Inferring Past Human Actions in Homes with Abductive Reasoning - Supplementary

Clement Tan[1] Chai Kiat Yeo[1] Cheston Tan[2] Basura Fernando[1,2]
[1]Nanyang Technological University, Singapore
[2]Centre for Frontier AI Research (CFAR), Agency for Science, Technology and Research (A*STAR)
{s190099, asckyeo}@ntu.edu.sg, {cheston−tan@i2r, fernando_basura@ihpc}.a−star.edu.sg

1. Other Baseline Methods

In this section, we give the details of all other baseline methods used in the paper.

1.1. Rule-based Abductive Past Action Inference

In abductive past action inference, we assume the following logical association holds,

$$\{a_1, a_2, a_3, \cdots, a_K\} \rightarrow \{R_1, R_2, \cdots, R_N\} \tag{1}$$

where $\{R_1, R_2, \cdots, R_N\}$ is the relation set $\mathcal{R}$ present in an image and $\{a_1, a_2, a_3, \cdots, a_K\}$ is the action set $\mathcal{A}$ executed by the human to arrive at the image. Note the set of all actions is denoted by A where $\mathcal{A} \subset A$.

In rule-based inference, each relation is in the symbolic form $R_k =< H, o_k >$ where H and o_k are the human feature and k^{th} object label in the image. As the human feature is common in all relations, we omit the human feature in each relation. Then, the relational association is updated as follows:

$$\{a_1, a_2, a_3, \cdots, a_K\} \rightarrow \{o_1, o_2, \cdots, o_N\} \tag{2}$$

for any image. In rule-based abductive past action set inference, for each given object pattern $\{o_1, o_2, \cdots, o_N\}$, we count the occurrence of each action a_j. Let us denote the frequency of action a_j for object pattern $\mathcal{O}_q = \{o_1, o_2, \cdots, o_N\}$ from the entire training set by C_j^q. Therefore, for each object pattern $\mathcal{O}_q$, we obtain a frequency vector over all past actions denoted by:

$$\mathbf{C^q} = [C_1^q, C_2^q, \cdots, C_{|A|}^q] \tag{3}$$

Then, we can convert these frequencies into probabilities using softmax:

$$P(A|\mathcal{O}_q) = softmax([C_1^q, C_2^q, \cdots, C_{|A|}^q]) \tag{4}$$

We use this to perform abductive past action set inference using the test set. Given a test image, we first obtain the object pattern $\mathcal{O}$. Next, we obtain the action probability vector for the object pattern from the training set using Equation 4. If an object pattern does not exist in the training set, we assign equal probability to each action.

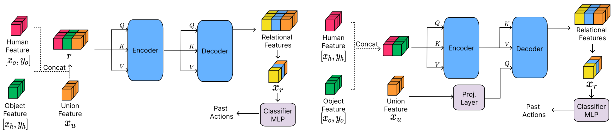


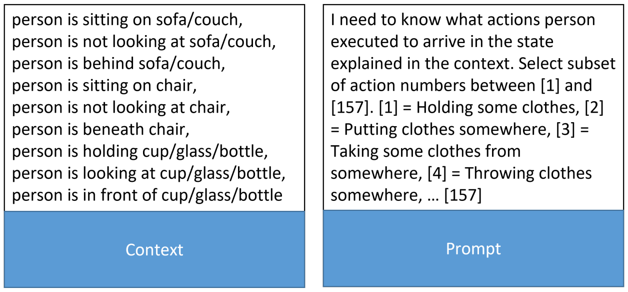
Figure 1. (left) The relational multi-head self-attention transformer. (right) The relational cross-attention transformer.

1.2. Relational MLP

The MLP consists of 2-layers. The human feature x_h, object feature x_o, and union region of both human and feature x_u obtained from the ResNet-101 FasterRCNN backbone are concatenated to form the joint relational visual features x_v. The semantic representation y_s is formed via a concatenation of the Glove [6] embedding of the human y_h and object y_o. We perform max pooling on the relational features, $\mathcal{R}_i = r_1, r_2, ..., r_n$ in a given image, where each $r_n = [x_v, y_s]$ is the concatenation of visual and semantic features. Afterward, we pass these features into the 2-layer MLP. The inputs and outputs of the first layer are D-dimensional and we apply dropout with $p = 0.5$. The last layer of the MLP is the classification layer. Lastly, we apply a sigmoid function before applying multi-label margin loss to train the model.

1.3. Relational Transformer

Transformers [11] are a popular class of models in deep learning. They are effective at capturing relationships between far-apart elements in a set or a sequence. In this work, we use transformers as a set summarization model.
Multi-head self-attention Transformer: Specifically, we utilize a multi-head self-attention (MSHA) transformer model. The MHSA transformer contains one encoder and three decoder layers by default. We do not use any positional en-



person is sitting on sofa/couch,
person is not looking at sofa/couch,
person is behind sofa/couch,
person is sitting on chair,
person is not looking at chair,
person is beneath chair,
person is holding cup/glass/bottle,
person is looking at cup/glass/bottle,
person is in front of cup/glass/bottle

I need to know what actions person executed to arrive in the state explained in the context. Select subset of action numbers between [1] and [157]. [1] = Holding some clothes, [2] = Putting clothes somewhere, [3] = Taking some clothes from somewhere, [4] = Throwing clothes somewhere, … [157]

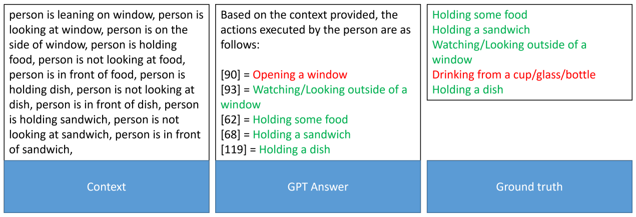
Figure 2. The context description and the textual prompt used for the GPT-3.5 turbo model.

coding as we are summarising a set. Given the set of relational representation of an image $\mathcal{R}_i = r_1, r_2, \cdots, r_n$, the transformer model outputs a Tensor of size $n \times d$ where d is the size of the relational representation. Afterward, we use max-pooling to obtain an image representation vector x_r. A visual illustration of this model is shown in Fig. 1 (left).
Cross-attention Transformer: Similar to the multi-head self-attention transformer, we use one encoder and three decoder layers. The inputs to the transformer encoder comprise concatenated visual and semantic features of a human and objects $[x_h, y_h, x_o, y_o]$, excluding the union features x_u.

1.4. Relational GPT-3.5 Past Action Inference

GPT and later versions [1, 8, 9] have revolutionized the AI field by solving many natural language processing and reasoning tasks. Here, we use the GPT-3.5 turbo version to perform abductive past action inference. To do this, we generate a query prompt as well as a contextual description for each image using the ground truth relational annotations based on the subject-predicate-object triplet relation. In contrast to the all other methods, we utilize the ground truth predicate label for GPT-3.5. An example of the contextual description and textual prompt is shown in Figure 2. In addition, an answer generated by GPT-3.5 is shown in Figure 3. We specifically created the prompt such that GPT-3.5 responses are constrained to the ground truth action sets within the dataset. Based on the responses from the GPT-3.5 model, we construct the score vector where the predicted action is marked with a score of 1 or 0 otherwise. We call this hard matching as we add 1 if and only if the GPT-3.5 model outputs the exact action class name given in the input prompt.

The GPT-3.5 model is able to generate reasonable answers in some images (see Fig 3). However, most of the time GPT-3.5 answers are either overly conservative or aggressive. For example, GPT responds *"There is not enough information given in the context to determine the specific actions the person executed to arrive in the described state"* and in some instances, it selects all action classes. This may be the main reason for the poor performance of GPT-3.5. However, it should be noted that the GPT model is fed with more information than all other baselines as we also provide the predicate relation to the GPT-3.5 model. We also note that the GPT-3.5 + CLIP (Text) model with both soft and hard scores performs better than the hard score method. Assuming that large language models such as GPT-3.5 are capable of human-like reasoning, we can perhaps suggest that abductive inference requires more than text-based reasoning and commonsense reasoning. Given the fact that pure rule-based inference performs better than GPT-3.5 with lesser information may suggest that GPT-3.5 is not suited for abductive past action inference due to it not having a detailed understanding of some of the human behaviors and effects of human actions.



person is leaning on window, person is looking at window, person is on the side of window, person is holding food, person is not looking at food, person is in front of food, person is holding dish, person is not looking at dish, person is in front of dish, person is holding sandwich, person is not looking at sandwich, person is in front of sandwich,

Based on the context provided, the actions executed by the person are as follows:

[90] = Opening a window
[93] = Watching/Looking outside of a window
[62] = Holding some food
[68] = Holding a sandwich
[119] = Holding a dish

Holding some food
Holding a sandwich
Watching/Looking outside of a window
Drinking from a cup/glass/bottle
Holding a dish

Figure 3. Answer generated by GPT-3.5 turbo model. The correct answers are shown in green color whereas false positives and negatives are shown in red. This example is cherry-picked.

1.5. VILA Fine-tuning for Past Action Infernce

With the proven success of Large Language Models (LLMs) across various NLP tasks, recent research has extended their capabilities towards vision tasks, resulting in the development of Visual Language Models (VLMs). These models are typically enhanced through prompt-tuning (where LLMs are frozen) or fine-tuning methods. We employ a fine-tuned VLM, VILA [4], which has not only advanced state-of-the-art performance in vision tasks but also retains robust capabilities in text processing. VILA demonstrates strong reasoning abilities in multi-image analysis, contextual learning, and zero/few-shot learning scenarios. Hence, we leverage VILA for the task of abductive past action set inference.

2. Details on Dataset Creation

How to generate action sets and sequences? To obtain the ground truth action set $\mathcal{A}$ for an image in the Action Genome dataset using the Charades action labels, we first compute the time t for each individual frame within a video sequence by using the formula: $t = \frac{v_d}{n}$, where v_d and n denote the video duration and the number of frames in the video respectively. Then, we multiply the current frame number f_n with t to obtain the current time, $t_c = t \times f_n$.
Action sets: As each video contains multiple actions, we

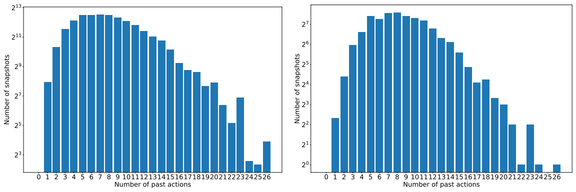


Figure 4. Number of snapshots (in log_2) for sets of n past actions in the Action Genome test set. (a) – *Abduct at T*, (b) – *Abduct last snapshot*

check whether the current time of the frame t_c, falls within the start t_s and end t_e time of the action. If it does, we add the ground truth action label to the action set $\mathcal{A}_n$ for the image. To obtain the ground truth action set for the t^{th} image, we combine all previous action sets from $t = 1$ up to and including the t^{th} image to form the set.

Action sequences: We sort the start time t_s of the actions contained in the video in ascending order. Then, for each image, if the current time of the frame is greater than the start time of the action ($t_c \geq t_s$), we add it to the sequence.

We provide details on the number of images for a set of n past actions in the AG dataset for these setups in Figure 4. As can be seen from these statistics, the majority of the images have more than five actions and some images have as many as 26 actions.

3. Implementation Details

We use FasterRCNN [10] with a ResNet-101 [3] backbone to extract human and object features from each image based on the ground truth person and object bounding boxes provided by AG for all object-relational models. We load pre-trained weights provided by [2] that were trained on the training set of AG which obtained 24.6 mAP at 0.5 IoU with COCO metrics. The parameters of the FasterRCNN during training and inference are fixed for the abductive past action inference task. Our default human and object visual representations have 512 dimensions obtained from 2048 dimensional visual features from the FasterRCNN. We use linear mappings to do this. During training, we train the models for 10 epochs and set the batch size to 1 video (there are many frames in a video). We assume the frames are i.i.d. Note that even though there are multiple images in a batch, the images are processed in parallel and individually for the transformer and graph models respectively. There is no sharing of information between images. We use the AdamW [5] optimizer with an initial learning rate of 1e-5 along with a scheduler to decrease the learning rate by a factor of 0.5 to a minimum of 1e-7. We utilize Glove [6] word embedding of size 200 for the human and object semantic features. In addition, gradient clipping with a maximal norm of 5 is applied. Moreover, we report the mean across 3 different runs for each configuration to ensure we report the most accurate performance of our models. All models (except end-to-end and ViT) are trained on a single RTX3090 or A5000 GPU. For CLIP, we use publicly available implementations [7]. We use the public API of OpenAI for GPT 3.5 models.

Table 1. Abductive past action sequence prediction using the proposed methods on the *Abduct at T* setup.

Model	**Accuracy**	
Human performance	14.00	
	GRU	**Transformer**
Relational MLP	9.43±0.13	9.59±0.06
Relational Self Att. Transformer	9.72±0.06	9.95±0.07
Relational Cross Att. Transformer	9.69±0.18	9.96±0.12
Relational GNNED	9.81±0.05	10.11±0.19
RBP	10.48±0.05	10.22±0.12
BiGED	**10.54±0.15**	**10.1±0.14**

4. Additional Experiments

4.1. Abductive Past Action Sequence Prediction

Next, we formulated the abductive past action sequence prediction task based on the *Abduct at T* setup. We attached a GRU / transformer decoder to our existing object-relational models. To train both sequence prediction models, we freeze the object detector and relational model ($\phi()$). Then, we use the relational vector x_r and action distribution obtained from $\phi_c()$ as the initial hidden state and pass it to the GRU respectively. The transformer decoder takes non-pooled relational features (a matrix of size $n \times d$) as the key, value, and max-pooled relational features x_r as the query. The output of these models is fed into a linear classifier to produce action sequences autoregressively. The results of these models are reported in Table 1. The BiGED model obtains slightly better performance than the rest. Although the performances of these models are suboptimal, we note that humans are also unable to obtain satisfactory results (only 14.00% accuracy). As we are constrained to only utilize available information in a single frame, the solution contains a substantial amount of sequence permutations. Therefore, the task is extremely challenging. The poor human performance also suggests how humans may use abduction. Perhaps humans do not resolve causal chains when performing abduction as it is a very challenging task.

We use the Hamming Loss to evaluate the action sequence prediction models as follows:

$$H = \frac{1}{N * L} \sum_{n=1}^{N} \sum_{l=1}^{L} [y_l \neq \hat{y}_l] \tag{5}$$

where N is the total number of samples and L is the sequence length. Finally, for a given sample, the accuracy is $(1 - H) \times 100$.

Table 2. Ablation on graph affinity using *Abduct at T* setup.

Model	mAP	R@10	mR@10
Jaccard Vector Similarity	**35.75**	**60.55**	**44.37**
Cosine Similarity	34.17	57.98	41.97
Dot product	28.81	54.68	38.38

Table 3. Ablation study for the impact of semantic features and scheduler on the abductive past action set inference for the *Abduct from current and previous images* setup using self-attention transformer.

Model	mAP	R@10	mR@10
Visual only	21.42±0.13	46.44±0.12	34.24±0.42
Visual + scheduler	21.93±0.16	47.04±0.44	34.80±0.47
Visual + semantic	35.40±0.16	68.47±0.06	54.90±0.52
Visual + semantic + scheduler	**35.77±0.30**	**69.16±0.50**	**55.70±0.47**

4.2. Ablation Study

Ablation on graph affinity function: By default, we use the Jaccard Vector Similarity as the affinity $W_A(i, j)$ for the GNNED and BiGED models. Here, we ablate the impact of this design choice by comparing it with cosine similarity and dot product. As can be seen from the results in Table 2, the Jaccard Vector Similarity (JVS) obtains better results than cosine similarity and dot product. This behavior can be attributed to the fully differentiable and bounded nature of JVS in contrast to the dot product or cosine similarity.

Impact of semantic features and learning scheduler: Apart from the two different setups mentioned, we also use a third setup for ablations. In the third setup, the action sets are formed from the current and previous images which form the ground truth denoted by $\mathcal{A} = \{\mathcal{A}_{t-1} \bigcup \mathcal{A}_t\}$ for faster experimentation. We retrain all object-relational models with the corresponding past action set obtained from the current and previous images. We perform ablation studies on the relational self-attention transformer based on this setup. These findings can also be generalized to the other setups as mentioned earlier.

We evaluate the effect of visual and semantic (Glove [6]) features in Table 3. The use of semantic features provides a huge performance boost across all metrics. We attribute the performance increase to the contextual information provided by the semantics. The semantics of objects enable the model to effectively identify and relate actions, providing a more intuitive means for reasoning about these actions. It is also interesting to see the impact of the learning rate scheduler which provides considerable improvement for the transformer model. Therefore, we use semantics and the learning rate scheduler for all our models.

Table 4. The object-relational model parameters for the abductive past action inference task.

Model	Parameters
Relational MLP	13.4M
Relational Self Att. Transformer	101.2M
Relational Cross Att. Transformer	65.9M
Relational GNNED	80.7M
RBP	373.4M
BiGED	213.6M

4.3. Object-Relational Model Parameters

The proposed object-relational model parameters are shown in Table 4. The rule-based inference model does not have any parameters and is therefore omitted from the table. Based on the results shown earlier, we note that the GNNED model obtains better performance than the transformer model even though it has lesser parameters. In addition, our proposed BiGED model has lesser parameters and performs comparable to or better than the RBP model. These further demonstrate the effectiveness of the proposed GNNED, RBP, and BiGED models for the challenging task of abductive past action inference.

4.4. Qualitative Results

We compare qualitative results for the abductive past action set prediction task in Figure 5. Depending on the number of past action labels an image has, we take the same number of top-k predicted actions from each model. All models demonstrate their ability to perform abductive past action inference. In the first image, there are objects such as a person, laptop, table, cup, and dish. In the second image, there are objects such as a person, floor, blanket, bag, and vacuum. In both scenarios, RBP and BiGED demonstrate that they can infer past actions more accurately.

References

[1] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020. 2

[2] Yuren Cong, Wentong Liao, Hanno Ackermann, Bodo Rosenhahn, and Michael Ying Yang. Spatial-temporal transformer for dynamic scene graph generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 16372–16382, 2021. 3

[3] 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. 3

[4] Ji Lin, Hongxu Yin, Wei Ping, Yao Lu, Pavlo Molchanov, Andrew Tao, Huizi Mao, Jan Kautz, Mohammad Shoeybi,

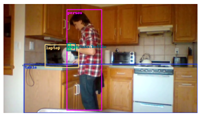
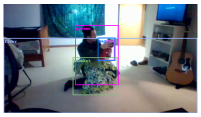
Snapshot	GT	Rule-based	MLP	Transformer	GNNED	RBP	BiGED
	Holding a dish Taking a cup/glass/bottle from somewhere Holding a cup/glass/bottle of something Working/Playing on a laptop Working at a table Watching a laptop or something on a laptop Drinking from a cup/glass/bottle	Putting something on a table Taking a cup/glass/bottle from somewhere Sitting at a table Working at a table Working/Playing on a laptop Holding a laptop Opening a laptop Sitting in a chair Drinking from a cup/glass/bottle Taking a dish/es from somewhere	Holding a dish Putting something on a shelf Walking through a doorway Closing a closet/cabinet Opening a closet/cabinet Putting a dish/es somewhere Taking a dish/es from somewhere Someone is smiling Someone is sneezing Someone is standing up from somewhere	Holding a dish Taking a cup/glass/bottle from somewhere Putting something on a table Closing a closet/cabinet Opening a closet/cabinet Putting a dish/es somewhere Taking a dish/es from somewhere Someone is cooking something Someone is smiling Someone is standing up from somewhere	Holding a dish Taking a cup/glass/bottle from somewhere Putting something on a table Working/Playing on a laptop Tidying up a table Putting a cup/glass/bottle somewhere Drinking from a cup/glass/bottle Putting a dish/es somewhere Taking a dish/es from somewhere Someone is standing up from somewhere	Holding a dish Taking a cup/glass/bottle from somewhere Holding a cup/glass/bottle of something Working/Playing on a laptop Putting something on a table Watching a laptop or something on a laptop Drinking from a cup/glass/bottle Putting a cup/glass/bottle somewhere Opening a closet/cabinet Taking a dish/es from somewhere	Holding a dish Taking a cup/glass/bottle from somewhere Holding a cup/glass/bottle of something Working/Playing on a laptop Putting something on a table Walking through a doorway Putting a cup/glass/bottle somewhere Putting a dish/es somewhere Taking a dish/es from somewhere Someone is standing up from somewhere
	Holding a blanket Holding a bag Snuggling with a blanket Sitting on the floor Lying on the floor Holding a vacuum Someone is awakening somewhere	Putting a broom somewhere Taking a broom from somewhere Throwing a broom somewhere Tidying up with a broom Fixing a light Turning on a light Turning off a light Drinking from a cup/glass/bottle Holding a cup/glass/bottle of something Pouring something into a cup/glass/bottle	Holding a blanket Holding some clothes Putting clothes somewhere Holding a towel/s Putting a blanket somewhere Taking a blanket from somewhere Walking through a doorway Someone is smiling Someone is sneezing Someone is standing up from somewhere	Holding a blanket Holding a bag Holding some clothes Putting clothes somewhere Taking some clothes from somewhere Opening a bag Taking a bag from somewhere Walking through a doorway Someone is smiling Someone is standing up from somewhere	Holding a blanket Holding a bag Snuggling with a blanket Opening a bag Putting a bag somewhere Taking a bag from somewhere Putting a blanket somewhere Taking a blanket from somewhere Tidying up a blanket/s Someone is standing up from somewhere	Holding a blanket Holding a bag Snuggling with a blanket Sitting on the floor Holding a vacuum Opening a bag Taking a bag from somewhere Taking a blanket from somewhere Drinking from a cup/glass/bottle Tidying something on the floor	Holding a blanket Holding a bag Snuggling with a blanket Sitting on the floor Holding a vacuum Opening a bag Taking a bag from somewhere Throwing a blanket somewhere Fixing a vacuum Someone is standing up from somewhere

Figure 5. Manually selected qualitative results produced by each model on the abductive past action set inference: *Abduct last image* setup on the AG test dataset. The first column shows the image followed by their corresponding ground truth past actions. The remaining columns display the actions predicted by each model, with correct predictions highlighted in green and incorrect predictions highlighted in red.

and Song Han. Vila: On pre-training for visual language models, 2023. 2

[5] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 3

[6] Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014. 1, 3, 4

[7] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021. 3

[8] Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. Improving language understanding by generative pre-training. 2018. 2

[9] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019. 2

[10] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28, 2015. 3

[11] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 1