VISUAL TRANSFORMATION TELLING

Anonymous authors
Paper under double-blind review

ABSTRACT

In this paper, we propose a new visual reasoning task, called Visual Transformation Telling (VTT). Given a series of states (i.e. images), a machine is required to describe what happened (i.e. transformation) between every two adjacent states. Different from most existing visual reasoning tasks, which focus on state reasoning, VTT concentrates on transformation reasoning. Moreover, describing the transformation in the form of language is more natural and closer to the real application than the property change way in the previous TVR task. We collect 13,547 samples from two instructional video datasets, i.e. CrossTask and COIN, and extract desired states and transformation descriptions to form a suitable VTT benchmark dataset. After that, we introduce an end-to-end learning model for VTT, named TTNet. TTNet consists of three components to mimic human's cognition process of reasoning transformation. First, an image encoder, e.g. CLIP, reads content from each image, then a context encoder links the image content together, and at last, a transformation decoder autoregressively generates transformation descriptions between every two adjacent images. This basic version of TTNet is difficult to meet the cognitive challenge of VTT, that is to identify abstract transformations from images with small visual differences, and the descriptive challenge, which asks to describe the transformation consistently. In response to these difficulties, we propose three strategies to improve TTNet. Specifically, TTNet leverages difference features to emphasize small visual gaps, masked transformation model to stress context by forcing attention to neighbor transformations, and auxiliary category and topic classification tasks to make transformations consistent by sharing underlying semantics among representations. We adapt some typical methods from visual storytelling and dense video captioning tasks, considering their similarity with VTT. Our experimental results show that TTNet achieves better performance on transformation reasoning. In addition, our empirical analysis demonstrates the soundness of each module in TTNet, and provides some insight into transformation reasoning.

1 INTRODUCTION

What will come to your mind when you are given a series of images, e.g. Figure 1? Probably we first notice the content of each image, then we link these images in our mind, and finally conclude a series of events from images, i.e. the whole intermediate process of cooking noodles. In fact, this is a typical reasoning process from states (i.e. single images) to transformation (i.e. changes between images), as described in Piaget's theory of cognitive development (Bovet, 1976; Piaget, 1977). More specifically, children at the preoperational stage (2-7 years old) usually pay their attention mainly to states and ignore the transformations between states, whereas the reverse is true for children at the concrete operational stage (7-12 years old). Interestingly, computer vision is developed through a similar evolution pattern. In the last few decades, image understanding, including image classification, detection, captioning, and question answering, mainly focusing on visual states, has been comprehensively studied and achieved satisfying results.

Now it is time to pay more attention to the visual transformation reasoning tasks. Recently, there have been some preliminary studies (Park et al., 2019; Hong et al., 2021) on transformation. For example, Hong et al. (2021) defines a transformation driven visual reasoning (TVR) task, where both initial and final states are given, and the changes of object properties including color, shape, and position are required to be obtained based on a synthetic dataset. However, the current studies

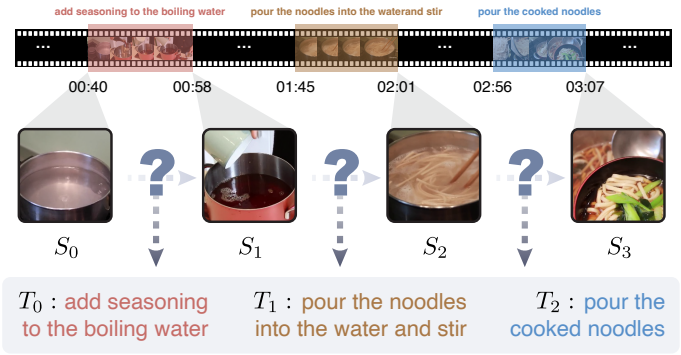


Figure 1: The illustration of Visual Transformation Telling. Given states represented by images, the goal of the task is to reason and describe transformations between every two adjacent states.

of transformation reasoning remain limited in two aspects. Firstly, the task is defined in an artificial environment that is far from reality. Secondly, the definition of transformation is limited to predefined properties, which cannot be well generalized to unseen or new environments. As a result, the existing transformation reasoning task cannot meet the requirement of real-world applications. Furthermore, the lack of strong transformation reasoning ability will hinder some more advanced event-level reasoning tasks, such as visual storytelling(Ting-Hao et al., 2016) and procedure planning(Chang et al., 2020), since transformation plays an important role in these tasks.

To tackle these limitations, we propose a new visual transformation telling (VTT) task in this paper. The main motivation is to provide descriptions for real-world transformations. For example, given two images with dry and wet ground respectively, it should be described it rained, which precisely describes a cause-and-effect transformation. Therefore, the formal definition of VTT is to output language sentences to describe the transformation for a given series of states, i.e. images. VTT is different from video description tasks, e.g. dense video captioning Krishna et al. (2017), since the complete process of transformations is shown by videos, which reduces the challenge of reasoning.

To facilitate the study of VTT, we collect 13,547 samples from two instructional video datasets, including CrossTask(Zhukov et al., 2019) and COIN (Tang et al., 2019; 2021). They are originally used for evaluating step localization, action segmentation, and other video analysis tasks. But we found them suitable to be modified to fit VTT, because the transformations are mainly about daily activities, and more importantly, some main steps to accomplish a certain job have been annotated in their data, including temporal boundaries and text descriptions. Therefore, we extract key images from a video as input, and directly use their text labels of the main steps as transformation descriptions. More details can be found in Section 3.2.

When designing an effective VTT model, we face two kinds of challenges. The first one is related to the cognitive challenge, which is to derive abstract transformation from images with small differences, e.g. from the difference between wet and dry ground to rained. The second one is the descriptive challenge, that is, the description of transformations should consider the consistency in a series of images to output a reasonable event. If we only consider the description for a single transformation, i.e. between two images, it is easy to output logical errors in the results.

In order to address these challenges, we propose a difference-sensitive and context-aware model, named TTNet (Transformation Telling Net). TTNet consists of three major components, to mimic the human cognition process of transformation reasoning. To be specific, CLIP (Radford et al., 2021) is utilized as the image encoder to read semantic information from images into image vectors. Then a transformer-based context encoder interacts image vectors together to capture context information. At last, a transformer decoder autoregressively generates descriptions according to context features. However, this basic model is not enough to meet the cognitive and descriptive challenges, so we use three well-designed strategies to improve TTNet. Specifically, the first strategy is to compute difference features on image vectors and fed them into the context encoder as well, to emphasize small visual gaps. Then, masked transformation model is applied to capture the context-aware information, by randomly masking out the inputs of the context encoder like masked

language model (Devlin et al., 2019). Finally, in addition to the general text generation loss, the whole network is also supervised under the auxiliary task of category and topic classification, which is to constrain the transformation representations to share underlying semantics, by mimicking human's behavior that forms a global event in mind.

Since the task of VTT is new, there is no ready-made baseline model. Considering the similarity of visual storytelling and dense video captioning to VTT, we modify typical methods including CST (Gonzalez-Rico & Fuentes-Pineda, 2018), GLACNet (Kim et al., 2019), and Densecap Johnson et al. (2016) in these two applications as our baseline methods. Our experimental results show that TTNet significantly outperforms these methods. Additionally, we conduct comprehensive studies to show the importance of contextual information for VTT and the effectiveness of three strategies, including difference features, masked transformation model, and auxiliary learning.

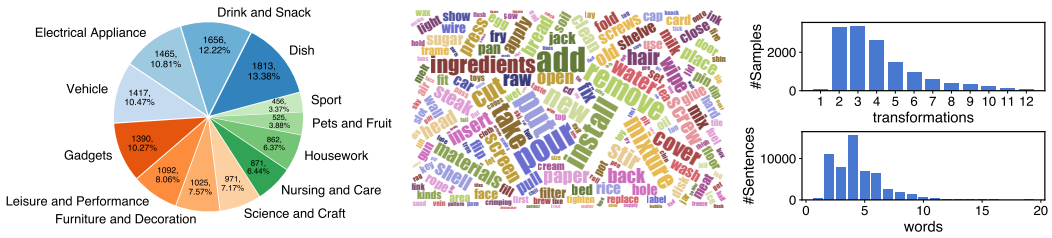
In conclusion, our major contributions include: 1) the proposal of a new task called visual transformation telling to emphasize the reasoning of transformation in real world applications; 2) the introduction of TTNet, which is a difference-sensitive and context-aware model for transformation reasoning. 3) extensive experiments on our collected data from instructional video applications, demonstrating the effectiveness of TTNet and providing many insights for understanding the VTT task.

2 Related Works

VTT belongs to the direction of visual reasoning, so we first list some typical visual reasoning tasks and then discuss the relation between VTT and these tasks. CLEVR (Johnson et al., 2017), one of the earliest visual reasoning tasks, as well as GQA (Hudson & Manning, 2019), concentrates on relation and logical reasoning on objects. RAVEN (Zhang et al., 2019) and V-PROM (Teney et al., 2020) care about the induction and reasoning of graphic patterns. VCR (Zellers et al., 2019) and Sherlock (Hessel et al., 2022) test whether machines are able to learn commonsense knowledge to answer some questions in everyday life scenarios. These tasks mainly concentrate on state-level reasoning. Apart from these tasks, there is a series of works related to dynamic reasoning. Physical reasoning (Bakhtin et al., 2019; Yi et al., 2020; Girdhar & Ramanan, 2020; Baradel et al., 2020; Riochet et al., 2022) evaluates the ability to learn physical rules from data to answer questions or solve puzzles. VisualCOMET (Park et al., 2020) asks to reason beyond the given state to answer what happened before and what will happen next. Visual storytelling (Park et al., 2020) even requires completing the missing information between given states to describe a story with complete logic. We can find that visual reasoning has a tendency to shift from static scenes to dynamic ones. While state and transformation are both important for reasoning in dynamic scenes, we would like to concentrate on transformation reasoning, between state-only scenarios and more complex composite scenarios.

To the best of our knowledge, there are rare studies on designing specific tasks for visual transformation reasoning. The only work is TVR (Hong et al., 2021). Given the initial and final states, TVR requires to predict a sequence of changes in properties, including size, shape, material, color, and position. However, the synthetic scenario is far from reality and property change is not a common fashion to describe transformations in life. A more natural way is the event-level description. For example, it is more natural to tell it rained when describing what happened between dry and wet ground outside. Visual storytelling (Ting-Hao et al., 2016) requires event-level description but transformations are mixed in the description, making it difficult to evaluate transformation only. Visual abductive reasoning (Liang et al., 2022) has a similar core idea to us, which aims to find the most likely explanation for an incomplete set of observations. The difference is they only require machines to reason one single missing transformation from multiple transformations, while our task aims to reason multiple logically related transformations from states.

Talking about transformation description, there is another topic related, i.e. visual description. Here we review some typical visual description tasks and discuss their differences. Tasks that describe a single image include image captioning (Farhadi et al., 2010; Kulkarni et al., 2011), dense image captioning (Johnson et al., 2016), and image paragraphing (Krause et al., 2017). The difference lies in the level of detail. Similarly, tasks for videos include video description (Venugopalan et al., 2015), video paragraph description (Yu et al., 2016), grounded video description (Zhou et al., 2019), and dense video captioning (Krishna et al., 2017). Different from image captioning tasks that focus only on a single state, video description tasks start to describe events. For example, dense video



(a) (b) (c)

Figure 2: Different distributions of VTT samples. (a) Category. (b) Words. (c) Transformation length (top), and sentence length (bottom).

captioning asks to predict temporal boundaries and descriptions of key events in a video. However, they provide the full process of transformation throughout videos, reducing the need for reasoning.

3 Visual Transformation Telling

3.1 Task Definition

Visual transformation telling aims to test the ability of machines to reason and describe transformations from a sequence of visual states, i.e. images. Formally, $N+1$ images $S = \{s_n\}_{n=1}^{N+1}$ are given, which are logically related and semantically different. Logically related means these images are associated with a certain event, e.g. completing a job, while semantically difference is to expect some substantial changes that are meaningful to people, i.e. transformation. The target is then to reason N transformations $T = \{t_n\}_{n=1}^{N}$ between every two adjacent images and describe them with natural languages, so that $s_1 \rightarrow t_1 \rightarrow s_2 \rightarrow \cdots \rightarrow t_n \rightarrow s_{n+1}$ is logically sound.

3.2 VTT Dataset

To construct a meaningful dataset for VTT, we require the data to cover a large scope of real world transformations. Therefore, we choose instructional videos as our basic library, because they contain many daily life activities. Specifically, we choose two typical instructional video datasets, i.e. CrossTask (Zhukov et al., 2019) and COIN (Tang et al., 2019; 2021), and construct our data. Figure 1 illustrates an instruction video from COIN for cooking noodles and how we transform their annotation into VTT dataset. We can see that the video is segmented into multiple main steps, and each step is annotated with temporal boundaries and text label. We directly use their text labels as transformation descriptions and extract states based on temporal boundaries. Specifically, for the first transformation, the first frame of the corresponding step segment becomes its start state and the last frame becomes its end state. For the remaining transformations, the end state is extracted in the same way, while the start state shares the end state of the last transformation. In this way, we collected 13,547 samples as well as 55,482 transformation descriptions from CrossTask and COIN, forming our new data for VTT. Figure 2 shows the distribution of the sample category, keyword, transformation length, and sentence length. From the category distribution and the word cloud, we can see that VTT data covers lots of daily activities, like dish, drink and snack, electrical application, vehicle, gadgets, leisure and performance, etc. Furthermore, the distribution of transformation and sentence length are all long-tailed. In most cases, there are about 2-5 transformations in each sample and 2-6 words in each transformation description.

4 Method

Problem Formulation. Our main idea for solving visual transformation telling is to find a parameterized model f_θ to estimate the conditional probability $p(T|S;\theta) = p(\{t_j\}_{j=1}^{N}|\{s_i\}_{i=1}^{N+1};\theta)$, where $s_i \in \mathbb{R}^{C\times W\times H}$ is a state represented as an image and $t_j = \{x_{j,l}\}_{l=1}^{L}$ is a sentence of length L. The

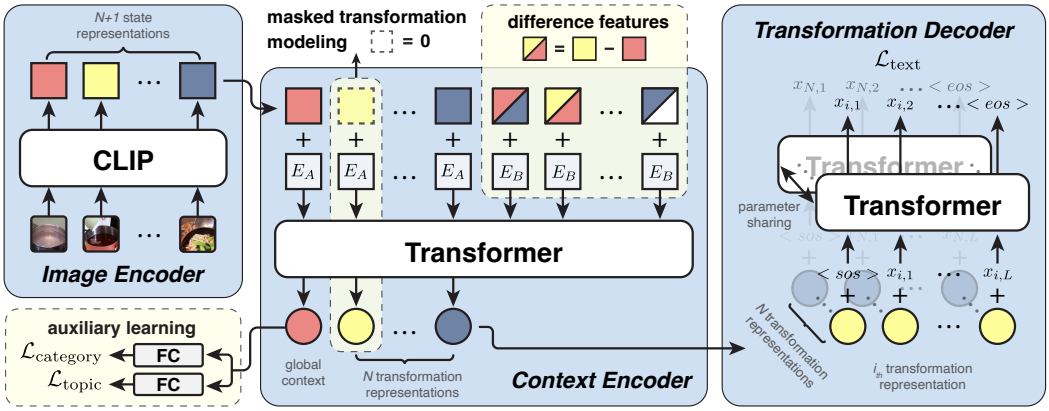


Figure 3: The architecture of TTNet.

conditional probability can also be written as auto-regressively generating N sentences:

$$p(T|S;\theta) = \prod_{j}^{N} \prod_{l}^{L} p(x_{j,l}|x_{j,<l}, \{s_i\}_{i=1}^{N+1}; \theta) \tag{1}$$

The non-autoregressive formulation also works, but will not be discussed in this paper.

Overview of TTNet. Our TTNet is designed to mimic human's cognitive process of transformation. The first step is independent recognition, which means that people may understand each image independently. Therefore, we introduce an **image encoder** f_{image} to represent each image to a vector and obtain a set of image vectors $V = \{v_i\}_{i=1}^{N+1} = \{f_{\text{image}}(s_i)\}_{i=1}^{N+1}$. After that, humans will associate these images together, and form an understanding of all images guided by a global event. To reflect this process, we introduce a **context encoder**, e.g. a bi-directional RNN or a transformer encoder, denoted as f_{context}, to obtain context-aware image representations $C = \{c_i\}_{i=1}^{N+1} = \{f_{\text{context}}(i, V)\}_{i=1}^{N+1}$ by considering contextual information. The final step is to describe these transformations based on previous understanding. In TTNet, we feed the last N context-aware image representations to a **transformation decoder** f_{caption}, implemented with an RNN or a transformer decoder, to generate transformation descriptions $T = \{t_i\}_{i=1}^{N} = \{f_{\text{caption}}(c_{i+1})\}_{i=1}^{N}$ auto-regressively.

The model is then trained with ground truth transformations $T^* = \{t_i^*\}_{i=1}^{N}$ by minimizing the following negative log-likelihood loss, where $t_i^* = \{x_{i,l}^*\}_{l=1}^{L}$ is the ground truth description of the i_{th} transformation.

$$\mathcal{L}_{\text{text}} = -\sum_{i=1}^{N} \sum_{l=1}^{L} \log p(x_{i,l}^*|x_{i,<l}^*) \tag{2}$$

In order to tackle the two unique challenges of VTT, i.e. cognitive challenge and descriptive challenge, we propose three specific strategies to enhance the above TTNet, including difference sensitive encoding, masked transformation model, and auxiliary learning. To distinguish more clearly, we called the model that does not use these three strategies $\text{TTNet}_{\text{base}}$.

4.1 Difference Sensitive Encoding

In visual transformation telling, the differences between two adjacent states are usually very small. Imagine the scene of cooking noodles, the whole picture does not change much before and after the noodles are added to the pot. This characteristic requires the model not only to understand the content of each image, but also to focus on differences between images to facilitate the understanding of transformations. For this purpose, we first utilize CLIP (Radford et al., 2021) as our image encoder, due to its strong semantic representation ability trained on large scale unsupervised data. We also introduce difference features, by subtracting the current state and the previous state representations $\Delta V = \{v_i - v_{i-1}\}_{i=1}^{N+1}$, where $v_0 = v_{N+1}$, to emphasize the subtle difference. The above two kinds

of representations are concatenated and fed to the context encoder. Furthermore, a type embedding is added to distinguish these two kinds of features.

Since transformations are not independent, we may meet the logical consistency problem in the transformation description process, named the descriptive challenge of VTT. For example, the logic of descriptions does not make sense as shown in Figure 4. $\text{TTNet}_{\text{base}}$ recognizes oranges as eggs, which is logically unreasonable with the two transformations before and after. In TTNet, we introduce masked transformation model in the context encoder and auxiliary learning in the loss function to alleviate this problem.

4.2 Masked Transformation Model

Masked transformation model (MTM) is inspired by masked language model (Devlin et al., 2019). The intuition behind is that one transformation can be reasoned from nearby transformations. For example, if you are told the previous transformation is washing the watermelon and the next is putting watermelon into a planet, it is obvious that the intermediate transformation is also related to watermelon instead of other fruit. Following this intuition, 15% of the input features in the context encoder are randomly masked during training. Furthermore, we empirically found that, for each sample, using this strategy with half the probability works better.

1. Cut both ends and remove fruit seeds.
2. Pour the egg into the bowl.
3. Pour the orange juice into the cup.

Figure 4: $\text{TTNet}_{\text{base}}$ failes to effectively use contextual information and mistakenly identifies the orange as an egg.

4.3 Auxiliary Learning

Human usually try to guess the category or topic before describing transformations, e.g. cooking noodles, cutting watermelons, and repairing a bicycle. Therefore, these category or topic information may help guide description generation. Inspired by this, we propose to leverage auxiliary tasks, i.e. category and topic classification, to supervise the training process. Specifically, we introduce two additional losses $\mathcal{L}_{\text{category}}$ and $\mathcal{L}_{\text{topic}}$ on the global context vector. We expect to make the learned transformation representations share underlying category and topic information to enhance the learning of consistent representations. So the final training loss becomes a combination of $\mathcal{L}_{\text{text}}$, $\mathcal{L}_{\text{category}}$, and $\mathcal{L}_{\text{topic}}$:

$$\mathcal{L} = \mathcal{L}_{\text{text}} + \alpha\mathcal{L}_{\text{category}} + \beta\mathcal{L}_{\text{topic}} \tag{3}$$

where α and β are two adjustment factors.

5 Experiments

In this section, we first introduce our empirical setups including baseline methods and evaluation metrics. Then we demonstrate the main empirical results on the collected VTT dataset, including both quantitative and qualitative results. After that, we show extensive ablation studies on different strategies used in TTNet.

5.1 Empirical Setups

Baseline Models. Visual storytelling and dense video captioning are the two most similar tasks to VTT. Visual storytelling requests to generate N descriptions from N images. We select two classic methods from the winners of visual storytelling challenge Mitchell et al. (2018), including CST Gonzalez-Rico & Fuentes-Pineda (2018), and GLACNet Kim et al. (2019) for comparison. CST contextualizes image features by LSTM and then generates descriptions with separate LSTMs for each image. GLACNet mixtures global LSTM features and local image features into context features and then generates descriptions with a shared LSTM decoder. When generating transformation descriptions, only the last N context features are used. Dense video captioning has a similar target to describe a series of events. The difference is the input is a video and it additionally requires to predict temporal boundaries for events. We choose DenseCap Johnson et al. (2016) for adaptation which proposed in the paper that introduces dense video captioning. DenseCap integrates the

Table 1: Performance on the test set of VTT dataset. B@4/M/R/C/BS are short for BLEU@4 / METEOR / ROUGE-L / CIDEr / BERT-Score. The architecture shows image encoder / context encoder / transformation decoder. * indicates to use CLIP as the image encoder for a fair comparison.

Model	Architecture	Params	B@4	M	R	C	BS
CST	InceptionV3 / LSTM / LSTM	379M	10.12	11.43	26.08	43.25	16.27
CST*	CLIP / LSTM / LSTM	661M	14.00	19.31	38.28	84.90	25.98
GLACNet	ResNet152 / LSTM / LSTM	128M	42.76	45.25	52.96	381.31	60.11
GLACNet*	CLIP / LSTM / LSTM	373M	55.24	59.48	66.25	508.19	71.13
DenseCap*	CLIP / Attention / LSTM	361M	48.24	51.98	59.77	439.53	66.28
$\text{TTNet}_{\text{Base}}$	CLIP / Transformer / Transformer	368M	55.70	60.49	67.05	515.28	72.22
TTNet	CLIP / Transformer / Transformer	368M	**61.22**	**66.31**	**71.84**	**570.63**	**76.25**

past and future information into image features to capture the context information. There are many advanced methods for dense video captioning but highly rely on fine-grained video features, which are not suitable for our task. All three methods are implemented as closely as possible according to the original paper and provide a fair comparison by using the same image encoder with TTNet. We describe the implementation details of TTNet as well as baseline models in supplementary materials.

Evaluation Metrics. Following previous works on visual descriptions (Ting-Hao et al., 2016; Krishna et al., 2017; Liang et al., 2022), automated metrics including BLEU@4 (Papineni et al., 2002), CIDEr (Vedantam et al., 2015), METEOR (Banerjee & Lavie, 2005), ROUGE-L (Lin & Hovy, 2002), and BERT-Score (Zhang et al., 2020) are selected for evaluation. Since descriptions in VTT are usually short, we follow the smooth strategy introduced by Chen & Cherry (2014) when computing BLEU@4, to provide more accurate results. In addition, BERT-Score is rescaled with the pre-computed baseline (Zhang et al., 2020) to have a more meaningful score range.

5.2 Results on VTT dataset

Quantitative Results. Table 1 summarizes the results of 7 models on the VTT dataset, including TTNet, $\text{TTNet}_{\text{base}}$, CST and its CLIP version, GLACNet and its CLIP version, and CLIP version DenseCap. From the results, TTNet surpasses other models on all metrics with a large margin, e.g. CIDEr is 11% higher than $\text{TTNet}_{\text{base}}$ which is the second best model. This large improvement comes from the three strategies we proposed, including difference sensitive encoding, masked transformation model, and auxiliary learning, which are the only differences between TTNet and $\text{TTNet}_{\text{base}}$. In Section 5.3, we further show their effectiveness with detailed ablation studies. It is not difficult to find that the performance gap between CST*, GLACNet*, and Densecap* is also very large. While they all use CLIP, the difference lies in the way of context decoding and text generation. GLACNet* outperforms DenseCap* mainly because LSTM captures more information than past and future attention features. The gap between GLACNet* and CST* is caused by the way of text generation. GLACNet uses word embeddings and context features as inputs in each LSTM step, while CST only uses the context as the initial state of LSTM. In our empirical studies, this little difference matters, and it is the reason that TTNet chooses to add context embedding to word embedding as the inputs of the transformation decoder rather than using the context feature as the start token. The underlying design philosophy between $\text{TTNet}_{\text{base}}$ and GLACNet* is similar, therefore, the performance is close. However, $\text{TTNet}_{\text{base}}$ converges faster than GLACNet* during training because the transformer captures the context information more efficiently than LSTM. Finally, we can see that the original version of CST and GLACNet, with Inception V3 and ResNet as image encoders accordingly perform worse than CST* and GLACNet*, indicating the choice of image encoder matters. We conduct a more detailed analysis of the image encoder in Section C of the supplementary material.

Qualitative Results. We show two examples from the VTT test data in Figure 5 about sowing and pasting a car sticker. From these two examples, we can first realize that the gap between the states is really small. For example, in the sticker case, only a small area of the sticker is changed, making it difficult to reason a certain transformation without considering the overall pasting process. We can see that when the states are confusing, e.g. DenseCap and GLACNet identify the wrong entity in the sow case, TTNet is able to reason the correct transformations from the differences and the context. Furthermore, when the difference between states becomes rather small and the transforma-

DenseCap:
1. Put the candle wick into a vessel.
2. Clean up the interior of the pumpkin.

GLACNet:
1. Dig some holes on the soil.
2. Dig some holes on the soil.

TTNet:
1. Sow on the soil.
2. Apply a cover on the soil.

Groundtruth: 1. Sow on the soil. 2. Apply a cover on the soil.

DenseCap:
1. Tear off the front of the sticker.
2. Tear off the front of the sticker.
3. Tear off the front of the sticker.
4. Tear off the front of the sticker.
5. Tear off the front of the sticker.
6. Tear off the front of the sticker.
7. Tear off the other side of the sticker.

GLACNet:
1. Clean the window surface.
2. Tear off the front of the sticker.
3. Tear off the front of the sticker.
4. Tear off the front of the sticker.
5. Tear off the front of the sticker.
6. Press the sticker.
7. Tear off the other side of the sticker.

TTNet:
1. Clean the window surface.
2. Put the sticker on the window.
3. Tear off the front of the sticker.
4. Press the sticker.
5. Tear off the front of the sticker.
6. Press the sticker.
7. Tear off the other side of the sticker.

Groundtruth:
1. Clean the window surface.
2. Put the sticker on the window.
3. Tear off the front of the sticker.
4. Press the sticker.
5. Tear off the front of the sticker.
6. Press the sticker.
7. Tear off the other side of the sticker.

Figure 5: Qualitative comparison on the VTT test data. Above: sow. Below: paste car sticker.

Table 2: CIDEr of independent transformation prediction.

Model	Original	Indep.
CST*	84.90	49.80
DenseCap*	439.53	295.75
GLACNet*	508.19	268.49
TTNet w/o diff	527.62	**422.04**
TTNet	**570.63**	349.96
TTNet (retrain)	-	459.84

Table 3: Ablation studies on the effect of key components.

Diff.	MTM	Aux.	C	BS
			515.28	72.22
√			556.85	75.00
	√		520.04	72.72
		√	521.93	72.97
√	√		562.25	75.62
√		√	562.83	75.72
	√	√	527.62	73.54
√	√	√	**570.63**	**76.25**

Table 4: Analysis of difference features and auxiliary tasks.

State	Difference	CIDEr
√	-	527.62
√	early	559.78
√	late	**570.63**

Category	Topic	CIDEr
√		549.44
	√	562.96
√	√	**570.63**

tion length becomes large, TTNet is still able to judge subtle differences between transformations. In contrast, GLACNet indeed understands the topic of pasting the car sticker but fails to distinguish some transformations. In conclusion, TTNet is able to reason transformations from confusing states and distinguish subtle differences between transformations, making it excel other methods.

5.3 Ablation Studies

In Section 4, we introduce three strategies to improve TTNet, including difference sensitive encoding, masked transformation model, and auxiliary learning. In this section, we discuss the effectiveness of these three strategies. But before that, we first need to answer the question that whether the context information is crucial for VTT, since all three strategies act on the context encoder to enhance the ability to capture context information. If the answer is yes, then it comes to answer how these strategies work and whether there exist alternative choices, e.g. other types of difference features. Experimental analyses are organized into five following topics according to this logic.

Importance Analysis of Context. To answer the question of whether the context is really important for reasoning transformations. We design to let models predict each transformation independently, i.e. only from two states before and after. If transformations can be reasoned without considering the context, model performance should remain roughly the same. However, from Table 2, the CIDEr score of all five models drops sharply from the original setting to the independent setting, showing that the context is clearly very important. Without context, reasoning transformations become rather difficult, and retraining the model with independent data does not help either.

Effectiveness of Three Strategies. Next, we move on to analyze the effectiveness of the three strategies and their combinations. The first row in Table 3 shows the result of $\text{TTNet}_{\text{Base}}$ and the next three rows show the results of using each strategy independently on the base model. Among them,

the improvement of using difference feature is the most significant, indicating the difference is also crucial for resolving transformation reasoning. The next four rows show the results of combining these strategies and the conclusion is combining all three strategies leads to the best result. The next three topics will go through all these strategies one by one in detail, to see how these strategies work.

Analysis of Difference Sensitive Encoding. We just show difference feature is the most significant strategy for TTNet. However, it is not clear how difference features help the TTNet model and if there are alternative choices for difference features, e.g. differences of raw images. To answer the first question, we need to go back to Table 2, which contains an interesting result that TTNet without using difference features overtakes the full model in the independent setting. This phenomenon suggests that difference features help to capture contextual information. Contextual information is more important for the original setting, and the model tries to capture it by attention more to the difference features. However, this does not prevail in the independent setting since contextual information is less effective and the model should attention more to the image features. This is why retraining the full model with the independent data works, because the focus of attention is adjusted during retraining. The second question is about the alternative type of difference features. We compare early and late differences. The early difference is pixel-level difference computed on raw images before input to the image encoder, while the late difference is used by TTNet and computed on encoded image vectors to become the semantic difference. In TVR (Hong et al., 2021), early difference is more effective. However, Table 4 shows the opposite result that late difference features perform the best. This is because TVR requires to predict property changes on synthetic data, which relies more on pixel differences. In contrast, VTT requires event-level descriptions, with more emphasis on semantic distinctions.

Analysis of MTM. We expect MTM to guide the model to reason transformation from nearby transformations. In order to validate this ability, we design to let models predict transformations with incomplete states, e.g mask one state of three. Specifically, we test models under two special settings. In the first setting, we randomly mask one state for all test samples. In the second setting, we give even fewer states on average by only providing start and end states for each sample. The results are shown in Figure 6. We can see that when there is less and less information, the performance of all models decreases. However, TTNet has the slowest decline in performance, showing its robustness to missing states. By further comparing the results between TTNet and TTNet w/o MTM, we can conclude this robustness is contributed by the MTM strategy.

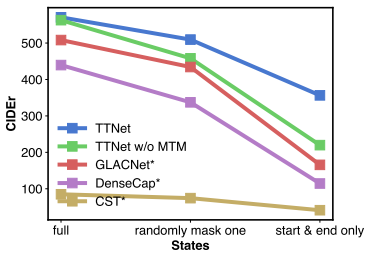


Figure 6: Effect of missing states.

Analysis of Auxiliary Tasks. Finally, we analyze the effects of different auxiliary tasks and report the results in Table 4. From the table, topic classification is more effective than category classification, since topics are more granular than categories. Supervision with two classification tasks simultaneously improves the overall performance, e.g. 562.25 $\rightarrow$ 570.63 in terms of CIDEr.

6 Conclusion

This paper introduces a new visual reasoning task to focus on transformation reasoning, i.e. changes between every two states, named visual transformation telling. Given a series of images as states, the description of each transformation is required to represent what happened between every two adjacent images. In this way, the task could be used to test the machines' ability of transformation reasoning, which is an important cognitive skill for humans, as described in Piaget's theory. To the best of our knowledge, this is the first real world application for transformation reasoning by defining transformation descriptions as output. To facilitate the study on VTT, we build benchmark data based on 13,547 samples from two instructional video datasets, i.e. CrossTask and COIN. After that, we design a model named TTNet, by applying three well-designed strategies into a basic human-inspired transformation telling model to make it difference-sensitive and context-aware. From the experiments, we find that the proposed strategies help VTT generate consistent transformation descriptions, and thus obtain better results in terms of natural language generation metrics. The empirical studies provide valuable insights for understanding VTT and the proposed model and may help to design more complicated transformation reasoning tasks or models in the future.

REFERENCES

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*, volume 32, 2019.

Satanjeev Banerjee and Alon Lavie. METEOR: An Automatic Metric for MT Evaluation with Improved Correlation with Human Judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pp. 65–72, 2005.

Hangbo Bao, Li Dong, Songhao Piao, and Furu Wei. BEiT: BERT Pre-Training of Image Transformers. In *International Conference on Learning Representations*, 2022.

Fabien Baradel, Natalia Neverova, Julien Mille, Greg Mori, and Christian Wolf. CoPhy: Counterfactual Learning of Physical Dynamics. In *International Conference on Learning Representations*, 2020.

Magali Bovet. Piaget's Theory of Cognitive Development and Individual Differences. In Bärbel Inhelder, Harold H. Chipman, and Charles Zwingmann (eds.), *Piaget and His School: A Reader in Developmental Psychology*, Springer Study Edition, pp. 269–279. 1976.

Chien-Yi Chang, De-An Huang, Danfei Xu, Ehsan Adeli, Li Fei-Fei, and Juan Carlos Niebles. Procedure Planning in Instructional Videos. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm (eds.), *Computer Vision – ECCV 2020*, volume 12356, pp. 334–350. 2020.

Boxing Chen and Colin Cherry. A Systematic Comparison of Smoothing Techniques for Sentence-Level BLEU. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, pp. 362–367, 2014.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. 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)*, pp. 4171–4186, 2019.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In *International Conference on Learning Representations*, 2022.

Ali Farhadi, Mohsen Hejrati, Mohammad Amin Sadeghi, Peter Young, Cyrus Rashtchian, Julia Hockenmaier, and David Forsyth. Every Picture Tells a Story: Generating Sentences from Images. In Kostas Daniilidis, Petros Maragos, and Nikos Paragios (eds.), *Computer Vision – ECCV 2010*, Lecture Notes in Computer Science, pp. 15–29, 2010.

Rohit Girdhar and Deva Ramanan. CATER: A diagnostic dataset for Compositional Actions & TEmporal Reasoning. In *International Conference on Learning Representations*, 2020.

Diana Gonzalez-Rico and Gibran Fuentes-Pineda. Contextualize, Show and Tell: A Neural Visual Storyteller, 2018.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016.

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, 2022.

Xin Hong, Yanyan Lan, Liang Pang, Jiafeng Guo, and Xueqi Cheng. Transformation Driven Visual Reasoning. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6899–6908, 2021.

Drew A. Hudson and Christopher D. Manning. GQA: A New Dataset for Real-World Visual Reasoning and Compositional Question Answering. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6693–6702, 2019.

Justin Johnson, Andrej Karpathy, and Li Fei-Fei. DenseCap: Fully Convolutional Localization Networks for Dense Captioning. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4565–4574, 2016.

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 *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1988–1997, 2017.

Taehyeong Kim, Min-Oh Heo, Seonil Son, Kyoung-Wha Park, and Byoung-Tak Zhang. GLAC Net: GLocal Attention Cascading Networks for Multi-image Cued Story Generation, 2019.

Jonathan Krause, Justin Johnson, Ranjay Krishna, and Li Fei-Fei. A Hierarchical Approach for Generating Descriptive Image Paragraphs, 2017.

Ranjay Krishna, Kenji Hata, Frederic Ren, Li Fei-Fei, and Juan Carlos Niebles. Dense-Captioning Events in Videos. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 706–715, 2017.

Girish Kulkarni, Visruth Premraj, Sagnik Dhar, Siming Li, Yejin Choi, Alexander C Berg, and Tamara L Berg. Baby talk: Understanding and generating simple image descriptions. In *CVPR 2011*, pp. 1601–1608, 2011.

Chen Liang, Wenguan Wang, Tianfei Zhou, and Yi Yang. Visual Abductive Reasoning, 2022.

Chin-Yew Lin and Eduard Hovy. Manual and automatic evaluation of summaries. In *Proceedings of the ACL-02 Workshop on Automatic Summarization - Volume 4*, AS '02, pp. 45–51, 2002.

Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin Transformer: Hierarchical Vision Transformer using Shifted Windows. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 9992–10002, 2021.

Margaret Mitchell, Ting-Hao 'Kenneth' Huang, Francis Ferraro, and Ishan Misra (eds.). *Proceedings of the First Workshop on Storytelling*. 2018.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. BLEU: A Method for Automatic Evaluation of Machine Translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pp. 311–318, 2002.

Dong Huk Park, Trevor Darrell, and Anna Rohrbach. Robust Change Captioning. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 4623–4632, 2019.

Jae Sung Park, Chandra Bhagavatula, Roozbeh Mottaghi, Ali Farhadi, and Yejin Choi. VisualCOMET: Reasoning about the Dynamic Context of a Still Image, 2020.

Jean Piaget. The Role of Action in the Development of Thinking. In Willis F. Overton and Jeanette McCarthy Gallagher (eds.), *Knowledge and Development: Volume 1 Advances in Research and Theory*, pp. 17–42. 1977.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning Transferable Visual Models From Natural Language Supervision, 2021.

Ronan Riochet, Mario Ynocente Castro, Mathieu Bernard, Adam Lerer, Rob Fergus, Véronique Izard, and Emmanuel Dupoux. IntPhys 2019: A Benchmark for Visual Intuitive Physics Understanding. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(9):5016–5025, 2022. ISSN 1939-3539.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the Inception Architecture for Computer Vision. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2818–2826, 2016.

Yansong Tang, Dajun Ding, Yongming Rao, Yu Zheng, Danyang Zhang, Lili Zhao, Jiwen Lu, and Jie Zhou. COIN: A Large-scale Dataset for Comprehensive Instructional Video Analysis. *arXiv:1903.02874 [cs]*, 2019.

Yansong Tang, Jiwen Lu, and Jie Zhou. Comprehensive Instructional Video Analysis: The COIN Dataset and Performance Evaluation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(9):3138–3153, 2021. ISSN 0162-8828, 2160-9292, 1939-3539.

Damien Teney, Peng Wang, Jiewei Cao, Lingqiao Liu, Chunhua Shen, and Anton van den Hengel. V-PROM: A Benchmark for Visual Reasoning Using Visual Progressive Matrices. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(07):12071–12078, 2020. ISSN 2374-3468.

Ting-Hao, Huang, Francis Ferraro, Nasrin Mostafazadeh, Ishan Misra, Aishwarya Agrawal, Jacob Devlin, Ross Girshick, Xiaodong He, Pushmeet Kohli, Dhruv Batra, C. Lawrence Zitnick, Devi Parikh, Lucy Vanderwende, Michel Galley, and Margaret Mitchell. Visual Storytelling, 2016.

Ramakrishna Vedantam, C. Lawrence Zitnick, and Devi Parikh. CIDEr: Consensus-based image description evaluation. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4566–4575, 2015.

Subhashini Venugopalan, Huijuan Xu, Jeff Donahue, Marcus Rohrbach, Raymond Mooney, and Kate Saenko. Translating Videos to Natural Language Using Deep Recurrent Neural Networks. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 1494–1504, 2015.

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.

Haonan Yu, Jiang Wang, Zhiheng Huang, Yi Yang, and Wei Xu. Video Paragraph Captioning Using Hierarchical Recurrent Neural Networks. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4584–4593, 2016.

Rowan Zellers, Yonatan Bisk, Ali Farhadi, and Yejin Choi. From Recognition to Cognition: Visual Commonsense Reasoning. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6713–6724, 2019.

Chi Zhang, Feng Gao, Baoxiong Jia, Yixin Zhu, and Song-Chun Zhu. RAVEN: A Dataset for Relational and Analogical Visual REasoNing. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5312–5322, 2019.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. BERTScore: Evaluating Text Generation with BERT. In *ICLR*, 2020.

Luowei Zhou, Yannis Kalantidis, Xinlei Chen, Jason J. Corso, and Marcus Rohrbach. Grounded Video Description. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6571–6580, 2019.

Dimitri Zhukov, Jean-Baptiste Alayrac, Ramazan Gokberk Cinbis, David Fouhey, Ivan Laptev, and Josef Sivic. Cross-Task Weakly Supervised Learning From Instructional Videos. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3532–3540, 2019.

A Additional Dataset Analysis

Table 5 show the samples and transformations in the VTT dataset. We split the data randomly into Train / Val / Test sets with samples of 10759 / 1352 / 1436 in the level of topic. The detailed topic distribution is shown in Figure 7. About half of the topics have over 100 samples.

Table 5: Statistics of the VTT dataset.

	Samples	Transformations
Train	10759	43957
Val	1352	5622
Test	1436	5903
Total	13547	55482

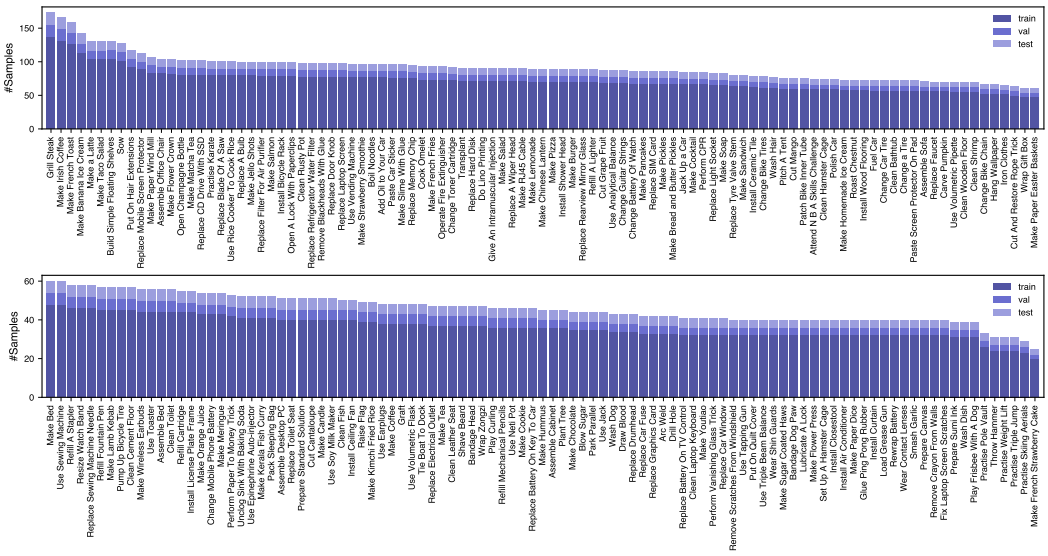


Figure 7: The sample distribution of all the topics in VTT.

B Implementation Details

We use PyTorch to implement all models. For the image encoder of TTNet, we use ViT-L/14 from CLIP, which is a modified version of the vision transformer. During training, images are randomly cropped 224 $\times$ 224 and random flipping is also used. The context encoder consists of two transformer encoder layers. The transformation decoder consists of two transformer decoder layers. We use relative position embeddings for all transformer layers. The adjustment factor α for $\mathcal{L}_{\text{category}}$ is 0.025 and β for $\mathcal{L}_{\text{topic}}$ is 0.1. Models are trained on one Tesla A100 80G GPU card with 50 epochs. The optimizer we used is AdamW, with the learning rate first warming up to 1e-4 in the first 2000 steps and then gradually decreasing to 0. For text sampling, we use top-k top-p filtering with top-k=100 and top-p=0.9.

C Comparison between Image Encoders

[1]Model weights and top-1 accuracy on ImageNet of ImageNet pretrained models are from: `https://github.com/rwightman/pytorch-image-models`

[2]Pretrained weights of CLIP models are from `https://github.com/openai/CLIP` and top-1 accuracy on ImageNet is from Table 10 of the original paper.

Table 6: Results of different image encoders.

	Image Encoder	Params	Acc	B@4	C	BS
ImageNet Pretrained[1]	InceptionV3 (Szegedy et al., 2016)	23M	77.44	44.88	404.85	61.75
	ResNet152 (He et al., 2016)	59M	82.82	50.71	464.01	67.40
	ViT-L (Dosovitskiy et al., 2022)	304M	85.84	58.26	540.46	73.59
	Swin-L (Liu et al., 2021)	196M	86.32	57.36	531.51	73.03
	BEiT-L (Bao et al., 2022)	306M	87.48	41.57	370.00	58.80
Image-text Pretrained[2]	RN50	39M	73.30	53.35	491.80	69.79
	RN101	57M	75.70	53.78	495.30	70.08
	ViT-B/32	88M	76.10	55.21	510.08	71.27
	ViT-B/16	86M	80.20	57.73	534.92	73.37
	ViT-L/14	304M	83.90	**61.22**	**570.63**	**76.25**

Image encoding quality is the basis for subsequent reasoning and description of the model, and thus greatly affects the overall performance of the model. We test 10 state-of-the-art image encoders, 5 were pretrained on ImageNet and 5 are CLIP models pretrained on large scale image-text data. In the table, we show their parameter size, ImageNet top-1 accuracy, and the performance on the VTT dataset. We can see that when the parameter sizes are similar, models pre-trained on image and text data perform better than that pre-trained only on image data, e.g. ViT-L/14 vs. ViT-L. This is consistent with the existing understanding that CLIP encodes more semantic information. In addition to training data, factors that affect model performance include model size, patch size used in vision transformers, and training strategies. For example, CLIP models have more parameters performs better. While the parameter size between ViT-B/16 and ViT-B/32 are similar, ViT-B/16 encodes finer image has smaller patch size resulting in a better image representation. BEiT-L has the highest accuracy on ImageNet but performs the worst among all models. Our explanation is that BEiT has learned enough image pattern information, but there is a defect in the capture of semantic information.

D More Qualitative Results.

We show more results in Figure 8 and Figure 9. Figure 9 shows several hard cases that TTNet fails to reason and describe. We point out three potential directions to improve the TTNet. The first one relates the image recognition ability. From the first case, TTNet recognizes the tent as platfond which is wrong. This error might respond to the image encoder that fails to distinguish these objects. Therefore, it may lead to a better result by using a more powerful image encoder. The major point here is that the model needs to identify subtle differences between states and determine specific transformations based on context. The last case is out-of-domain cases, that is, the test samples are quite different from the training data.

DenseCap:
1. Draw black patterns.
2. Cut the raw materials.
3. Cut the packing paper.
4. Cut the raw materials.
5. Nail down or stick together.

GLACNet:
1. Fold or bent the paper.
2. Fold or bent the paper.
3. Fold or bent the paper.
4. Nail down or stick together.
5. Nail down or stick together.

TTNet:
1. Draw pictures on the materials.
2. Fold or bent the paper.
3. Cut the raw materials.
4. Nail down or stick together.
5. Nail down or stick together.

Groundtruth:
1. Draw pictures on the materials.
2. Fold or bent the paper.
3. Cut the raw materials.
4. Nail down or stick together.
5. Nail down or stick together.

DenseCap:
1. Remove the air nozzle.
2. Remove the air nozzle.
3. Pump up to the tire.
4. Pump up to the tire.

GLACNet:
1. Screw off the valve cap and open the valve.
2. Pump up to the tire.
3. Pump up to the tire.
4. Tighten the valve and screw on the valve cap.

TTNet:
1. Screw off the valve cap and open the valve.
2. Install the air nozzle.
3. Pump up to the tire.
4. Remove the air nozzle.

Groundtruth:
1. Screw off the valve cap and open the valve.
2. Install the air nozzle.
3. Pump up to the tire.
4. Remove the air nozzle.

DenseCap:
1. Do the first two jumps.
2. Do the first two jumps.
3. Do the third jump.

GLACNet:
1. Begin to run up.
2. Do the third jump.
3. Do the third jump.

TTNet:
1. Begin to run up.
2. Do the first two jumps.
3. Do the third jump.

Groundtruth:
1. Begin to run up.
2. Do the first two jumps.
3. Do the third jump.

DenseCap:
1. Dig a pit with proper size.
2. Fill the tree into the pit.
3. Fill the tree into the pit.
4. Fill the tree into the pit.
5. Fill the tree into the pit.
6. Fill the pit with some soil.

GLACNet:
1. Dig a pit with proper size.
2. Fill the tree into the pit.
3. Fill the tree into the pit.
4. Fill the pit with some soil.
5. Fill the pit with some soil.
6. Pour water to the tree.

TTNet:
1. Dig a pit with proper size.
2. Fill the tree into the pit.
3. Fill the pit with some soil.
4. Pour water to the tree.
5. Fill the pit with some soil.
6. Pour water to the tree.

Groundtruth:
1. Dig a pit with proper size.
2. Fill the tree into the pit.
3. Fill the pit with some soil.
4. Pour water to the tree.
5. Fill the tree into the pit.
6. Pour water to the tree.

Figure 8: More qualitative results in the VTT test data.

DenseCap:
1. Set up the brackets.
2. Set up the platfond.
3. Fix the ground nail.

GLACNet:
1. Lay the cushion evenly.
2. Set up the platfond.
3. Set up the platfond.

TTNet:
1. Lay the cushion evenly.
2. Set up the platfond.
3. Fix the ground nail.

Groundtruth:
1. Lay the cushion evenly.
2. Set up the brackets.
3. Fix the ground nail.

DenseCap:
1. Rise to the sky.
2. Rise to the sky.
3. Rise to the sky.
4. Rise to the sky.
5. Rise to the sky.
6. Rise to the sky.
7. Rise to the sky.
8. Rise to the sky.
9. Rise to the sky.

GLACNet:
1. Rise to the sky.
2. Rise to the sky.
3. Rise to the sky.
4. Rise to the sky.
5. Rise to the sky.
6. Rise to the sky.
7. Rise to the sky.
8. Rise to the sky.
9. Rise to the sky.

TTNet:
1. Ski down from the hill.
2. Ski down from the hill.
3. Rise to the sky.
4. Ski down from the hill.
5. Ski down from the hill.
6. Rise to the sky.
7. Ski down from the hill.
8. Ski down from the hill.
9. Rise to the sky.

Groundtruth:
1. Ski down from the hill.
2. Ski up from the hill.
3. Rise to the sky.
4. Ski down from the hill.
5. Ski up from the hill.
6. Rise to the sky.
7. Fall to the ground.
8. Ski up from the hill.
9. Rise to the sky.

DenseCap:
1. Wipe the glue to a layer.
2. Wipe the glue to a layer.
3. Wipe nose.

GLACNet:
1. Wipe up the face.
2. Wipe the glue to a layer.
3. Wipe up the face.

TTNet:
1. Disinfect the injecting place.
2. Disinfect the injecting place.
3. Pull out the needle and press.

Groundtruth:
1. Add some water to the vessel.
2. Grind roundly and evenly.
3. Grind roundly and evenly.

Figure 9: Bad cases on the VTT dataset.