Language-Conditional Imitation Learning

Julian Skirzyński
Dept. of Computer Science and Engineering,
University of California,
San Diego, CA 92093
jskirzynski@ucsd.edu

Bobak Baghi and **David Meger**
School of Computer Science,
McGill University,
Montreal, QC H3A 0G4
{bobhb,dmeger}@cim.mcgill.ca

Abstract

This work introduces the Language-Conditional Imitation Learning (L-CIL) algorithm which auto-encodes input language and learns a distribution of output actions conditioned on the resulting context. We test L-CIL in autonomous driving tasks. For tasks seen during training, our method is on par with the state-of-the-art in conditional IL. More importantly, it largely outperforms other conditional methods when tested on an unseen task, likely generalizing due to its use of language conditioning to discover the proper similarity between train and test sentences. We conclude that our research may be stimulating to the field of Human-Computer Interaction or robotics, where there are continuing opportunities to explore the use of language to generalize behavior.

1 Introduction

Imitation learning (IL) is one of the major tools used in robotics and fields for which it is easier for humans to convey information by *showing* what to do rather than to express it on a different level of generality. The simplest formulation assumes that we present an agent with a set of state-action pairs and hope it will learn how to approximate the function that generated this data, and thus, master the general set of presented skills. Until recently, learning multiple behaviors at once was highly problematic, as information included in the state alone was sometimes insufficient to elicit proper actions. Conditional Imitation Learning, which made the process of imitation learning to depend on additional data, solved that issue (Codevilla et al., 2018; Chowdhuri et al., 2019; Mehta et al., 2018). It is possible that the benefits would be even greater, however, if there was a method that relied not only on *showing*, but also on *telling*. Since language is compositional (the meaning of an expression is determined by its structure and the meaning of its constituents), understanding a set of expressions entails generalization: understanding similar expressions. For artificial agents, however, the effects of compositionality are unclear. The major motivation for this study is thus to investigate generalization with language in conditional imitation learning.

2 Background

2.1 Imitation learning via behavioral cloning

Imitation learning is the problem of finding a policy mimicking transitions provided in a dataset of trajectories $\tau_i = [\phi_0, ..., \phi_{N-1}]$. In our case, trajectories stand for sequences of observation-action pairs, i.e. $\phi_i = (o_i, a_i)$ for $o_i \in \mathcal{O}, a_i \in \mathcal{A}, i = 0, ..., N-1$.

One approach for solving imitation learning problems is *behavioral cloning* (Pomerleau, 1991). This term is used to describe all methods which approximate mappings from the set of states to the set of actions via supervised learning on imitation data. Formally, for finding policy π^* which generated demonstrations $\mathcal{D} = \{(o_i, a_i)\}_{i=0}^N$ we set up a supervised regression problem of the form:

$$\underset{\theta}{\text{minimize}} \sum_t \mathcal{L}\left(\pi_\theta(o_t; \theta), a_t\right) \qquad (1)$$

for $o_t, a_t \in \mathcal{D} = \{(o_i, a_i)\}_{i=0}^N$, a function approximator π_θ defined with parameters θ, and a loss function $\mathcal{L}$.

2.2 Conditional Imitation Learning

The Conditional Imitation Learning (CIL) framework tackles the assumption that proper behavior can be inferred from the representation of the environment in which the expert is taking actions (Codevilla et al., 2018). Codevilla et al. (2018) suggest modelling latent information which additionally explains expert's behavior by vector h and expose the learner to this possibly over-complex

representation through command $c = c(h)$. The optimization problem then becomes:

$$\underset{\theta}{\text{minimize}} \sum_{t} \mathcal{L}\left(\pi_\theta(o_t, c_t; \theta), a_t\right). \qquad (2)$$

The variable c could be an actual command issued in natural language or a form of a signal associated with some behavior (e.g. command "right" or car's blinking light can be both associated with turning). The information carried by c enables the method to distinguish between different behaviors that occur in the same area of the environment's state-space, and at test time gives the possibility to query it to perform them.

3 Related Work

Multiple works on imitation learning relate to our method, e.g. (Ross et al., 2011; Babes et al., 2011; Ho and Ermon, 2016; Duan et al., 2017; Dadashi et al., 2020), but here we only discuss the most relevant literature. A comprehensive overview can be found in the author's thesis (Skirzyński, 2020).

3.1 Imitation learning with conditioning

Prior work attempted learning a finite set of behaviors using a demonstration set that included all of them. Codevilla et al. (2018) considered a branched version of a feed-forward neural network where context modulated which branch of the network would be utilized to predict the action (steering angle and acceleration). Technically, the authors conditioned input data on one-hot encodings $c(h)$ which corresponded to behavior types h. Mehta et al. (2018) enlarged the command vector with visual affordances – quantitative statistics computed from the visual scene that served as the main input to the learning algorithm. Affordances along with action primitives were firstly used as auxiliary tasks that needed to be computed based on the state alone, and then their predictions $c(h)$ conditioned the action module of the network. Chowdhuri et al. (2019) applied principles of conditioning to teach a fleet of model cars to drive in different behavioral modes. Information about the mode was encoded in a form of a binary tensor $c(h)$ that was concatenated with the current image of the environment before being passed to an intermediate layer of a convolutional neural network.

3.2 Instruction Following

Current methods to instruction following draw from the successes of deep learning and apply conditioning directly by computing language and state representations end-to-end. Misra et al. (2017) applied LSTM layers to language input and alongside a history of previous states and actions, conditioned vanilla policy gradient algorithm (Sutton and Barto, 2018) to perform the desired high-level actions. Wang et al. (2019) encoded language with LSTMs and multiplying it by attention matrices conditioned an action module to take appropriate actions. Their algorithm utilized standard reinforcement learning (RL) mechanisms adding a reward from a matching critic that computed alignment between the command and the generated trajectory. In Chen et al. (2019), the authors considered language-conditioned image reconstruction problem to teach a robot navigation in real-life visual urban environment. Instructions transformed into a coherent representation by an LSTM network were concatenated to the output of intermediate layers within an encoder-decoder architecture. This data was used to predict the distribution over the location of a queried item. In Chaplot et al. (2018) language was encoded and mixed with an input image by a fusion model. Specifically, the authors were using Gated-Attention units, and A3C algorithm for policy learning (Mnih et al., 2016). For more references, please see Luketina et al. (2019) survey on language in RL.

4 Method

4.1 Overview

We call our algorithm Language-Conditional Imitation Learning (L-CIL). The input to L-CIL consists of a sum of N sets of M expert trajectories, each generated for a different behavior, and N sets of K sentences describing these behaviors, $K \gg M$. Sentences are randomly assigned to appropriate trajectories resulting in a dataset $\mathcal{D} = \{(o_t, s_t, a_t)\}_{t=1}^{T}$ of observation, descriptive sentence and action triples.

Our algorithm starts by creating a language model to represent words as vectors. It uses word2vec (Mikolov et al., 2013) to construct representations of words based on similarities between their neighborhoods. Sentences are then turned into sequences of vectors obtained using this technique. If v_ϕ is a function approxima-

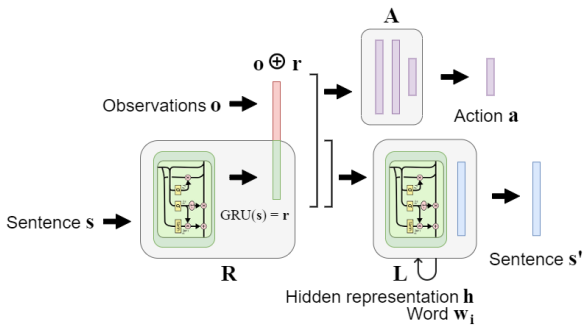

Figure 1: Network architecture for L-CIL[1]

tion for word2vec, and $s_i = \langle w^1_{s_i}, ..., w^{l_{s_i}}_{s_i} \rangle$ is a sentence of length l_{s_i} then it is transformed into $v_\phi(s_i) := \langle v_\phi(w^1_{s_i}), ..., v_\phi(w^{l_{s_i}}_{s_i}) \rangle$. In consequence $\mathcal{D} = \{(o_t, v_\phi(s_t), a_t)\}_{t=1}^{T}$.

With this $\mathcal{D}$ the algorithm begins the optimization process. Let $\mathcal{O}$ be the observation space, $\mathcal{S}$ the discrete sentence space and $\mathcal{A}$ the actions space. Additionally, let $\ell_a(x_1, x_2), \ell_s(x_1, x_2)$ be loss functions that compare actions and sentences representations, respectively, and let $\chi_i(\mathbf{x})$ denote a projection of vector $\mathbf{x}$ on its i-th dimension. Finally, let $F(\cdot, \cdot; \theta)$ be a mapping approximating transformation $(o_t, v_\phi(s_t)) \underset{\theta}{\mapsto} (a_t, v_\phi(s_t))$ through parameters θ, where $o_t \in \mathcal{O}, a_t \in \mathcal{A}, s_t \in \mathcal{S}, t \in [T]$. In mathematical terms, instead of using CIL's objective from equation (2), the algorithm uses:

$$\begin{aligned} \underset{\theta}{\text{minimize}} \sum_t \ell_a \left(\chi_1 \left(F(o_t, v_\phi(s_t); \theta) \right), a_t \right) \\ + \sum_t \ell_s \left(\chi_2 (F(o_t, v_\phi(s_t); \theta)), v_\phi(s_t) \right). \end{aligned} \tag{3}$$

4.2 Implementation

Mapping F is a composition of three modules: a representation module R that maps input to context vectors $(o_t, v_\phi(s_t)) \underset{\theta_R}{\mapsto} r_t$, a language decoder module L that decodes the context to input sentence $r_t \underset{\theta_L}{\mapsto} v_\phi(s_t)$ and an action module A that maps the observation conditioned on the context to an action $(o_t, r_t) \underset{\theta_A}{\mapsto} a_t$. All of the modules are function approximators whose parameters $\theta_R, \theta_L, \theta_A$ make up θ. In this work, L-CIL was implemented as a composition of a feed-forward neural network and an autoencoder. Let $\oplus$ denote vector concatenation. The encoder network implemented the representational module R and the decoder network implemented the language module L. Feed-forward layers operating on a concatenation of $o \oplus r, o \in \mathcal{O}, r = R(o, v_\phi(s); \theta_R)$ for some $s \in \mathcal{S}$ implemented the action module A. The general structure of L-CIL is depicted in Figure 1. The particular structure of the autoencoder network was based on uni-directional Gated Recurrent Unit (Cho et al., 2014). Parameter-wise, ℓ_a was the MSE loss and ℓ_s was the cross-entropy loss. All the feed-forward layers were of size 128, whereas the embedding layer had 32 dimensions. The encoder network was frozen during training after reaching the best encoding loss. A mini-batch contained 128 elements, the learning rate was set to $3e{-5}$ and the weights were updated based on the Adam optimizer. The training time was set to 100 epochs.

5 Experiments

5.1 Setup

We measured the efficacy of L-CIL in driving imitation tasks developed in the Monicar car simulator[2] that uses a 4-dimensional observation and a 2-dimensional action space (Patel, 2019), both continuous.We defined two disjoint collections of behaviors there (see Figure 2), and by dividing them between train and test sets, created 3 experiments. For the **Multi-confusion (MC)** experiment the train and the test set contained behaviors from the **multi-behavior** collection (presented in red), and checked whether a single method can learn to replicate all of them only cued by the provided sentences. For the **Composite-confusion (CC)** experiment the train set and the test set contained behaviors from the **composite-behavior** collection (presented in blue), and checked the imitation capability of a method using behaviors with longer trajectories. For the **Composite-ambiguous (CA)** experiment the test set contained the **ambiguous** behavior made out of partial trajectories for composite behaviors (blue with yellow glow), and the train set comprised behaviors from the **composite-behavior** collection. Here, the algorithms were expected to succeed only if enough information was extracted from the context sentences enabling a never-before-seen behavior to be executed.

The train and test sets (with 80 and 20 trajectories, respectively) established according to the above specification were generated using a hand-made controller. Along with the trajectories we created over 600 000 sentences that described the

[1] Image of the GRU taken from https://colah.github.io

[2] Named after its creator Monica Patel.

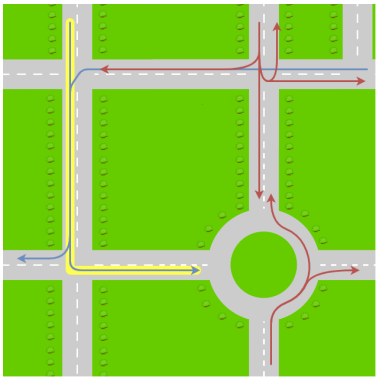

Figure 2: Map for the experiments with sample trajectories.

	Experiment		
Algorithm	MC	CC	CA
BC	0.062*	0.014*	**0.028**
CIL	0.021	**0.008**	1.064*
EL-CIL	**0.017**	0.016*	0.101*
L-CIL	0.029*	0.015*	0.033

* difference to the lowest, bolded value is significant with $p<0.05$

Table 1: Mean error for different experiments and algorithms.

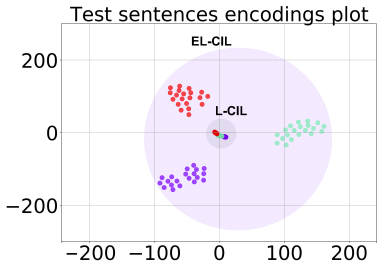

Figure 3: Test sentence embeddings for EL-CIL and L-CIL.

behaviors, using a context-free grammar and a vocabulary with 71 words. The length of the sentences varied between 11 and 31. Although this linguistic input did not allow free expression normally found in the speech, our setup is on par with the most advanced studies on incorporating language in reinforcement or imitation learning (see the Background section). Therefore, our experimentation is innovatory.

5.2 Quantitative results

We compared L-CIL against 3 baseline models: behavioral cloning (BC), CIL and EL-CIL, a version of L-CIL that does not use the additional decoding loss from equation (3). We expected that L-CIL will be the only algorithm that allows generalizing to new behaviors, and it's performance in discriminating between seen behaviors will be reasonable, but lower than CIL's (due to the additional task of decoding language vectors). Table 1 summarizes the obtained results showing the mean action error attained at the end of the training. Firstly, we see that L-CIL indeed exhibits generalization properties, as it gained a threefold improvement over EL-CIL, and a nearly 30-fold improvement over CIL in the **Composite-ambiguous** (**CA**) experiment. Moreover, its performance almost matched this for the standard discrimination tasks (**MC** and **CC** experiments). Surprisingly, however, L-CIL fell short to BC, and further studies are needed to elucidate the reasons for this counter-intuitive result. Secondly, as expected, CIL was on average the best method in the **Confusion** experiments, but its performance did not surpass L-CIL's by a large margin. This confirmed that one-hot vectors differentiate between behaviors the most accurately, but vectors found by L-CIL are not too far off.

5.3 Language encoding analysis

A key enabler of our algorithm's performance is the quality of its encodings. Figure 3 shows T-SNE plots for several sentences' hidden representations produced by EL-CIL and L-CIL and projected onto a 2-dimensional plane. The embeddings are clustered well in both cases. However, for EL-CIL they are very distant and clearly separate, contrary to the embeddings found by L-CIL, which preserve the relations between the sentences. Thanks to that, embeddings of the **ambiguous** behavior found by L-CIL more closely resemble those of the behavior which turns left at the ambiguous area (same as the **ambiguous** behavior), which in turn invokes similar actions. Note that CIL's one-hot vectors are unable to generalize since they belong to independent dimensions of the latent space by definition.

6 Discussion and conclusion

This work presented an algorithm called Language-Conditional Imitation Learning (L-CIL), which optimizes behavioral cloning loss paired with the reconstruction loss for language input. Our experiments revealed that L-CIL successfully imitates multiple training behaviors while exhibiting quality performance with the ambiguous one. The analysis we performed elucidated that L-CIL succeeds in this generalization due to its architectural setup. The auxiliary loss enables it to capture the proper similarity between input sentences and in consequence produce conditioning vectors similar to those successfully used during training. Further studies should measure the real extent to which L-CIL generalizes by modifying the **CA** experiment so that it became insurmountable to BC. Our current insights are nevertheless promising for Human-Computer Interaction or robotics research at large.

References

Monica Babes, Vukosi Marivate, Kaushik Subramanian, and Michael L Littman. 2011. Apprenticeship learning about multiple intentions. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pages 897–904.

Devendra Singh Chaplot, Kanthashree Mysore Sathyendra, Rama Kumar Pasumarthi, Dheeraj Rajagopal, and Ruslan Salakhutdinov. 2018. Gated-attention architectures for task-oriented language grounding. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Howard Chen, Alane Suhr, Dipendra Misra, Noah Snavely, and Yoav Artzi. 2019. Touchdown: Natural language navigation and spatial reasoning in visual street environments. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 12538–12547.

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

Sauhaarda Chowdhuri, Tushar Pankaj, and Karl Zipser. 2019. Multinet: Multi-modal multi-task learning for autonomous driving. In *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1496–1504. IEEE.

Felipe Codevilla, Matthias Miiller, Antonio López, Vladlen Koltun, and Alexey Dosovitskiy. 2018. End-to-end driving via conditional imitation learning. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1–9. IEEE.

Robert Dadashi, Léonard Hussenot, Matthieu Geist, and Olivier Pietquin. 2020. Primal wasserstein imitation learning. *arXiv preprint arXiv:2006.04678*.

Yan Duan, Marcin Andrychowicz, Bradly Stadie, OpenAI Jonathan Ho, Jonas Schneider, Ilya Sutskever, Pieter Abbeel, and Wojciech Zaremba. 2017. One-shot imitation learning. In *Advances in neural information processing systems*, pages 1087–1098.

Jonathan Ho and Stefano Ermon. 2016. Generative adversarial imitation learning. In *Advances in neural information processing systems*, pages 4565–4573.

Jelena Luketina, Nantas Nardelli, Gregory Farquhar, Jakob Foerster, Jacob Andreas, Edward Grefenstette, Shimon Whiteson, and Tim Rocktäschel. 2019. A survey of reinforcement learning informed by natural language. In *International Joint Conferences on Artificial Intelligence*.

Ashish Mehta, Adithya Subramanian, and Anbumani Subramanian. 2018. Learning end-to-end autonomous driving using guided auxiliary supervision. *arXiv preprint arXiv:1808.10393*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

Dipendra Misra, John Langford, and Yoav Artzi. 2017. Mapping instructions and visual observations to actions with reinforcement learning. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1004–1015.

Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. 2016. Asynchronous methods for deep reinforcement learning. In *International conference on machine learning*, pages 1928–1937.

Monica Patel. 2019. Active preference learning using trajectory segmentation. Master's thesis, McGill University, Retrieved from http://digitool.library.mcgill.ca.

Dean A Pomerleau. 1991. Efficient training of artificial neural networks for autonomous navigation. *Neural computation*, 3(1):88–97.

Stéphane Ross, Geoffrey Gordon, and Drew Bagnell. 2011. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 627–635.

Julian Skirzyński. 2020. Language-conditional imitation learning. Master's thesis, McGill University, Retrieved from http://digitool.library.mcgill.ca.

Richard S Sutton and Andrew G Barto. 2018. *Reinforcement learning: An introduction*. MIT press.

Xin Wang, Qiuyuan Huang, Asli Celikyilmaz, Jianfeng Gao, Dinghan Shen, Yuan-Fang Wang, William Yang Wang, and Lei Zhang. 2019. Reinforced cross-modal matching and self-supervised imitation learning for vision-language navigation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6629–6638.