# Transformation Driven Visual Reasoning

Xin Hong[1,2] Yanyan Lan[3,*] Liang Pang[1,2] Jiafeng Guo[1,2] Xueqi Cheng[1,2]
[1] CAS Key Laboratory of Network Data Science and Technology,
Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China
[2] University of Chinese Academy of Sciences, Beijing, China
[3] Institute for AI Industry Research, Tsinghua University, Beijing, China
{hongxin19b, pangliang, guojiafeng, cxq}@ict.ac.cn lanyanyan@tsinghua.edu.cn

## Abstract

*This paper defines a new visual reasoning paradigm by introducing an important factor, i.e. transformation. The motivation comes from the fact that most existing visual reasoning tasks, such as CLEVR in VQA, are solely defined to test how well the machine understands the concepts and relations within static settings, like one image. We argue that this kind of* ***state driven visual reasoning*** *approach has limitations in reflecting whether the machine has the ability to infer the dynamics between different states, which has been shown as important as state-level reasoning for human cognition in Piaget's theory. To tackle this problem, we propose a novel* ***transformation driven visual reasoning*** *task. Given both the initial and final states, the target is to infer the corresponding single-step or multi-step transformation, represented as a triplet (object, attribute, value) or a sequence of triplets, respectively. Following this definition, a new dataset namely TRANCE is constructed on the basis of CLEVR, including three levels of settings, i.e. Basic (single-step transformation), Event (multi-step transformation), and View (multi-step transformation with variant views). Experimental results show that the state-of-the-art visual reasoning models perform well on Basic, but are still far from human-level intelligence on Event and View. We believe the proposed new paradigm will boost the development of machine visual reasoning. More advanced methods and real data need to be investigated in this direction. The resource of TVR is available at* https://hongxin2019.github.io/TVR.

## 1. Introduction

Visual reasoning is the process of solving problems on the basis of analyzing the visual information, which goes well beyond object recognition [11, 20, 30, 31]. Though the task is easy for the human, it is tremendously difficult for vision systems, because it usually requires higher-order cognition and reasoning about the world. In recent years, several visual reasoning tasks have been proposed and attract a lot of attention in the community of computer vision, machine learning, and artificial intelligence. For example, the most representative visual question answering (VQA) tasks, such as CLEVR [18], define a question answering paradigm to test whether machines have spatial, relational, and other reasoning abilities for a given image. Visual entailment tasks such as NLVR [32, 33] ask models to determine whether a sentence is true about the states of two images. Visual commonsense reasoning tasks, such as VCR [42], further require the model to provide a rationale explaining why its answer is right.

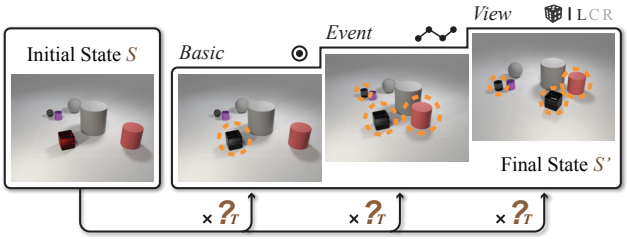


Figure 1. Illustration of three settings in TRANCE. **Basic:** Find the single-step transformation between the initial and final state. **Event:** Find the multi-step transformation between two states. **View:** Similar to Event but the view angle of the final state is randomly chosen from *Left*, *Center* (Default), and *Right*.

We can see that these visual reasoning tasks are all defined at *state* level. For example, the language descriptions in NLVR as well as the questions and answers in VQA and VCR are just related to the concepts or relations within states, i.e. an image, or two images. We argue that this kind of *state driven visual reasoning* fails to test the ability of reasoning dynamics between different states. Take two images as an example. In the first image, there is a cat on a

*Corresponding author.

tree, and in the second image, the same cat is under the tree. It is natural for a human to reason that the cat jumps down the tree after analyzing the two images. Piaget's cognitive development theory [29] describes the dynamics between states as transformation, and tells that human intelligence must have functions to represent both the transformational and static aspects of reality. In addition, transformation is the key to tackle some more complicated tasks such as storytelling [13] and visual commonsense inference [28]. Though these tasks are closer to reality, they are too complicated to serve as a good testbed for transformation based reasoning. Because many other factors like representation and recognition accuracy may have some effects on the performance. Therefore, it is crucial to define a specific task to test the transformation reasoning ability.

In this paper, we define a novel *transformation driven visual reasoning* (TVR) task. Given the initial and final states, like two images, the goal is to infer the corresponding single-step or multi-step transformation. Without loss of generality, in this paper, transformations indicate changes of object attributes, so a single-step and multi-step transformation are represented as a triplet *(object, attribute, value)* and a sequence of triplets, respectively.

Following the definition of TVR, we construct a new dataset called TRANCE (Transformation on CLEVR), to test and analyze how well machines can understand the transformation. TRANCE is a synthetic dataset based on CLEVR [18], since it is better to first study TVR in a simple setting and then move to more complex real scenarios, just like people first study VQA on CLEVR and then generalize to more complicated settings like GQA. CLEVR has defined five types of attributes, i.e. color, shape, size, material, and position. Therefore, it is convenient to define the transformation for each attribute, e.g. the color of an object is changed from red to blue. Given the initial and final states, i.e. two images, where the final state is obtained by applying a single-step or multi-step transformation on the initial state, a learner is required to well infer such transformation. To facilitate the test for different reasoning levels, we design three settings, i.e. Basic, Event, and View. Basic is designed for testing single-step transformation. Event and View are designed for more complicated multi-step transformation reasoning, where the difference is that View further considers variant views in the final state. Figure 1 gives an example of three different settings.

In the experiments, we would like to test how well existing reasoning techniques [14, 19] work on this new task. However, since these models are mainly designed for existing reasoning tasks, they cannot be directly applied to TRANCE. To tackle this problem, we propose a new encoder-decoder framework named TranceNet, specifically for TVR. With TranceNet, existing techniques can be conveniently adapted to TVR. We test several different encoders, e.g. ResNet [12], Bilinear-CNN [22] and DUDA [27]. While for the decoder, an adapted GRU [6] network is used to employ the image features and additional object attributes from TRANCE to predict the transformation, which is a sequence of triplets. Experimental results show that deep models perform well on Basic, but are far from human's level on Event and View, demonstrating high research potentials in this direction.

In summary, the contributions of our work include: 1) the definition of a new visual reasoning paradigm, to learn the dynamics between different states, i.e. transformation; 2) the proposal of a new dataset called TRANCE, to test three levels of transformation reasoning, i.e. Basic, Event, and View; 3) experimental studies of the existing SOTA reasoning techniques on TRANCE show the challenges of the TVR and some insights for future model designs.

## 2. Related Works

The most popular visual reasoning task is VQA. Questions in the earliest VQA dataset [2, 10, 43] are usually concerned about the category or attribute of objects. Recent VQA datasets have improved the requirements on image understanding by asking more complex questions, e.g. Visual7W [44], CLEVR [18], OK-VQA [25], and GQA [15]. There are two other forms of visual reasoning tasks that need to be mentioned. Visual entailment tasks, such as NLVR [32, 33], and SNLI-VE [39, 40], ask models to determine whether a given description is true about a visual input. Visual commonsense reasoning [35, 42] tasks require to use commonsense knowledge [34] to answer questions. It is meaningful to solve these tasks which require various reasoning abilities. However, all the above tasks are defined to reason within a single state, which ignore the dynamics between different states.

Recently, several visual reasoning tasks have been proposed to consider more than one state. For example, CATER [9] tests the ability to recognize compositions of object movements. While our target is to evaluate transformations, and our data contains more diverse transformations rather than just moving. Furthermore, CATER along with other video reasoning tasks such as CLEVRER [41] and physical reasoning tasks [3, 4] are usually based on dense states, which make the transformations hard to define and evaluate. Before moving to these complex scenarios, our TVR provides a simpler formulation by explicitly defining the transformations between two states, which is more suitable for testing the ability of transformation reasoning. CLEVR-Change [27] is the most relevant work, which requires to caption the change between two images. The novelty is that TVR isolates the ability to reason about state-transition dynamics and supports a more thorough evaluation than captioning. Furthermore, CLEVR-Change only focuses on single-step transformations.

The concept of transformation has also been mentioned in many other fields. In [16, 21, 26], transformations are used to learn good attribute representations to improve the classification accuracy. In [1, 8, 24, 36, 45], object or environment transformations are detected to improve the performance of action recognition. However, those works in attribute learning and action recognition fields only consider single step transformation, thus not appropriate for testing a complete transformation reasoning ability. Some people may feel that procedure planning [5] has a similar task formulation to TVR and they are closer to reality. However, actions in planning data are usually very sparse. More importantly, both high-quality recognition and transformation reasoning are crucial to well model the task. As they are coupled together, it is not appropriate to use such tasks to evaluate the ability of transformation reasoning. That is why in this paper we use a more simple yet effective way to define the TVR task.

## 3. The Definition of TVR

TVR (Transformation driven Visual Reasoning) is a visual reasoning task that aims at testing the ability to reason the dynamics between states. Formally, we denote the state and transformation space as $\mathcal{S}$ and $\mathcal{T}$ respectively. The process of transforming the initial state into the final state can be illustrated as a function $f : \mathcal{S} \times \mathcal{T} \rightarrow \mathcal{S}$. Therefore, the task of TVR can be defined as inferring the transformation $T \in \mathcal{T}$ given both the initial state $S \in \mathcal{S}$ and final state $S' \in \mathcal{S}$. The space of transformation is usually very large, e.g. any changes of pixel value can be treated as a transformation. Therefore, without loss of generality, we define the atomic transformation as an attribute-level change of an object, represented as a triplet $t = (o, a, v)$, which means the object $o$ with the attribute $a$ is changed to the value $v$. For example, the color of an object is changed to blue. And further taking the order of atomic transformations into account, the transformation can be formalized as a sequence of atomic transformations, denotes as $T = \{t_1, t_2, \ldots, t_n\}$, where $n$ is the number of atomic transformations.

We make a distinction between the *single-step* ($n = 1$) and *multi-step* ($n \geq 1$) transformation setting because they can be evaluated in different ways to reflect different levels of transformation reasoning abilities. For single-step problems, we can directly compare the prediction $\hat{T}$ with the ground-truth $T$ to obtain overall accuracy, as well as the fine-grained accuracy of each element in the triplet $(o, a, v)$. With these metrics, it is convenient to know how well a learner understands atomic transformations and to analyze why a learner performs not well. However, multi-step transformation problems cannot be evaluated in this way. This is because atomic transformations sometimes are independent so that the order of them can be varying, which makes the answer not unique anymore. For example, the procedures

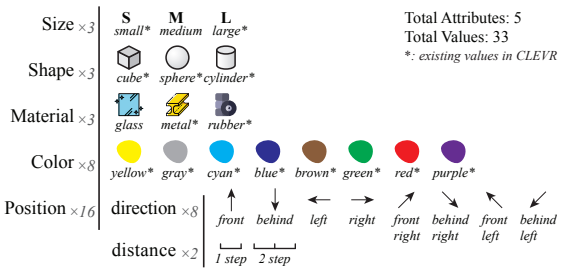


Table 1. Attributes and values in TRANCE.

of cooking can not be disrupted but changing the order of cooking different dishes makes no difference to the final result. To tackle this multi-solution problem, we consider the reconstruction error as the evaluation metric. Specifically, the predicting transformation $\hat{T}$ is first applied to the initial state $S$ to obtain the predicted final state $\hat{S}'$. Then the transformed $\hat{S}'$ is compared with the ground-truth final state $S'$ to decide whether the predicting transformation is correct and how far is the prediction from a correct transformation. Therefore, the evaluation of multi-step problems focuses more on the ability to find all atomic transformations and a feasible order to arrange them.

With this definition, most existing state driven visual reasoning tasks can be extended to the corresponding transformation driven ones. For example, the VQA task, such as CLEVR, can be extended to ask the transformation between two given images, with answers as the required transformation. In the extension of NLVR, the task becomes to determine whether a sentence describing the transformation is true about the two images, e.g. the color of the bus is changed to red. Since TVR itself is defined as an interpretation task, we do not need any further rational explanations, and the extension of VCR will stay the same as NLVR. We can see that the intrinsic reasoning target of these tasks is the same, that is to infer the correct transformation. While the difference lies in the manifestation.

## 4. The TRANCE Dataset

In this paper, we extend CLEVR by asking a uniform question, i.e. what is the transformation between two given images, to test the ability of transformation reasoning. This section introduces how the TRANCE (Transformation on CLEVR) dataset built following the definition of TVR.

### 4.1. Dataset Setups

We choose CLEVR [18] to extend because CLEVR defines multiple object attributes, which can be changed conveniently. With the powerful Blender [7] engine used by CLEVR, we are able to collect over 0.5 million samples with only computational costs.

According to our definition, an atomic transformation is

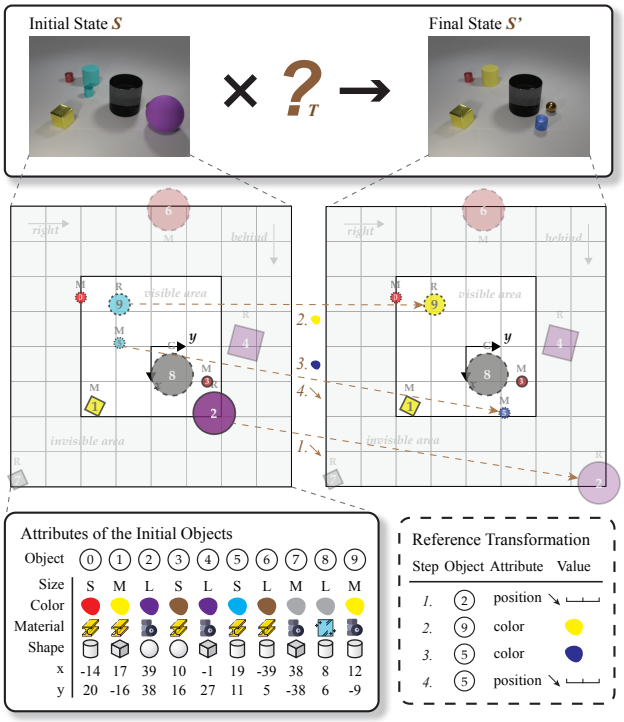


Figure 2. An example from the Event setting.

represented as a triplet $(o, a, v)$. In the following, we introduce attribute, value, and object one by one to demonstrate how to ground these factors.

The setup of the attribute is exactly the same as CLEVR. There are five attributes for each object, i.e. size, color, shape, material, and position.

The value in an atomic transformation is defined corresponding to the concerned attribute, as shown in Table 1. The values of the attributes except for position are similar to the default setting of CLEVR. Medium size and glass material are added to enrich the values. The value of position is carefully designed since it can be infinite in the space $\mathbb{R}^2$. To reduce the computation, we replace the absolute value of the position with a relative one by using direction and step to represent the value of a position transformation. We consider eight values for the variable direction, as shown in the Table 1. Then we define a coordinate system, in which $x$ and $y$ are both restricted to $[-40, 40]$, and objects can only be placed on integer coordinates. The variable step can be valued as 1 or 2, where 1 step equals 10 in our coordinate system. Except for normal moving action, we are also interested in whether the vision system could understand actions like moving in and moving out, so the plane is split into the visible area and the invisible area as shown in the middle two images of Figure 2, and the moving in and out operations can be defined correspondingly. To be reasonable, objects shouldn't be overlapped and moved out of the plane during transformation.

The setup of the object is basically the same as CLEVR. The only problem is how to represent an object in the answer. Existing methods such as CLEVR and CLEVR-Change use text which has ambiguity issues making the evaluation unreliable, while CLEVR-Ref+ [23] employs bounding boxes which is specific but requires the additional ability of detection. Therefore, we propose a simple method that is specific and easy to evaluate by providing the attributes of the initial objects, as shown in Figure 2. In this way, an object can be referred to with the assigned number. Note machines still need to perform their own recognition to align objects in images with given attributes.

To generate TRANCE, the first step is the same as CLEVR, which is randomly sampling a scene graph. According to the scene graph, CLEVR then generates questions and answers with a functional program and renders the image with Blender. Different from CLEVR, the next step in TRANCE generation becomes randomly sampling a sequence of atomic transformations ($n \in \{1, 2, 3, 4\}$), which is called as the reference transformation, to transform the initial scene graph to the final scene graph. At last, two scene graphs are rendered into images ($h : 240 \times w : 320$). The attributes of the initial objects can be easily obtained from the initial scene graph.

To reduce the potential bias in TRANCE, we carefully control the sampling process of scene graph and transformation by balancing several factors. In scene graph sampling, we balance objects' attributes and the number of visual objects in the initial state. In transformation sampling, the length of the transformation, the object number, n-gram atomic transformation, and the move type are all balanced. Throughout all elements, N-gram atomic transformation is the hardest to be balanced and it refers to the sub-sequence of atomic transformations with the length of $n$. By balancing these factors, we reduce the possibility that a learner utilizes statistics features in the data to predict answers. In the supplementary material, we show the statistics of the dataset and our balancing method in detail.

### 4.2. Three Levels of Settings

To facilitate the study on different levels of transformation reasoning, we design three settings, i.e. Basic, Event, and View. Basic is designed for single-step transformation and Event is for multi-step transformation. To further evaluate the ability of reasoning transformation under a more real condition, we extend Event with variant views to propose View. Figure 1 compares three different settings, more examples can be found in the supplementary material.

**Basic.** Basic is set as the first simple problem to mainly test how well a learner understands atomic transformations. The target of Basic is to infer the single-step transformation between the initial and final states. That is, given a pair of images, the task is to find out which attribute $a$ of which

object $o$ has been changed to which value $v$. We can see that this task is similar to the previous game 'Spot the Difference' [17], in which the player is asked to point out the differences between two images. However, Basic is substantially different from the game. The game focuses on the pixel level differences while Basic cares about the object level differences. Therefore, Basic can be viewed as a more advanced visual reasoning task than the game.

**Event.** It is obviously not enough to consider only the single-step transformation. In reality, it is very common that multi-step transformation exists between two states. Therefore, we construct this multi-step transformation setting to test whether machines can handle this situation. The number of transformations between the two states is randomly set from 1 to 4. The goal is to predict a sequence of atomic transformations that could reproduce the same final state from the initial state. To resolve this problem, a learner must find all atomic transformations and arrange them with a feasible order. Compared with Basic, it is possible to have multiple transformations, which improves the difficulty of finding them all. Meanwhile, the order is essential in the Event setting because atomic transformations may be dependent. For example, in Figure 2, two moving steps, i.e. 1st step and 4th step, cannot be exchanged, otherwise, object 5 and object 2 will overlap.

**View** The view angle of Basic and Event is fixed, which is not the case in real applications. To tackle this problem, we extend the Event setting to View, by capturing two states with cameras in different positions. In practice, for simplicity but without loss of generality, we set three cameras, placed in the left, center, and right side of the plane. The initial state is always captured by the center camera, while for the final state, images are captured with all three cameras. Thus, for each sample, we obtain three pairs for training, validation, and testing with the same initial state but different views of final states. In this way, we are capable to test whether a vision system can understand object-level transformation with variant views.

### 4.3. Evaluation Metrics

For the single-step transformation setting, i.e. Basic, the answer is unique. Therefore, we can evaluate the performance by directly comparing the prediction with the reference transformation, which is also the ground-truth transformation. Specifically in this paper, we consider two types of accuracy. The first one is fine-grained accuracy corresponds to three elements in transformation triplet, including object accuracy, attribute accuracy, and value accuracy, denoted as *ObjAcc*, *AttrAcc*, and *ValAcc* respectively. The other one is the overall accuracy, which only counts the absolutely correct transformation triplets, denoted as *Acc*.

For multi-step transformation settings, i.e Event and View, it is not suitable to use the above evaluation metrics, because the answers may not be unique. Therefore, the predicting atomic transformation sequence is evaluated by checking whether it could reproduce the same final state as the reference transformation. Specifically, we first obtain the corresponding final state $\hat{S}'$ by applying the predicting transformation $\hat{T}$ to the initial state $S$, i.e. $S \times \hat{T} \rightarrow \hat{S}'$. Then a *distance* is computed by counting the attribute level difference between $\hat{S}'$ and the ground-truth final state $S'$. To eliminate the influence of different transformation lengths on distance, we normalize it by the length of the reference transformation to get a *normalized distance*. Averaging these two metrics on all samples, we obtain *AD* and *AND*. We further consider the overall accuracy denoted as *Acc* when the distance equals to zero. In addition, we are interested to see without considering the order, whether all atomic transformations are found, by omitting all constraints such as no overlapping to compute the loose accuracy, which denotes as *LAcc*. At last, to measure the ability of assigning the right order when all atomic transformations have been found, the error of order $EO = (LAcc - Acc)/LAcc$ is computed. In summary, five evaluation metrics are used in multi-step transformation settings, i.e. *AD*, *AND*, *LAcc*, *Acc*, and *EO*.

In the evaluation of multi-step transformation problems, an important step is to obtain the predicting final state $\hat{S}'$ by applying the predicting transformation $\hat{T}$ to the initial state $S$. This function has already been implemented in the previously mentioned data generation system and we reuse it in our multi-step transformation evaluation system. Except for the usage of evaluation, this evaluation system can also be used to generate signals as rewards for reinforcement learning, which is explored in Section 5.2.

## 5. Experiments

In this section, we show our experimental results on the three settings of TRANCE, i.e. Basic, Event, and View. We also conduct analyses to show some insights about machines' ability of reasoning transformation.

### 5.1. Models

Firstly, we would like to test how well existing methods work on this new task. However, since the inputs and outputs of TVR are quite different from existing visual reasoning tasks, existing methods like [14, 19] cannot be directly applied. So we design a new encoder-decoder style framework named TranceNet. As Figure 3 shows, start from the input image pairs, an encoder first extracts features, and then a GRU based decoder is employed to generate transformation sequences. The following of this section briefly introduces our TranceNet framework while the implementation details can be found in the supplementary material. To compare with the human, for each of the three settings, we also collect results of 100 samples in total. These results

come from 10 CS Ph.D. candidates who are familiar with our problems and the testing system.

**Encoder.** The goal of an encoder is to extract effective features from image pairs, which are mainly associated with the difference between the two states. There are two ways to extract these features, namely single-stream way and two-stream way. Single-stream way directly inputs two images into a network to extract features, while two-stream way first separately extracts image features and then interacts them in feature-level. In this paper, we evaluate six methods. In terms of the single-stream way, we test two networks, i.e. Vanilla **CNN** and **ResNet** [12], combined with two preprocessing strategies, i.e. subtraction ($-$) and concatenation ($\oplus$). For example, we use ResNet$_{\oplus}$ to represent a ResNet fed with concatenated image pairs. Another two methods are Bilinear CNN (**BCNN** [22]) and a recently proposed method called **DUDA** [27], which operate as the two-stream way. BCNN is a classical model for fine-grained image classification to distinguish categories with small visual differences. DUDA is originally proposed for change detection and captioning. The main difference between BCNN and DUDA lies in the way of feature-level interaction.

**Decoder.** The decoder is used to output a feasible transformation sequence from the extracted image features. All six encoders share the same modified GRU [6] network, which is a commonly used technique for sequence generation. As shown in Figure 3, the major difference between our GRU network and a standard one lies in the additional classifiers. A classifier unit accepts attributes of the initial objects and the current hidden state from the GRU cell, and then outputs the object and value of the current step. In detail, at a certain step, an object vector is first computed from the hidden state. Then the object number is obtained by matching the object vector with the initial objects using cosine similarity. Finally, the most similar object vector from initial objects and the hidden state are used to predict the value. In TRANCE, attributes are implied by values, for example, blue indicates that the attribute is color, so that the output of a classifier does not explicitly include an attribute.

Since all these models share the same decoder, we denote these models by their encoders' names hereafter.

**Training.** The loss function of a single sample consists of two cross-entropy losses for object and value respectively, which can be represented as:

$$\mathcal{L}(T, \hat{T}) = -\frac{1}{n}\sum_{i=1}^{n}(t_i^o \cdot \log \hat{t}_i^o + t_i^v \cdot \log \hat{t}_i^v), \qquad (1)$$

where $n$ is the transformation length of the sample, $t_i^o$ and $t_i^v$ denote the object and value in the $i$-th step of transformation. The training loss is the average of losses over all training examples. During training, we use teacher forcing [38] for faster convergence in two parts of TranceNet. Firstly, at each step, we follow the practice in sequence learning such as machine translation by using the object and value from the given reference transformation as the inputs of the GRU unit. Additionally, in the classifier, objects from reference transformations are used to predict values.

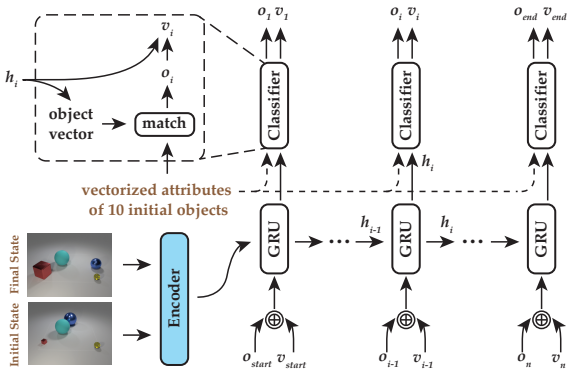

Figure 3. The architecture of TranceNet.

## 5.2. Results on Three Settings

In this section, we first present our experimental results on the three settings of TRANCE, and then provide some in-depth analyses on the results.

Six models are tested in the Basic setting of TRANCE, i.e. CNN$_{-}$, CNN$_{\oplus}$, ResNet$_{-}$, ResNet$_{\oplus}$, BCNN, DUDA. From the results in the left part of Table 2, we can see that all models perform quite well, in the sense that the performance gap between these models and the human is not very large. Comparing these models, both versions of ResNet perform better than BCNN and DUDA. As we mentioned before, CNN and ResNet are single-stream methods while BCNN and DUDA are two-stream methods. Since the model size of ResNet, BCNN, and DUDA is similar, we can conclude that the single-stream way is better than the two-stream way on the Basic setting. Further checking the fine-grained accuracy, we can see this gap comes from the ability to find the correct objects and values, while all models are good at distinguishing different attributes.

The experimental results on Event are shown in the middle part of Table 2. We can see that this task is very challenging for machines, since there is an extremely big performance gap between models and the human. That is because the answer space rises exponentially when the number of steps increases. In our experiments, the size of answer space is $\sum_{i=1}^{4}(33 \times 10)^i$, about 11.86 billion. The performance (e.g. Acc) gap between CNN and ResNet models becomes larger from Basic to Event, which suggests larger encoders have advantages in extracting sufficient features to decode transformation sequences.

We also employ reinforcement learning to train models. Specifically, the evaluation system introduced in Section 4.3 can provide signals include the *correctness* of a prediction and the *distance* of a prediction to the correct

| Model | Basic | | | | Event | | | | View | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | ObjAcc↑ | AttrAcc↑ | ValAcc↑ | Acc↑ | AD↓ | AND↓ | LAcc↑ | Acc↑ | AD↓ | AND↓ | LAcc↑ | Acc↑ |
| CNN$_-$ | 0.9584 | 0.9872 | 0.9666 | 0.9380 | 1.5475 | 0.5070 | 0.4587 | 0.4442 | 2.2376 | 0.8711 | 0.2344 | 0.2286 |
| CNN$_\oplus$ | 0.9581 | 0.9889 | 0.9725 | 0.9420 | 1.4201 | 0.4658 | 0.4981 | 0.4838 | 2.2517 | 0.8764 | 0.2350 | 0.2285 |
| ResNet$_-$ | 0.9830 | 0.9969 | **0.9935** | 0.9796 | 1.0974 | **0.3417** | 0.5972 | 0.5750 | **1.1068** | 0.3749 | 0.5484 | 0.5272 |
| ResNet$_\oplus$ | **0.9852** | **0.9980** | 0.9928 | **0.9810** | **1.0958** | 0.3469 | **0.6019** | **0.5785** | 1.1148 | **0.3731** | **0.5525** | **0.5305** |
| BCNN | 0.9705 | 0.9950 | 0.9788 | 0.9571 | 1.1081 | 0.3582 | 0.5746 | 0.5560 | 1.2633 | 0.4395 | 0.4977 | 0.4784 |
| DUDA | 0.9453 | 0.9888 | 0.9692 | 0.9320 | 1.5261 | 0.4975 | 0.5025 | 0.4856 | 1.5352 | 0.5242 | 0.4746 | 0.4590 |
| Human | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 0.3700 | 0.1200 | 0.8300 | 0.8300 | 0.3200 | 0.0986 | 0.8433 | 0.8433 |

Table 2. Model and human performance on Basic, Event, and View.

| Model | AD↓ | AND↓ | LAcc↑ | Acc↑ |
|---|---|---|---|---|
| ResNet$_\oplus$ | 1.0958 | 0.3469 | 0.6019 | 0.5785 |
| + *corr* | 1.0579 | 0.3316 | 0.6215 | 0.5978 |
| + *dist* | 1.0528 | 0.3319 | 0.6180 | 0.5938 |
| + *corr & dist* | **1.0380** | **0.3251** | **0.6230** | **0.6001** |

Table 3. Results of ResNet$_\oplus$ trained using REINFORCE [37] with different rewards on Event.

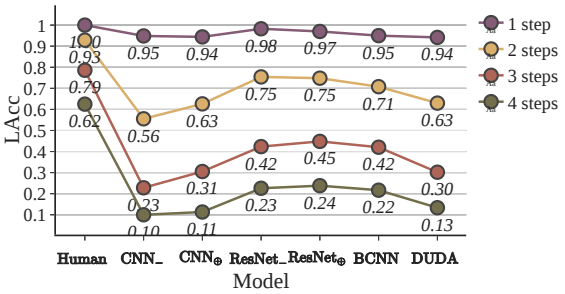


Figure 4. Results on Event with respect to different steps.

| Encoder | LAcc↑ | Acc↑ | EO↓ |
|---|---|---|---|
| Random (avg. of 100) | 1.0000 | 0.4992 | 0.5008 |
| CNN$_-$ | 0.2276 | 0.2067 | **0.0915** |
| CNN$_\oplus$ | 0.2596 | 0.2244 | 0.1358 |
| ResNet$_-$ | 0.3478 | 0.2997 | 0.1382 |
| ResNet$_\oplus$ | **0.3862** | **0.3205** | 0.1701 |
| BCNN | 0.2949 | 0.2612 | 0.1141 |
| DUDA | 0.2500 | 0.2147 | 0.1410 |
| Human | 0.7273 | 0.7273 | 0.0000 |

Table 4. Results on 7.8% order sensitive samples from Event.

one. These signals are able to be used as rewards in REINFORCE [37] algorithm to further train ResNet$_\oplus$ models. Table 3 shows that all three rewards significantly improve the performance, and the difference among them is small.

The right part of Table 2 shows the results on the View setting. While humans are insensitive to view variations, the performances of all deep models drop sharply from Event to View. Among these models, the most robust one is DUDA. From the fact that BCNN has a similar two-stream architecture but performs worse, we can see that DUDA's way of directly interacting two state features is more effective for tackling the view variation.

### 5.3. Detailed Analysis on Event and View

According to the above experimental results, the performances on Event and View are not good. So we conduct some detailed analysis to help understand the task and provide some insights for future model designs.

Firstly, we analyze the effect of transformation sequence length on Event, which is the main factor to make performance worse than Basic. Specifically, we separate all test samples into four groups based on their lengths, i.e. samples with $k$-step transformation ($k = 1, 2, 3, 4$). Then we plot the LAcc for each group in Figure 4. From the results, we can see that both human and deep models work quite well when the length is short, e.g. 1. As the length increases, humans still have the ability to well capture the complicated transformations. However, the deep models decline sharply. Take CNN$_-$ as an example, the performances for the four different groups are 95%, 56%, 23%, and 10%. These results indicate that future studies should focus more on how to tackle transformations with long steps.

Then we analyze the effect of the order on Event, which is another important factor in this data. According to our statistics, there are about 7.8% test samples[1] exist certain permutations that violate our constraints such as no overlapping. That is to say, these samples are order sensitive. Even if a model is able to find all correct atomic transformations, the result still could be wrong without carefully considering an order to arrange them. Table 4 shows the results on these order sensitive samples, where EO is directly defined to measure the influence of order, LAcc and Acc are just listed for reference. From the results, we can see that

[1] We have also tried other data with different percentages, e.g. 25%, and the results are similar.

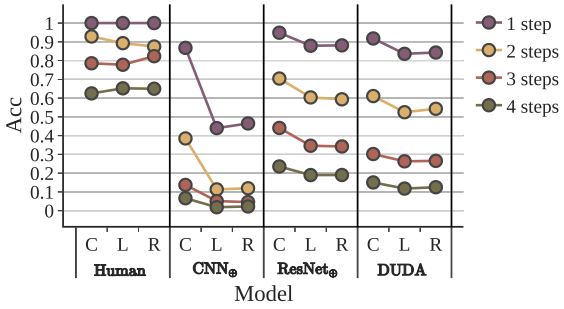


Figure 5. Results for different final views (**C**enter, **L**eft, **R**ight).

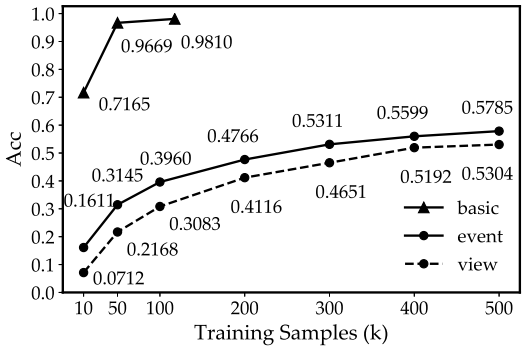


Figure 6. Results of ResNet$_{\oplus}$ with different training size.

EO of the human is zero. Therefore, once humans find all correct atomic transformations, it is easy to figure out the orders meanwhile. However, for all deep models, the EOs are larger than zero, which indicates a clear effect of the order on the reasoning process. In order to find out the extent of the effect, i.e., whether $0.0915 \sim 0.1701$ means a large deviation, we perform an experiment on 100 randomly selected order sensitive samples. Specifically for each sample, we randomly assign an order for ground truth atomic transformations. As a result, the EO is 0.5008, which could be viewed as an upper bound of the order error. Therefore, the current deep models indeed have some ability to tackle the orders, but there still leaves some room for improvements.

At last, we analyze the effect of view variation. For each model, we provide the results of different final views, as shown in Figure 5. Please note that the results of CNN$_{-}$, ResNet$_{-}$, and BCNN are quite similar to CNN$_{\oplus}$, ResNet$_{\oplus}$ and DUDA, so we just give the results from latter three typical models. Firstly, the results of humans across different views change small, demonstrating human's powerful ability of adapting to different views. In some cases, humans perform even better when views are changed than unchanged. That is because when the view is altered, humans usually spend more time solving the problems, which decreases the chances of making errors. Conversely, deep learning models share a similar trend that view variations will hurt the performance. Among these models, CNN decreases the most, while DUDA shows its robustness. In conclusion, models with more parameters are more robust to view variations and feature-based interaction like the way used in DUDA, is helpful.

### 5.4. Analysis of Training Data Size

In our experiments, we find that data size is an important factor for training and evaluation. Therefore, we use ResNet$_{\oplus}$ as an example to study the influence of this factor. From Figure 6, we can see that more training samples bring significant benefits when the number is less than 50k on Basic and 200k on Event and View. After that, the benefits become smaller and smaller. Those results are consistent with the common knowledge that relatively large data is required to well train a deep model. These results also show TRANCE has sufficient size in our experiments.

## 6. Conclusion

To tackle the problem that most existing visual reasoning tasks are solely defined on static settings and cannot well capture the dynamics between states, we propose a new visual reasoning paradigm, namely transformation driven visual reasoning (TVR). Given the initial and final states, the target is to infer the corresponding single-step transformation or multi-step transformations, represented by a triplet (object, attribute, value) or a sequence of triplets, respectively. In this paper, as an example, we use CLEVR to construct a new synthetic data, namely TRANCE, which includes three different levels of settings, i.e. Basic for single-step transformation, Event for multi-step transformation, and View for multi-step transformation with variant views. To study the effectiveness of existing SOTA reasoning techniques, we propose a new encoder-decoder framework named TranceNet and test six models under this framework. The experimental results show that our best model works well on Basic, while still has difficulties solving Event and View. Specifically, the difficult point of Event is to find all atomic transformations and arrange them with a feasible order, especially when the length of the sequence is large. While for View, the view variations bring great challenges to these models, but have little impact on humans.

In the future, we plan to investigate from both model and data perspectives by testing methods like neural symbolic approaches and constructing a real dataset to study TVR.

## Acknowledgement

This work was supported by the National Natural Science Foundation of China (NSFC) under Grants No. 61773362, and 61906180, Beijing Academy of Artificial Intelligence (BAAI) under Grants BAAI2020ZJ0303, the National Key R&D Program of China under Grants No. 2016QY02D0405, the Tencent AI Lab Rhino-Bird Focused Research Program (No. JR202033).

## References


[1] Jean-Baptiste Alayrac, Ivan Laptev, Josef Sivic, and Simon Lacoste-Julien. Joint discovery of object states and manipulation actions. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2127–2136, 2017.

[2] Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C. Lawrence Zitnick, and Devi Parikh. VQA: Visual Question Answering. In *International Conference on Computer Vision (ICCV)*, 2015.

[3] Anton Bakhtin, Laurens van der Maaten, Justin Johnson, Laura Gustafson, and Ross Girshick. Phyre: A new benchmark for physical reasoning. In *Advances in Neural Information Processing Systems*, pages 5082–5093, 2019.

[4] Fabien Baradel, Natalia Neverova, Julien Mille, Greg Mori, and Christian Wolf. Cophy: Counterfactual learning of physical dynamics. In *ICLR*, 2020.

[5] Chien-Yi Chang, De-An Huang, Danfei Xu, Ehsan Adeli, Li Fei-Fei, and Juan Carlos Niebles. Procedure planning in instructional videos. *arXiv preprint arXiv:1907.01172*, 2019.

[6] Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, 2014.

[7] Blender Online Community. Blender–a 3d modelling and rendering package, 2016.

[8] Alireza Fathi and James M Rehg. Modeling actions through state changes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2579–2586, 2013.

[9] Rohit Girdhar and Deva Ramanan. CATER: A diagnostic dataset for Compositional Actions and TEmporal Reasoning. In *ICLR*, 2020.

[10] Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. Making the V in VQA matter: Elevating the role of image understanding in Visual Question Answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6904–6913, 2017.

[11] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017.

[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[13] Ting-Hao Huang, Francis Ferraro, Nasrin Mostafazadeh, Ishan Misra, Aishwarya Agrawal, Jacob Devlin, Ross Girshick, Xiaodong He, Pushmeet Kohli, Dhruv Batra, et al. Visual storytelling. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1233–1239, 2016.

[14] Drew A Hudson and Christopher D Manning. Compositional attention networks for machine reasoning. *arXiv preprint arXiv:1803.03067*, 2018.

[15] Drew A. Hudson and Christopher D. Manning. GQA: A new dataset for real-world visual reasoning and compositional question answering. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[16] Phillip Isola, Joseph J Lim, and Edward H Adelson. Discovering states and transformations in image collections. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1383–1391, 2015.

[17] Jung-Hwan Jin, Hyun Joon Shin, and Jung-Ju Choi. Spoid: a system to produce spot-the-difference puzzle images with difficulty. *The Visual Computer*, 29(6-8):481–489, 2013.

[18] 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.

[19] Justin Johnson, Bharath Hariharan, Laurens Van Der Maaten, Judy Hoffman, Li Fei-Fei, C Lawrence Zitnick, and Ross Girshick. Inferring and executing programs for visual reasoning. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2989–2998, 2017.

[20] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[21] Yong-Lu Li, Yue Xu, Xiaohan Mao, and Cewu Lu. Symmetry and group in attribute-object compositions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11316–11325, 2020.

[22] Tsung-Yu Lin, Aruni RoyChowdhury, and Subhransu Maji. Bilinear CNN models for fine-grained visual recognition. In *Proceedings of the IEEE international conference on computer vision*, pages 1449–1457, 2015.

[23] Runtao Liu, Chenxi Liu, Yutong Bai, and Alan L Yuille. CLEVR-Ref+: Diagnosing visual reasoning with referring expressions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4185–4194, 2019.

[24] Yang Liu, Ping Wei, and Song-Chun Zhu. Jointly recognizing object fluents and tasks in egocentric videos. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2924–2932, 2017.

[25] Kenneth Marino, Mohammad Rastegari, Ali Farhadi, and Roozbeh Mottaghi. OK-VQA: A visual question answering benchmark requiring external knowledge. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3195–3204, 2019.

[26] Tushar Nagarajan and Kristen Grauman. Attributes as operators: factorizing unseen attribute-object compositions. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 169–185, 2018.

[27] Dong Huk Park, Trevor Darrell, and Anna Rohrbach. Robust change captioning. In *The IEEE International Conference on Computer Vision (ICCV)*, October 2019.

[28] Jae Sung Park, Chandra Bhagavatula, Roozbeh Mottaghi, Ali Farhadi, and Yejin Choi. Visualcomet: Reasoning about the dynamic context of a still image. In *In Proceedings of the European Conference on Computer Vision (ECCV)*, 2020.

[29] Jean Piaget. The role of action in the development of thinking. In *Knowledge and development*, pages 17–42. Springer, 1977.

[30] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016.

[31] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015.

[32] Alane Suhr, Mike Lewis, James Yeh, and Yoav Artzi. A corpus of natural language for visual reasoning. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 217–223, 2017.

[33] Alane Suhr, Stephanie Zhou, Ally Zhang, Iris Zhang, Huajun Bai, and Yoav Artzi. A corpus for reasoning about natural language grounded in photographs. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6418–6428, 2019.

[34] Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. CommonsenseQA: A Question Answering Challenge Targeting Commonsense Knowledge. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4149–4158, 2019.

[35] Peng Wang, Qi Wu, Chunhua Shen, Anthony Dick, and Anton van den Hengel. FVQA: Fact-based visual question answering. *IEEE transactions on pattern analysis and machine intelligence*, 40(10):2413–2427, 2018.

[36] Xiaolong Wang, Ali Farhadi, and Abhinav Gupta. Actions˜ transformations. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 2658–2667, 2016.

[37] Ronald J Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3-4):229–256, 1992.

[38] R. J. Williams and D. Zipser. A learning algorithm for continually running fully recurrent neural networks. *Neural Computation*, 1(2):270–280, 1989.

[39] Ning Xie, Farley Lai, Derek Doran, and Asim Kadav. Visual entailment task for visually-grounded language learning. *arXiv preprint arXiv:1811.10582*, 2018.

[40] Ning Xie, Farley Lai, Derek Doran, and Asim Kadav. Visual entailment: A novel task for fine-grained image understanding. *arXiv preprint arXiv:1901.06706*, 2019.

[41] 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 *International Conference on Learning Representations*, 2020.

[42] Rowan Zellers, Yonatan Bisk, Ali Farhadi, and Yejin Choi. From Recognition to Cognition: Visual Commonsense Reasoning. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[43] Peng Zhang, Yash Goyal, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. Yin and Yang: Balancing and answering binary visual questions. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[44] Yuke Zhu, Oliver Groth, Michael Bernstein, and Li Fei-Fei. Visual7w: Grounded question answering in images. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4995–5004, 2016.

[45] Tao Zhuo, Zhiyong Cheng, Peng Zhang, Yongkang Wong, and Mohan Kankanhalli. Explainable video action reasoning via prior knowledge and state transitions. In *Proceedings of the 27th ACM International Conference on Multimedia*, pages 521–529, 2019.