Program Generation from Diverse Video Demonstrations

Anthony Manchin, Jamie Sherrah, Qi Wu, Anton van den Hengel

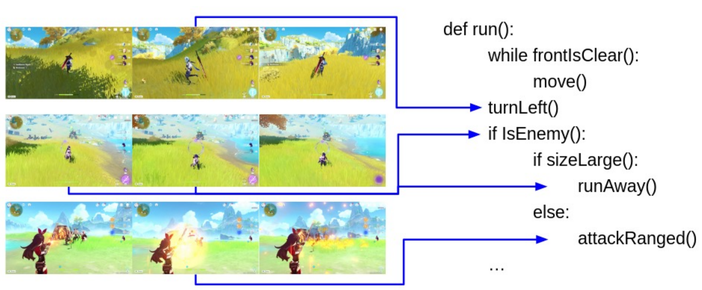
ABSTRACT

The ability to use inductive reasoning to extract general rules from multiple observations is a vital indicator of intelligence. As humans, we use this ability to not only interpret the world around us, but also to predict the outcomes of the various interactions we experience. Generalising over multiple observations is a task that has historically presented difficulties for machines to grasp, especially when requiring computer vision. In this paper, we propose a model that can extract general rules from video demonstrations by simultaneously performing summarisation and translation. Our approach differs from prior works by framing the problem as a multi-sequence-to-sequence task, wherein summarisation is learnt by the model. This allows our model to utilise edge cases that would otherwise be suppressed or discarded by traditional summarisation techniques. Additionally, we show that our approach can handle noisy specifications without the need for additional filtering methods. We evaluate our model by synthesising programs from video demonstrations in the Vizdoom environment achieving state-of-the-art results with a relative increase of 11.75% program accuracy on prior works.

INTRODUCTION

We seek to create a model that can generate executable code by inferring the specifications from multiple visual demonstrations. Predominately, research has primarily focused on the generation of code, *given* the desire specifications. However, the problem becomes much more difficult when the model is also required to infer the specifications for itself. This task presents a unique challenge as it requires a model to accurately detect the relevant semantics of a demonstration, understand the relationships between demonstrations, define a set of specifications that captures this information, and finally generate a program that satisfies these specifications. Our contributions are as follows.

- We present video-to-text transfer transformer for the task of executable program generation from video demonstrations capable of generating programs from a long, disjointed set of demonstrations without the need for a summarised representation of the average demonstration.
- We evaluate the effects of noisy specifications on program accuracy and show that our model is robust to significant levels of erroneous detections.
- We evaluate our model's ability to generate programs from partially observable, visually complex demonstrations. We achieve increases of 9% and 11.75% for exact and aliased program accuracy relative to previous state-of-the-art. This translates to absolute increases of 5.3% and 7.7% respectively and represents the largest single increase in performance on this task to date.



METHODOLOGY

PROGRAM DEFINITION

A program $\pi_\theta(s_t) = a_t$ is defined as a deterministic function, which given an input of state $s \in S$ at time t, returns an action $a \in A$. For this task we limit the parameters of the program to a vectorized domain specific language (DSL), which we donate as $\theta \in \Theta$ which consist of perception primitives, action primitives and control flow statements. These parameters are what would typically be referred to as the 'code' of the program.

```
def run():
    while isTargetDemon:
        moveForward()
    if  isTargetRevenant:
        moveRight()
    else:
        if  isTargetHellKnight:
            shoot()
        else:
            moveLeft()
```

Control flow statements
Percept primitives
Action primitives

MODEL OVERVIEW

Our model can be separated into two main sections; i) the semantic encoder and ii) the program generator network. We approach this problem of generating an executable program from video demonstrations as a combined translation and summarisation task. Our approach frames the problem as a multi-sequence-to-sequence task thus eliminating the inherent probability of information loss associated with summarising multiple diverse demonstrations.

SEMANTIC ENCODER

Given a set of video demonstrations $V = \{v_i\}_{i=1}^{k}$, we desire the corresponding perceptions p and actions a sequences for each demonstration.

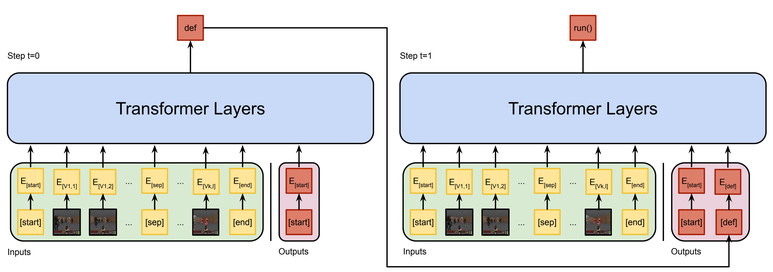
$$p_{i,j} = MLP(CNN(v_{i,j}))$$
$$a_{i,j} = MLP(CNN(v_{i,j}), CNN(v_{i,j+1}))$$

Having obtained the predicted actions $a_{i,j}$ and perceptions $p_{i,j}$ for each frame of every video, we are able to use these to create semantic tokens which completely encapsulates all the required information from each frame.

$$\Psi_{i,j} = \sum_{n=0}^{n} pa_n \times 2^n$$

PROGRAM GENERATION NETWORK

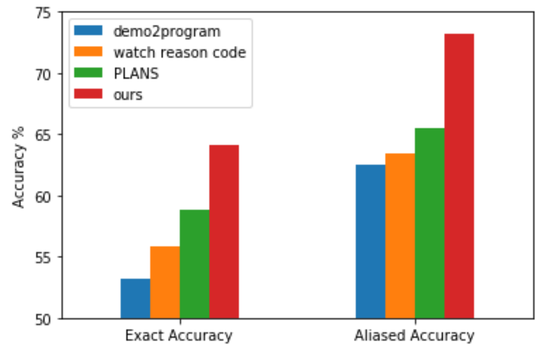
Our program generator is an encoder-decoder transformer network. We utilise the "Text-to-Text Transfer Transformer" which is possible due to the visual language created by the semantic encoder. The self-attention layers of the network allow the generator to simultaneously perform summarisation and translation.



RESULTS

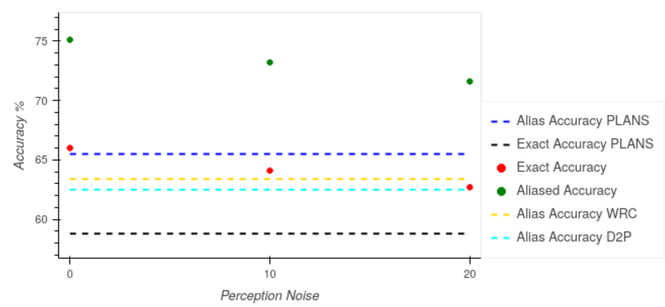
ALIASED AND EXACT ACCURACY

We consider a program to be an exact match if, and only if, the synthesised parameters $\hat{\theta}$ is an exact match to that of the instantiated ground truth parameters θ^*. While the exact accuracy is a simplistic measure of the performance of our model, it does not account for the ambiguity of the program space. It is possible to exploit the simplistic syntax of our DSL and enumerate multiple variations of the code following a set of defined rules. As such we provide results for both exact and aliased accuracy.



NOISE ABLATION

While our model is clearly capable of inductive reasoning over multiple demonstrations, we consider the implication of noise with respect to its ability to accurately generate programs. Previous approaches that utilise rule-based solvers have been highly sensitive to input noise. This inspired us to test our model's ability to deal with noise in its inputs by devising a noise ablation study. By separately predicting the actions and perceptions with two distinct networks we can vary the accuracy of the perception predictions independently of the actions. Our semantic encoder in this setup has independent action and perception encoders.



AUTHOR INFORMATION

Lead Author: Anthony Manchin
anthony@apollolabs.com.au

Affiliations:
Anthony Manchin: UofA, AIML, Apollo Labs
Jamie Sherrah and Qi Wu: UofA and AIML
Anton van den Hengel: UofA