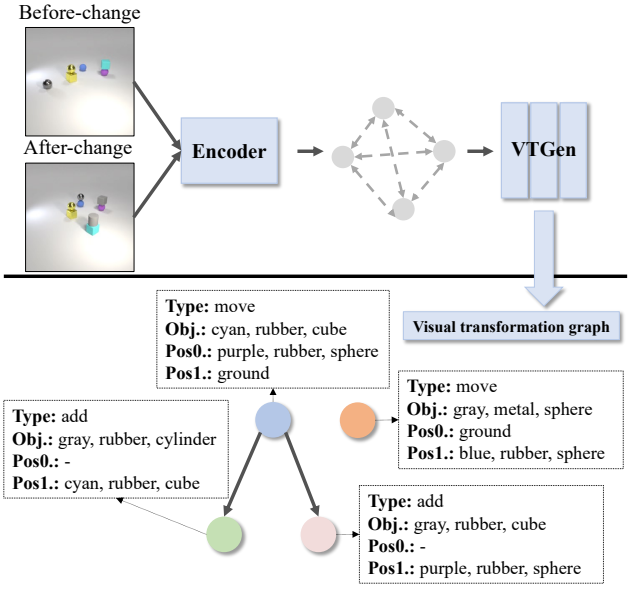

Figure 1. Overview of the proposed order-aware change recognition model VTGen (top). From an image pair observed before and after multiple synchronous and asynchronous changes, VTGen generates a visual transformation graph (bottom) where nodes indicate change contents (including type, object attributes, original position described by what is underneath it, and new position, and directed edges indicate temporal orders of changes.

Due to variations in the spatial positions of objects and the temporal order of human activities, changes within a pair of observations could occur simultaneously or asynchronously. Several recent studies have already discussed recognizing and describing synchronous changes from a pair of images via natural language texts [14–16] while neglecting the potential orders between changes. However, identifying temporal orders is an integral aspect of revealing how scene dynamics occur in time, making it possible to restore scenes to their previous states. Temporal orders are also critical in a variety of applications, such as room rearrangement [17], assembly operation [18, 19], and instruction following [20, 21]. Change order recognition presents new challenges as it requires reasoning over underlying tempo-

ral events, and it also complicates single change recognition due to entangled appearances and localization. However, despite its complexity, humans exhibit high performance in finding multiple changes and determining their temporal orders from a pair of images. For example, even human children under age five can perform assembly operations with toys in games like LEGO building blocks. Therefore, in this work, we are particularly interested in how well the current AI methods perform in order-aware change recognition.

Similar to discovering dynamics and orders from a pair of scene observations, there is a group of works that discusses the step-by-step assembly of objects from their component parts [18, 22–24]. Such part assembly studies tend to focus on recovering the sequence steps for rebuilding objects and are thus highly useful in robotic applications used for assembly operations or instruction following. However, part assembly operations focus on the reconstruction of objects from their parts, and not finding the differences between two discrete scene observations. Moreover, instead of directly recovering all steps from two single observations, existing part assembly methods require additional information, such as language instructions or demonstration videos, for their step generation processes.

As shown in Figure 1, this study proposes a new task to identify order-aware changes directly from a pair of images. Most existing studies generate a single sentence [14, 15], paragraph [16], or triplets [25] for describing changes. However, sentences are lengthy and less suitable for simultaneously indicating change contents and their orders, and make model analysis and evaluation opaque. Hence, we propose the use of an order-aware transformation graph (Figure 1 bottom). Change contents are represented by nodes and their chronological orders by directed edges. To diagnose model performance, we generated a dataset, named order-aware visual transformation (OVT), consisting of asynchronous and synchronous changes between scene observations.

We then conducted benchmark experiments using existing methods and found they showed seriously degraded performance in terms of order-aware change recognition. Although neglected by existing methods, associations between changes, and disentangled representations of change contents and orders are useful in identifying order-aware changes. Therefore, we propose a novel method called visual transformation graph generator (VTGen) that explicitly associates different changes and generates a graph that describes change contents and their orders in a disentangled manner. VTGen achieved state-of-the-art performance in the OVT dataset and an existing benchmark CLEVR-Multi-Change [16], and outperformed existing methods by large margins. However, we also found a significant performance gap between the best-performing model and humans. We hope our research and OVT dataset can contribute to achieving human-level visual reasoning in scene dynamics.

Our contributions are three-fold: i. We propose a novel task and a dataset named OVT for order-aware visual transformation. ii. We report on benchmark evaluations of existing change recognition methods in order-aware change recognition and discuss their shortcomings. iii. We propose a novel method VTGen that achieves state-of-the-art performance in the OVT dataset and an existing change recognition benchmark.

2. Related Work

2.1. Change Recognition

Change detection, which aims to identify changed regions from a pair of images or point clouds, has been discussed extensively for use in various scenarios, such as robotics [26, 27] and street scenes [28, 29]. However, the methods discussed in those studies were limited to identifying the changed regions and were unable to recognize the details of the scene changes.

More recently, the change captioning task has been proposed for describing scene changes from a pair of images [14–16], or 3D data [30]. The CLEVR-Change dataset [15] is constructed based on the CLEVR engine [1] for describing a single change between two images, while the CLEVR-Multi-Change dataset [16] deals with scenes with multiple object changes. Despite its importance, none of the above datasets deal with changes occurring in specific orders. More similar to our work, the authors of the TRANCE dataset [25] also discussed change orders. However, they used triplet (objects, change types, and contents) lists that only represent changes in a specific order. Instead, we propose using graph representation to identify changes and their orders explicitly. Additionally, the TRANCE dataset only considers potential temporal orders between two changes, whereas our dataset extensively examines various change order patterns within four changes.

Jhamtani and Berg-kirkpatrick [14] proposed the use of pixel-level image differences to identify changes. Park *et al*. [15] instead proposed the use of feature-level differences for enhancing robustness to camera pose changes. Hong *et al*. [25] proposed an encoder-decoder structure and evaluated a range of model choices to boost performance. Qiu *et al*. [16] proposed a transformer structure to associate image patches and words in sentences to identify and describe multiple changes. These aforementioned methods mainly generate a sequence to describe changes without explicitly considering the associations between changes. To better recognize entangled changes and identify their orders, we propose a method to associate changes explicitly and then generate a graph structure for describing changes.

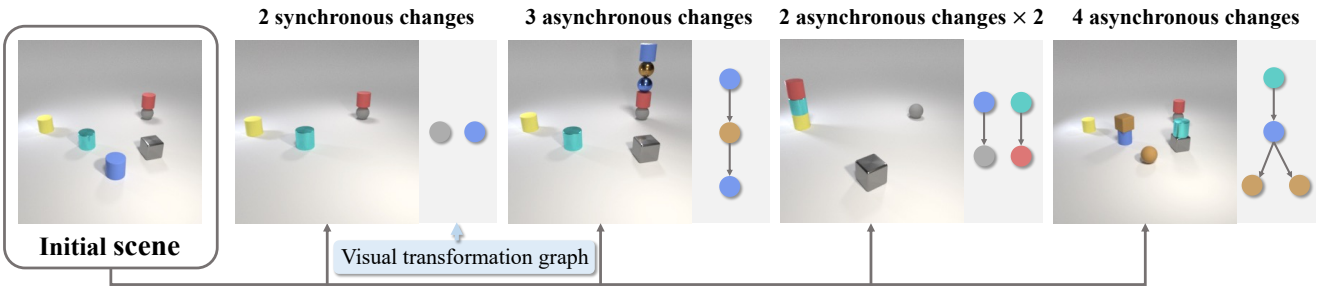

Figure 2. Illustration of different change patterns used in the OVT dataset. The dataset consists of image pairs containing synchronous and asynchronous changes and visual transformation graphs where nodes and edges indicate changes (contents are omitted here) and orders.

2.2. Structured Scene Representation

Recently, a group of studies with the aim of recovering graph [31–33] or program-like descriptions [34–36] from an input image were conducted. Johnson *et al*. first proposed scene graphs to represent images [31], which now have been used in many tasks. Cong used an encoder-decoder structure to generate subject-predicate-object triplets from input queries [32]. Li reduced computation cost by proposing an equivalent bipartite graph generation process [33]. Ellis *et al*. introduced a CNN-based method [34] that describes hand-drawn primitive shapes, such as groups of circles or triangles, in programs, and then transfers programs into LaTeX-style figures. Liu *et al*. [35] further proposed a dataset with various 3D shapes and distribution patterns of shapes and achieved the dataset with photorealistic images. Liu *et al*. also proposed a novel method that consists of an object parser, a group recognizer, and a program synthesizer to generate image-describing programs hierarchically. Wu *et al*. proposed an encoder-decoder-based method that first recovers the physical world representations from images and then applies a graphical engine for re-rendering the input scene [36]. Different from these above studies that describe an image with structured representations, we focus on the structured representation of changed parts in image pairs instead from a single image.

2.3. Part Assembly

Part assembly has been long discussed in robotics and computer vision studies [18, 22–24]. Chen *et al*. proposed the neural shape mating task to identify 6-DoF poses to assemble two object parts into a whole, and a method that integrates pose estimation and implicit shape reconstruction structures [18]. Similarly, Willis *et al*. also tackled assembling two object parts together, and they achieved this by estimating the possible connections between the joints of two object parts using a graph neural network (GNN) [22].

Unlike studies in which only assembling a shape from two parts is considered, Li *et al*. proposed a novel task - single-image-guided 3D part assembly along with a dataset consisting of multiple 3D parts of various types of furniture [23]. They also proposed a model which uses the correspondences between 3D parts shapes and its 2D projected images for part assembly. Similarly, Zhang *et al*. tackled the 3D part assembly task and focused on one of its subproblems, predicting 6-DoF poses for object parts [24]. More specifically, they proposed a GNN-based method to reason over parts and the whole object and then to optimize the parts' poses in a coarse-to-fine manner.

Similar to our work, these studies also recover a sequence of actions from images or 3D shapes. However, instead of assembling an object from its parts, we focus on identifying changes and their temporal orders from two images. Huang *et al*. proposed neural task graphs to predict a sequence of robotic grasping by watching a demonstration video [19]. They also considered the transformation between discrete scene observations, but their process appears to require additional videos for action sequence prediction.

3. Order-aware Visual Transformation (OVT) dataset

Change order recognition remains less discussed in previous change detection [26–29] and captioning [14–16] studies. Recognizing the temporal order of changes can be beneficial in providing a better understanding of how the changes happened, and that the process requires various visual reasoning abilities. We created a dataset that includes both asynchronous and synchronous changes and introduced a graph representation of changes to reveal how well existing methods comprehend order-aware changes and to facilitate models with this critical visual reasoning ability.

3.1. Dataset Generation

To achieve large-scale dataset generation with lower labeling cost, similar to the existing change captioning dataset CLEVR-Change [15] and CLEVR-Multi-Change [16], we also use the CLEVR-Engine [1] for dataset generation. The

Split	Total scenes	Change num.	Change layer	Add	Delete	Move
Train	120,000	3.26	2.13	1.35	0.76	1.15
Test	30,000	3.26	2.13	1.35	0.75	1.16

Table 1. OVT dataset statistics. Change num. stands for number of changes. The averages of the change number and layer, along with the number of add, delete, and move changes are recorded.

CLEVR-Engine allows generation of photo-realistic scenes with various objects. Hence, we can generate scene changes by replacing the objects in a scene.

We use three atomic change types: "add", "delete", and "move", in order to create image pairs with changes. Importantly, we focus on order-aware change recognition. To create changes with specific orders, we record the location of each change and randomly add futher changes to these recorded change locations. To precisely represent the changes and their orders, we introduce an visual transformation graph (Figure 1, bottom), in which we record change contents, including the change type, changed objects, original position (Pos0), and new positions (Pos1) in the nodes and directed edges. Several examples of asynchronous and synchronous changes are shown in Figure 2. Similar to existing datasets [15, 16], we also introduce random illumination and camera angle changes between each set of paired images. Here, it should be noted that to alleviate the ambiguity in change orders, we also add a restriction that ensures every object has a unique combination of attributes, and that each object has a maximum of one change.

3.2. Dataset Statistics

Following the generation process mentioned above, we generated the OVT dataset, using objects with random attributes, including three shape types (cube, cylinder, and sphere), eight colors (red, yellow, cyan, gray, green, blue, purple, and brown), and two materials (rubber and metal). We placed six to eight objects randomly in a scene, and randomly adopted changes. The maximum number of changes and change layers (depths of transformation graphs) were both set to four. Additionally, we used 15 types of change order patterns, including asynchronous and synchronous changes. All the change patterns are provided in the supplementary material. The dataset statistics are summarized in Table 1. The dataset generation process allows for incorporating various change patterns.

4. Approach

In this paper, we focus on a novel task of recognizing order-aware visual transformation by simultaneously distinguishing changes and their temporal orders between images. This task requires models to recognize correlations of image patches from image pairs to identify changes. Due to the existence of asynchronous changes, the change regions have spatial intersections. Thus, to precisely distinguish changes from the others, and determine their temporal orders, the correlations between different changes are critical.

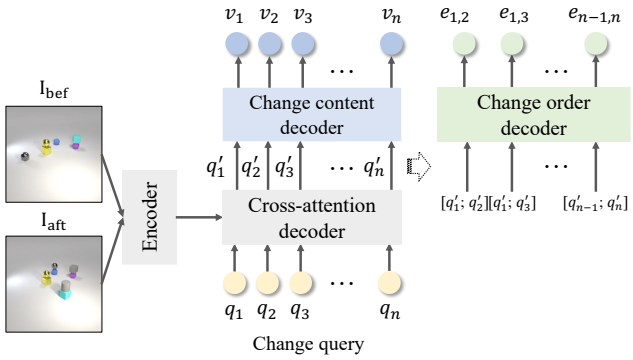

Figure 3. Overview of our proposed method VTGen. Position embedding, change embedding, and classification heads are omitted in the figure. [;] denotes the concatenation operation.

Most existing change recognition methods generate a single sentence [14, 15] or a paragraph [16] for describing change contents without disentangling change contents and orders and neglecting the correlations between changes. On the contrary, we propose a graph structure for describing order-aware visual transformation instead of language text. We also propose the VTGen model, which first coarsely identifies changes from images and then recognizes the detailed change contents and their orders based on the correlations between changes.

4.1. Problem Formulation

Given two images $\mathrm{I}_{\mathrm{bef}}$ and $\mathrm{I}_{\mathrm{aft}}$ of the same scene observed at two different times, we generate a visual transformation graph G to describe the changes and their temporal orders. The transformation graph $G = \{V, E\}$, where V denotes the change contents, and E describes the orders of the changes.

In detail, the nodes $V = \{v_1, v_2, ..., v_n\}$ describe change contents, where n denotes the number of changes. Each node describes a combination of change content as $v_n = \{c_n, o_n, s_n, d_n\}$, where c, o, s, d denotes the change types, the attributes of the changed objects, and the before- and after-change positions of the changed objects, respectively. The edges $E = \{e_{1,2}, e_{1,3}, ..., e_{i,j}, ..., e_{n-1,n}\}_{i \neq j}$ describe the temporal orders of the changes, where $e_{i,j} = 1$ only when change v_j happened just after v_i.

4.2. Transformation Graph Generation

Given image pairs $\mathrm{I}_{\mathrm{bef}}$ and $\mathrm{I}_{\mathrm{aft}}$, we use an encoder-decoder structure VTGen (Figure 3) to recognize changes in images and then generate a visual transformation graph G consisting of nodes V (changes) and edges E (temporal

orders).

Transformation Encoder. Identifying associations between images and inner-image patches is crucial in discerning changes. Similar to MCCFormers [16], we use a transformer [37] structure for finding correlations between image regions from two images. More specifically, we first introduce a CNN structure to encode the input image and then spatially divide each image into m patches, which results in $\mathrm{I}^{'}_{\mathrm{bef}} = \{\mathrm{i}^{1}_{\mathrm{bef}}, \mathrm{i}^{2}_{\mathrm{bef}}, ..., \mathrm{i}^{\mathrm{m}}_{\mathrm{bef}}\}$ and $\mathrm{I}^{'}_{\mathrm{aft}} = \{\mathrm{i}^{1}_{\mathrm{aft}}, \mathrm{i}^{2}_{\mathrm{aft}}, ..., \mathrm{i}^{\mathrm{m}}_{\mathrm{aft}}\}$, where $\mathrm{i}^{\mathrm{m}}_{\mathrm{bef}}, \mathrm{i}^{\mathrm{m}}_{\mathrm{aft}} \in \mathbb{R}^{\mathrm{D}}$. We also introduce position embedding for encoding a patch index in a D−dimensional feature using a linear layer and then sum up the image features with the patch features and concatenate the two image features together, resulting in $\mathrm{I}^{'} \in \mathbb{R}^{2\mathrm{m}\times\mathrm{D}}$. Finally, we use transformer self-attention to obtain the correlations between patches, as $\mathrm{I}^{''} = \mathrm{Attention}\,(\mathrm{I}^{'}, \mathrm{I}^{'}, \mathrm{I}^{'})$.

Transformation Graph Generator. We obtain $\mathrm{I}^{''}$ through the encoder structure, which contains image features for identifying changes. Here, we correlate image features with the contents of each change to identify individual changes. After that, we use a change content decoder and an order decoder to generate transformation graphs. In detail, we first use transformer structure to correlate change contents with image features $\mathrm{I}^{''}$. To embed changes, we adapt a linear and a transformer self-attention layer to change contents $\{c_n, o_n, s_n, d_n\}$ to obtain a D−dimensional feature for each detailed change content. Then we sum up all contents belonging to one change, which results in the change query $Q = \{q_1, q_2, ..., q_n\}$, where $q \in \mathbb{R}^{D}$. After that, we use a transformer structure to obtain correlations between $\mathrm{I}^{''}$ and Q by Attention $(\mathrm{Q}, \mathrm{I}^{''}, \mathrm{I}^{''})$, thereby resulting in $Q^{'} \in \mathbb{R}^{n\times D}$.

To recognize order-aware changes, the correlations between changes are critical in distinguishing each change from the others. Hence, we further correlate each q' in $Q^{'}$ using a change content decoder (node) as shown in Figure 3, resulting in change content features $\{v_1, v_2, ..., v_n\}$. To determine pairwise temporal orders, we first concatenate each pair of change features in $Q^{'}$, resulting in $n \times (n-1)$ pairs of features with $\mathbb{R}^{2\times D}$−dimension. Next, we introduce a change order decoder (edges) to find the relationships between edges, resulting in change order features $\{e_{1,2}, e_{1,3}, ..., e_{n-1,n}\}$. Finally, we add classification heads over obtained change content and order features for predicting detailed changes and their binary temporal orders.

During the experiments, we used two structures, a transformer and a graph convolutional network (GCN) [38], in the implementation of change content and order decoders. Same to other transformers used in our model, the transformer node decoder conducts self-attention operations over features. Similar to [39, 40], we implemented a GCN using the MLP structure. For each layer, we first concatenate features of each pair of nodes (or edges) and process features by an MLP layer. Then updated node (or edge) features are obtained by summing up all features containing that node (or edge). Please refer [39, 40] for more details.

Loss Function. Since there are synchronous changes in OVT where change orders are arbitrary, inspired by [41], we adopted a graph matching loss for change content recognition to find the correspondence of n elements $\sigma \in \mathfrak{S}_n$, which minimizes the loss between predicted results v and ground truth v^{gt} with the following equation, where L_{match} is cross-entropy loss.

$$\hat{\sigma} = \underset{\sigma\in\mathfrak{S}_n}{\arg\min} \sum_{i=1}^{n} L_{match}(v_i, v^{gt}_{\sigma(i)}) \tag{1}$$

For computing edge losses, we first re-arrange edges with the order of the correspondence obtained above and then adopt a cross-entropy loss. We sum up node and edge losses for model training.

5. Experiments

To evaluate the proposed order-aware visual transformation recognition, we conducted comparison experiments on the proposed OVT dataset. We also report results on an existing CLEVR-Multi-Change dataset [16]. We compared the proposed method VTGen with state-of-the-art methods, which generate representations, such as captions [15, 16], or triplets [25], for describing changes directly from the input of image pairs without additional object detectors.

5.1. Experimental Setups

Existing Methods. In order to facilitate existing change captioning methods DUDA [15], MCCFormers-D, and MCCFormers-S [16] on the OVT dataset, we transfer transformation graphs in our dataset OVT to sentences by listing all change information, including change type, changed object, original position, and new position one-by-one. We separate each individual change that can happen simultaneously with ";". We successively list changes with specific temporal orders and separate them with ",". We add a ":" after a change with two following changes where these two changes could happen simultaneously. In this manner, the change graph given in Figure 1 (bottom) is "move gray metal sphere ground blue rubber sphere; move cyan rubber cube purple rubber sphere ground: add gray rubber cylinder none cyan rubber cube; add gray rubber cube none purple rubber sphere ;". For the change triplet generation method TranceNet [25], instead of change triplets, we generate each change set with 11-dimensional information (change contents and its order indicated by ";", "," or ":").

Evaluation Metrics. We evaluate methods on the test split of the OVT dataset with change recognition accuracy and recall evaluation metrics. In detail, we adopted per-scene accuracy where we count predicted results with the

same change contents (including change type, object attributions, original and new positions, and orders) as 1 and the predicted result with incorrect change contents as 0, and we compute correctness over all scenes. We also introduced per-change accuracy and recall in a with-order manner and without-order manner. In the with-order manner, a single change is counted as 1 when all nodes in the sub-graph (connected graph) have correct change contents and temporal orders. For the without-order manner, a single change is counted as 1 when it has correct change contents. We also evaluated per-change-type accuracy to evaluate the model performance for changes with different change types in the without-order manner. Additionally, we evaluated change content accuracy in the without-order manner.

In comparison experiments on the change captioning dataset CLEVR-Multi-Change, we evaluated sentence generation performance on evaluation metrics BLEU [42], CIDER [43], METEOR [44], and SPICE [45]. These evaluation metrics evaluate the similarities between generated sentences and ground truth sentences.

Implementation Details. Similar to the existing methods DUDA and MCCFormers, the input image features were obtained using a pre-trained ResNet101 model [46]. We set the learning rates of both the encoder and decoder to 0.0001. Detailed training parameters are shown in the supplementary material.

5.2. Results on OVT Dataset

Ablation study. We first conducted an ablation study on model designs. More specifically, we conducted experiments on the model structure of the node and edge decoder (transformer and GCN), loss function (cross-entropy loss and graph matching loss) for nodes and edges prediction, and head and layers (ranging from 1 to 2, and 2 to 4, respectively) for models. Here, we implemented GCN consisting of MLPs and adjusted the layers of the MLP structure.

We then evaluated the performance of all models for per-change accuracy, both with- and without-order setup on OVT dataset. The results are summarized in Table 2. We found that models with transformer-structured decoders outperformed those with GCN. Additionally, models with graph matching loss obtained higher accuracy compared with those that adopted cross-entropy losses. We attribute this result to the fact that the graph matching loss considers all possible combinations between ground truth and predicted results, thus making it possible to choose the best-matched one for the optimization process. Since the model that adopted the transformer (one layer and four heads) and graph matching loss obtained the highest accuracy, we mainly implemented this model (VTGen (trans)) for model comparison in the remaining experiments.

Quantitative Comparison. We show the experimental results of models and humans on the OVT dataset in Table 3.

Decoders	Loss	Layers	Heads	Accuracy (w)	Accuracy (w/o)
Transformer	CE	1	2	50.2	78.5
		1	4	49.6	78.8
		2	2	29.2	60.7
		2	4	41.1	72.8
	GM	1	2	50.8	79.1
		1	4	**52.4**	**80.5**
		2	2	36.6	68.6
		2	4	41.1	72.4
GCN	CE	1	-	47.2	76.8
		2	-	42.6	72.1
	GM	1	-	49.0	78.6
		2	-	45.2	75.7

Table 2. Ablation study on model and loss design. CE and GM stand for cross-entropy and graph matching loss, respectively.

500 examples were randomly selected for human evaluation. In terms of per-scene accuracy, which reports the correctness of recognizing all changes from each scene, our proposed method obtained 57.8%, thus outperforming the best performing previous method MCCFormers-S by +15.7%. For per-change accuracy, we found that all methods exhibited lower performance in the with-order setup than in the without-order setup. The proposed method obtained the highest performance among all per-change evaluations and achieved a significant performance gap of +15.1% compared to previous methods in with-order setup.

All existing methods directly generate a whole sequence to represent all changes and their orders. In the OVT dataset, the models need to classify all changes and their temporal orders from two images. Incorrect recognition of change contents will affect the change order recognition and vice-versa. The proposed method adopts the graph structure to recognize change contents and determine their orders separately, making it beneficial in the OVT dataset. Additionally, the proposed method explicitly considers the associations between changes, which is critical in distinguishing changes.

We also evaluated per-change type and change content (combinations of object attributes, positions, and change types) accuracy. We found that existing methods showed a relatively significant performance downgrade for add and move changes, which involve relatively more objects than delete changes. In contrast, our method obtained nearly the same levels of accuracy for all three change types, thus indicating that our method is better at distinguishing multiple changes. For detailed change content, the proposed method outperformed existing methods in determining the challenging location recognition while slightly outperforming existing methods in determining objects and change types.

Methods	Per-scene Accuracy	Per-change (w) Accuracy	Per-change (w) Recall	Per-change (w/o) Accuracy	Per-change (w/o) Recall	Per-change-type accuracy Add	Delete	Move	Change content accuracy Object	Location	Change
DUDA [15]	25.7	27.2	25.4	59.4	55.8	54.1	71.5	57.8	88.6	69.9	97.2
MCCFormers-D [16]	38.9	35.2	33.8	69.8	67.0	64.2	78.3	70.7	91.1	77.5	97.7
MCCFormers-S [16]	42.1	37.3	36.2	72.2	69.9	67.7	80.3	71.9	91.5	79.5	97.9
TranceNet [25]	30.0	30.6	28.4	64.9	60.3	59.8	73.8	64.8	89.7	73.9	97.0
VTGen (trans)	**57.8**	**52.4**	**51.0**	**80.5**	**78.0**	**79.1**	**82.0**	**81.1**	**92.0**	**86.3**	**98.2**
VTGen (GCN)	53.7	49.0	47.6	78.6	76.2	79.0	81.6	76.5	91.0	84.3	97.6
Human	76.6	70.1	70.9	91.9	93.0	89.4	90.0	92.7	97.7	96.5	98.2

Table 3. Model and human performance on the OVT dataset.

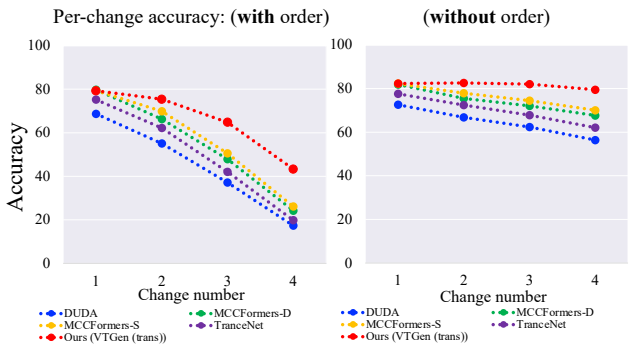

Figure 4. Average per-change accuracy for scenes with different change numbers on the OVT dataset.

Notably, there is still a huge performance gap between models and humans, especially in asynchronous changes and location prediction, indicating the existence of unsolved questions in this visual reasoning in scene dynamics.

Detailed Analysis on Transformation Steps. Next, we further evaluated model performance for scenes with different change numbers in Figure 4. Similar to results in Table 3, all methods obtained lower accuracy for the challenging with-order setup compared to the without-order setup. All methods obtained degraded performance for scenes with multiple changes, while our method maintained over 40% accuracy for four-change scenes in the with-order setup, outperforming existing methods by nearly 20%. In the without-order setup, our method obtained very similar results, having nearly 80% of accuracy for scenes with different change numbers in the without-order setups, thereby indicating its ability to distinguish multiple changes.

Qualitative Results. We show four result examples (with four changes each) of MCCFormers-S and MCCFormers-D [16] and VTGen (trans) in Figure 5. All three methods obtained correct results in example (a). In (a), all four changes could happen simultaneously, making it relatively less complicated to distinguish each change from the others. In examples (b,c,d), where there are specific temporal orders between changes, all methods obtained relatively lower performance in determining change contents, including object attributes and change locations. We also found that existing methods struggled in (c,d) by mistaking change numbers or change orders. On the contrary, our proposed method correctly predicted change orders in all four examples despite having incorrect object attribute prediction.

Methods	BLEU	CIDER	METEOR	SPICE
DUDA [15]	76.1	480.1	47.4	66.6
M-VAM [47]	62.9	338.1	41.3	55.9
MCCFormers-D [16]	82.3	539.3	52.1	71.7
MCCFormers-S [16]	83.3	523.3	51.5	70.0
VTGen (trans)	**85.2**	**584.9**	**54.0**	**81.3**

Table 4. Results on CLEVR-Multi-Change dataset [16].

5.3. Results on CLEVR-Multi-Change Dataset

To compare the proposed method with previous methods in the existing dataset, we evaluated our method in the CLEVR-Multi-Change dataset [16]. The dataset consists of scene pairs and sentences describing the changes within scene pairs. The CLEVR-Multi-Change dataset deals with multiple changes where all changes happen simultaneously without specific temporal orders. Because there are no temporal orders, we implemented our model without using the edge prediction part. Also, we facilitated sentence generation in our model by instantiating sentences from predicted change contents and sentence templates.

As shown in Table 4, the proposed method outperformed existing methods in all evaluation metrics. All these methods do not explicitly consider the relationships between different changes, making it challenging to distinguish changes from each other in scenes containing multiple changes. On the contrary, our proposed method explicitly determines different changes after correlating different changes, which is also beneficial in the CLEVR-Multi-Change dataset.

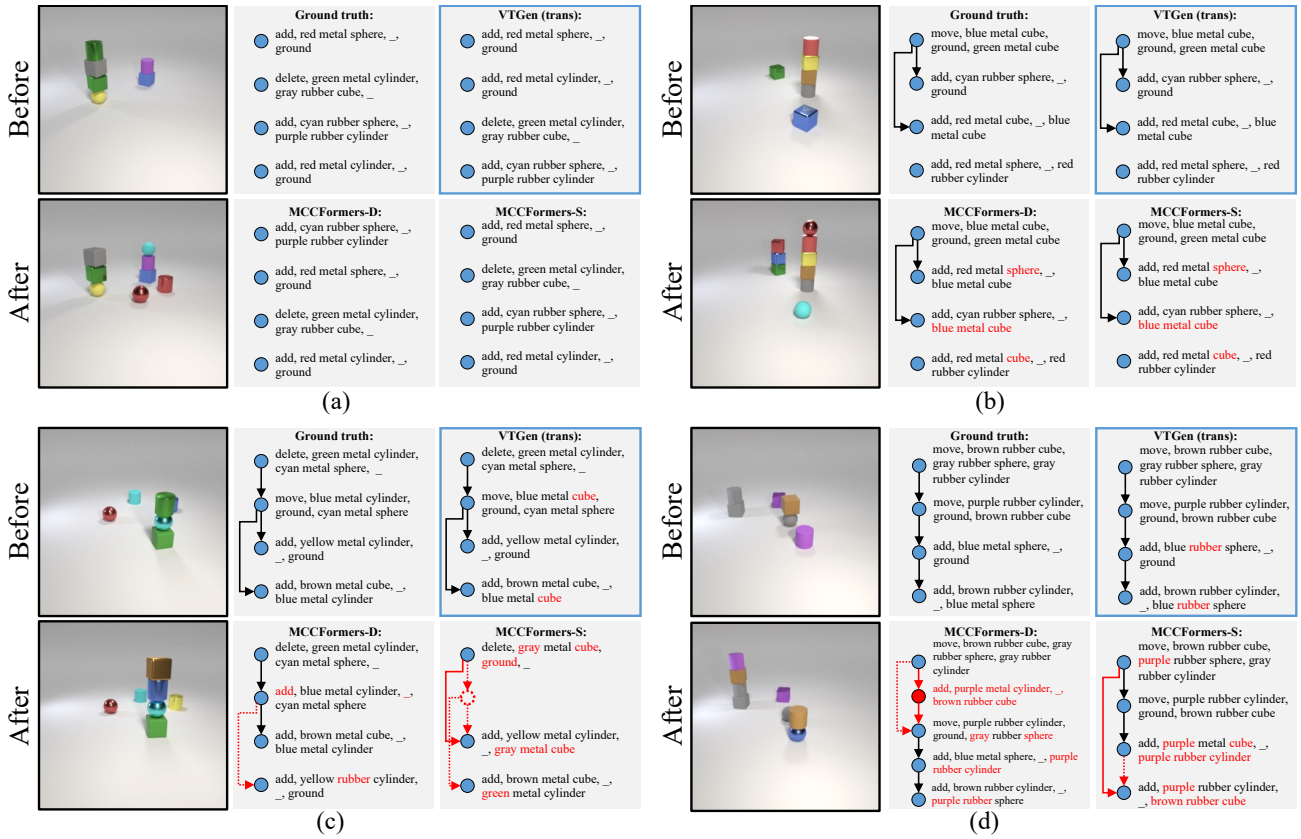

Figure 5. Experimental results on the OVT dataset. Incorrect and omitted predictions are marked in red and red dashed lines, respectively.

5.4. Analysis

The OVT dataset deals with both asynchronous and synchronous changes. Change content and order recognition affect each other mutually, and the associations between changes are critical in distinguishing one change from among many. Existing methods generate a sequence to represent the whole scene and neglect the change relationships. In contrast, our method adopts a graph representation for order-aware changes and explicitly associates changes, therefore enhancing its ability to distinguish changes in both OVT and an existing dataset. The experimental results show that the proposed model outperforms existing methods, especially in complicated situations, such as the with-order setup, scenes with more changes, challenging add or move change types, and change location recognition.

Also, the experimental results in Figure 4 and 5 reveal that there is still room for improvement, especially when the number of order-aware changes increases. Moreover, there is still a significant gap between model and human performance (Table 3). We plan to enhance model performance by disentangling change recognition from image pairs with scene state transformation recognition from textual information. Also, we only conducted experiments on synthetic datasets. We also consider further investigations of real scenarios and applications such as robotic manipulation.

6. Conclusion

This paper addresses a novel order-aware visual transformation task. Existing methods mainly focus on changes that occur synchronously without considering their underlying temporal orders. Change orders, although still less studied, are indispensable in discovering how changes occur and restoring scenes to their previous states and are essential in various applications, such as assembly operations. Hence, we facilitate the discussion of order-aware visual transformation by introducing a new dataset. Based on observation of the lack of understanding of change relationships and the unsuitable sentence representation used in existing methods, we proposed a method that explicitly associates changes and then generates a graph representation for describing order-aware changes. Our proposed method outperformed existing methods in both the proposed dataset and an existing dataset. However, we also found that there is still a significant performance gap between current models and human performance. We hope these results can call attention to resolving visual reasoning in scene dynamics.

References

[1] Justin Johnson, Bharath Hariharan, Laurens Van Der Maaten, Li Fei-Fei, C Lawrence Zitnick, and Ross Girshick. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. In *CVPR*, 2017. 1, 2, 3

[2] Drew A Hudson and Christopher D Manning. Gqa: A new dataset for real-world visual reasoning and compositional question answering. In *CVPR*, 2019. 1

[3] Madeleine Grunde-McLaughlin, Ranjay Krishna, and Maneesh Agrawala. Agqa: A benchmark for compositional spatio-temporal reasoning. In *CVPR*, 2021. 1

[4] Tristan Thrush, Ryan Jiang, Max Bartolo, Amanpreet Singh, Adina Williams, Douwe Kiela, and Candace Ross. Winoground: Probing vision and language models for visio-linguistic compositionality. In *CVPR*, 2022. 1

[5] Kexin Yi, Chuang Gan, Yunzhu Li, Pushmeet Kohli, Jiajun Wu, Antonio Torralba, and Joshua B Tenenbaum. Clevrer: Collision events for video representation and reasoning. In *ICLR*, 2019. 1

[6] Li Xu, He Huang, and Jun Liu. Sutd-trafficqa: A question answering benchmark and an efficient network for video reasoning over traffic events. In *CVPR*, 2021. 1

[7] Chi Zhang, Feng Gao, Baoxiong Jia, Yixin Zhu, and Song-Chun Zhu. Raven: A dataset for relational and analogical visual reasoning. In *CVPR*, 2019. 1

[8] Guangyu Robert Yang, Igor Ganichev, Xiao-Jing Wang, Jonathon Shlens, and David Sussillo. A dataset and architecture for visual reasoning with a working memory. In *ECCV*, 2018. 1

[9] Chi Zhang, Baoxiong Jia, Mark Edmonds, Song-Chun Zhu, and Yixin Zhu. Acre: Abstract causal reasoning beyond covariation. In *CVPR*, 2021. 1

[10] Jack Hessel, Jena D Hwang, Jae Sung Park, Rowan Zellers, Chandra Bhagavatula, Anna Rohrbach, Kate Saenko, and Yejin Choi. The abduction of sherlock holmes: A dataset for visual abductive reasoning. *arXiv preprint arXiv:2202.04800*, 2022. 1

[11] Chen Liang, Wenguan Wang, Tianfei Zhou, and Yi Yang. Visual abductive reasoning. In *CVPR*, 2022. 1

[12] Rowan Zellers, Yonatan Bisk, Ali Farhadi, and Yejin Choi. From recognition to cognition: Visual commonsense reasoning. In *CVPR*, 2019. 1

[13] Jae Sung Park, Chandra Bhagavatula, Roozbeh Mottaghi, Ali Farhadi, and Yejin Choi. Visualcomet: Reasoning about the dynamic context of a still image. In *ECCV*, 2020. 1

[14] Harsh Jhamtani and Taylor Berg-Kirkpatrick. Learning to describe differences between pairs of similar images. In *EMNLP*, 2018. 1, 2, 3, 4

[15] Dong Huk Park, Trevor Darrell, and Anna Rohrbach. Robust change captioning. In *ICCV*, 2019. 1, 2, 3, 4, 5, 7

[16] Yue Qiu, Shintaro Yamamoto, Kodai Nakashima, Ryota Suzuki, Kenji Iwata, Hirokatsu Kataoka, and Yutaka Satoh. Describing and localizing multiple changes with transformers. In *ICCV*, 2021. 1, 2, 3, 4, 5, 7

[17] Luca Weihs, Matt Deitke, Aniruddha Kembhavi, and Roozbeh Mottaghi. Visual room rearrangement. In *CVPR*, 2021. 1

[18] Yun-Chun Chen, Haoda Li, Dylan Turpin, Alec Jacobson, and Animesh Garg. Neural shape mating: Self-supervised object assembly with adversarial shape priors. In *CVPR*, 2022. 1, 2, 3

[19] De-An Huang, Suraj Nair, Danfei Xu, Yuke Zhu, Animesh Garg, Li Fei-Fei, Silvio Savarese, and Juan Carlos Niebles. Neural task graphs: Generalizing to unseen tasks from a single video demonstration. In *CVPR*, 2019. 1, 3

[20] Peter Anderson, Qi Wu, Damien Teney, Jake Bruce, Mark Johnson, Niko Sünderhauf, Ian Reid, Stephen Gould, and Anton Van Den Hengel. Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments. In *CVPR*, 2018. 1

[21] Howard Chen, Alane Suhr, Dipendra Misra, Noah Snavely, and Yoav Artzi. Touchdown: Natural language navigation and spatial reasoning in visual street environments. In *CVPR*, 2019. 1

[22] Karl DD Willis, Pradeep Kumar Jayaraman, Hang Chu, Yunsheng Tian, Yifei Li, Daniele Grandi, Aditya Sanghi, Linh Tran, Joseph G Lambourne, Armando Solar-Lezama, et al. Joinable: Learning bottom-up assembly of parametric cad joints. In *CVPR*, 2022. 2, 3

[23] Yichen Li, Kaichun Mo, Lin Shao, Minhyuk Sung, and Leonidas Guibas. Learning 3d part assembly from a single image. In *ECCV*, 2020. 2, 3

[24] Guanqi Zhan, Qingnan Fan, Kaichun Mo, Lin Shao, Baoquan Chen, Leonidas J Guibas, Hao Dong, et al. Generative 3d part assembly via dynamic graph learning. In *NeurIPS*, 2020. 2, 3

[25] Xin Hong, Yanyan Lan, Liang Pang, Jiafeng Guo, and Xueqi Cheng. Transformation driven visual reasoning. In *CVPR*, 2021. 2, 5, 7

[26] Rareş Ambruş, Nils Bore, John Folkesson, and Patric Jensfelt. Meta-rooms: Building and maintaining long term spatial models in a dynamic world. In *IROS*, 2014. 2, 3

[27] Edith Langer, Bram Ridder, Michael Cashmore, Daniele Magazzeni, Michael Zillich, and Markus Vincze. On-the-fly detection of novel objects in indoor environments. In *ROBIO*, 2017. 2, 3

[28] Rongjun Qin and Armin Gruen. 3d change detection at street level using mobile laser scanning point clouds and terrestrial images. *ISPRS Journal of Photogrammetry and Remote Sensing*, 90, 2014. 2, 3

[29] Pablo F Alcantarilla, Simon Stent, German Ros, Roberto Arroyo, and Riccardo Gherardi. Street-view change detection with deconvolutional networks. *Autonomous Robots*, 42(7), 2018. 2, 3

[30] Yue Qiu, Yutaka Satoh, Ryota Suzuki, Kenji Iwata, and Hirokatsu Kataoka. Indoor scene change captioning based on multimodality data. *Sensors*, 20(17), 2020. 2

[31] Justin Johnson, Ranjay Krishna, Michael Stark, Li-Jia Li, David Shamma, Michael Bernstein, and Li Fei-Fei. Image retrieval using scene graphs. In *CVPR*, 2015. 3

[32] Yuren Cong, Michael Ying Yang, and Bodo Rosenhahn. Reltr: Relation transformer for scene graph generation. *arXiv preprint arXiv:2201.11460*, 2022. 3

[33] Rongjie Li, Songyang Zhang, and Xuming He. Sgtr: End-to-end scene graph generation with transformer. In *CVPR*, 2022. 3

[34] Kevin Ellis, Daniel Ritchie, Armando Solar-Lezama, and Josh Tenenbaum. Learning to infer graphics programs from hand-drawn images. In *NeurIPS*, 2018. 3

[35] Yunchao Liu and Zheng Wu. Learning to describe scenes with programs. In *ICLR*, 2019. 3

[36] Jiajun Wu, Erika Lu, Pushmeet Kohli, Bill Freeman, and Josh Tenenbaum. Learning to see physics via visual de-animation. In *NeurIPS*, 2017. 3

[37] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017. 5

[38] Max Welling and Thomas N Kipf. Semi-supervised classification with graph convolutional networks. In *ICLR*, 2017. 5

[39] Justin Johnson, Agrim Gupta, and Li Fei-Fei. Image generation from scene graphs. In *CVPR*, 2018. 5

[40] Johanna Wald, Helisa Dhamo, Nassir Navab, and Federico Tombari. Learning 3d semantic scene graphs from 3d indoor reconstructions. In *CVPR*, 2020. 5

[41] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *ECCV*, 2020. 5

[42] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *ACL*, 2002. 6

[43] Ramakrishna Vedantam, C Lawrence Zitnick, and Devi Parikh. Cider: Consensus-based image description evaluation. In *CVPR*, 2015. 6

[44] Satanjeev Banerjee and Alon Lavie. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *ACL workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, 2005. 6

[45] Peter Anderson, Basura Fernando, Mark Johnson, and Stephen Gould. Spice: Semantic propositional image caption evaluation. In *ECCV*, 2016. 6

[46] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 6

[47] Xiangxi Shi, Xu Yang, Jiuxiang Gu, Shafiq Joty, and Jianfei Cai. Finding it at another side: A viewpoint-adapted matching encoder for change captioning. In *ECCV*, 2020. 7