Generalization in Instruction Following Systems

Soham Dan and **Michael Zhou** and **Dan Roth**
{sohamdan,mizho,danroth}@seas.upenn.edu
University of Pennsylvania

Abstract

Understanding and executing natural language instructions in a grounded domain is one of the hallmarks of artificial intelligence. In this paper, we focus on instruction understanding in the blocks world domain and investigate the language understanding abilities of two top-performing systems for the task. We aim to understand if the test performance of these models indicates an understanding of the spatial domain and of the natural language instructions relative to it, or whether they merely over-fit spurious signals in the dataset. We formulate a set of expectations one might have from an instruction following model and concretely characterize the different dimensions of robustness such a model should possess. Despite decent test performance, we find that state-of-the-art models fall short of these expectations and are extremely brittle. We then propose a learning strategy that involves data augmentation and show through extensive experiments that the proposed learning strategy yields models that are competitive on the original test set while satisfying our expectations much better.[1].

1 Introduction

Building agents that can understand and execute natural language instructions in a grounded environment is a hallmark of artificial intelligence (Winograd, 1972). There is wide applicability of this technology in navigation (Chen et al., 2019; Tellex et al., 2011; Chen and Mooney, 2011), collaborative building (Narayan-Chen et al., 2019), and several others areas (Li et al., 2020b; Branavan et al., 2009). The key challenge underlying these and many other applications is the need to understand the natural language instruction (to the extent that it is possible) and ground relevant parts of it in the environment. While the use of deep networks has led to significant progress on several benchmarks (Abiodun et al., 2018) an investigation into the instruction understanding capabilities of such systems remains lacking. We do not know the extent to which these models *truly understand* the spatial relations in the environment, nor their robustness to variability in the environment or in the instructions. This understanding is also important from the viewpoint of safety critical applications , where robustness to variability is essential. While robustness to input perturbations at test-time has been studied in computer vision (Goodfellow et al., 2014) and in certain natural language tasks (Alzantot et al., 2018; Wallace et al., 2019; Shah et al., 2020), it remains relatively elusive in the instruction following task in a grounded environment. This can be attributed to the difficulty in characterizing the different expectations of robustness in this setting, due to the multiple channels of input involved, which semantically constrain each other.

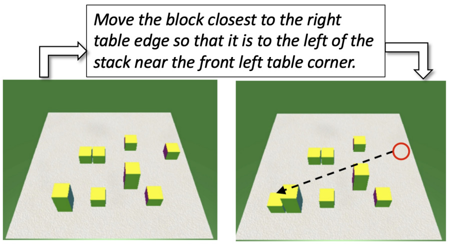
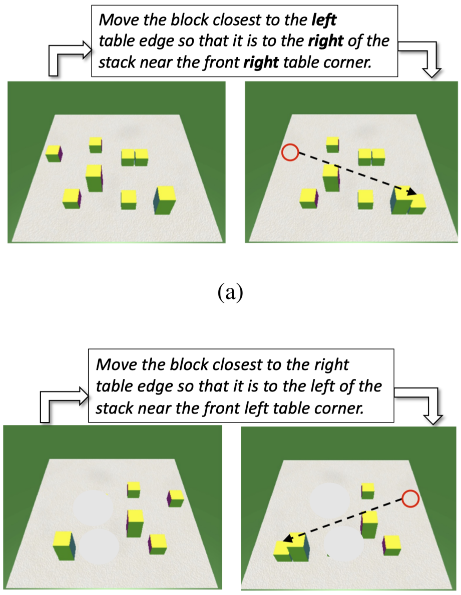

Move the block closest to the right table edge so that it is to the left of the stack near the front left table corner.

Figure 1: Task: Given a configuration of blocks and an instruction, predict the source and target location.

The Blocks World domain is an ideal platform to study the abilities of a system to understand instructions (Winograd, 1972; Bisk et al., 2016; Narayan-Chen et al., 2019; Misra et al., 2017; Bisk et al., 2018; Mehta and Goldwasser, 2019; Tan and Bansal, 2018). Despite being seemingly simple, it presents key reasoning challenges, including compositional language understanding and spatial understanding, that need to be addressed in any instructional domain. In Bisk et al. (2016), the en-

[1] Our code is publicly available at: http://cogcomp.org/page/publication_view/936

Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 976–981
June 6–11, 2021. ©2021 Association for Computational Linguistics


Move the block closest to the **left** table edge so that it is to the **right** of the stack near the front **right** table corner.
Move the block closest to the right table edge so that it is to the left of the stack near the front left table corner.

(a)

(b)

Figure 2: (a) A symmetric example to Fig. 1. The model should respect this symmetry equivariance (SA). (b) A count example (SPC): the model should not overfit on the number of *distractor* blocks.

vironment consists of a number of blocks placed on a flat board. The model is provided with the current configuration of blocks in the environment along with an instruction, and is tasked with executing the instruction by manipulating appropriate blocks. In this work, we follow the more challenging setting in Bisk et al. (2016) where the blocks are unlabeled, necessitating the use of involved referential expressions in the instructions. Fig.1 shows that the instruction and block configuration are semantically dependent, jointly determining the outcome.

Despite the success of top performing models (Tan and Bansal, 2018; Bisk et al., 2016) on the test set for this task, we question if the models are able to reason about the complex language and spatial concepts of this task and generalize or are merely over fitting the test set. To investigate these questions we formulate the following expectations one should have from an instruction following model:
(1) **Identity Invariance Expectation**: The performance of the model on an input should not degrade on slightly perturbing the input.
(2) **Symmetry Equivariance Expectation**: A symmetric transformation of an input should cause an equivalent transformation of model prediction and performance should not degrade.
(3) **Length Invariance Expectation**: The performance of a model should not depend on the length of the input, as long as the semantics is unchanged.

Our expectations complement existing work in three dimensions: (1) is related to adversarial perturbations (Goodfellow et al., 2014) and (2) is related to equivariance of CNNs explored in computer vision (Cohen and Welling, 2016). It is also related to contrast sets (Gardner et al., 2020; Li et al., 2020a) and counterfactual data augmentation (Kaushik et al., 2019). Here, we extend the investigation to this new task of instruction following involving both natural language and an environment, discrete and continuous perturbations and both regression and classification tasks. Contrast (3) is related to Lake and Baroni (2018) where vulnerability to length in a toy sequence-sequence task was demonstrated. Here we show that length-based vulnerability exists in another modality—the number of blocks present on the board, for this much more complicated task.

While these form only a subset of the expectations one might have from an instruction following model, it already allows us to formally characterize some of the dimensions of robustness an instruction following agent must have. As an example, a tiny shift in the location of each block should not affect the model prediction (identity invariance). In Sec. 2, we formulate concrete perturbations to test whether a given model satisfies these expectations. The space of perturbations that we consider have the following attributes: (a) Semantic Preserving or Semantic Altering. (b) Linguistic or Geometric. (c) Discrete or Continuous. We find that both models studied suffer a large performance drop under each of the perturbations, and fall short of satisfying our expectations. We then present a data augmentation scheme designed to better address our expectations from such models. Our extensive experiments in Sec. 2.3 indicate that our learning strategy results in more robust models that perform much better on the perturbed test set while maintaining similar performance on the original test set.

2 Robustness to Expectations

Given the block configuration $W \in \mathbb{R}^{20\times3}$ (three-dimensional coordinate locations of a maximum of 20 unlabeled blocks $B = b_1, ..., b_{20}$ and an instruction I, the model has to move the appropriate block. There are two sub-tasks: (i) predicting the source block to be moved and (ii) predicting the

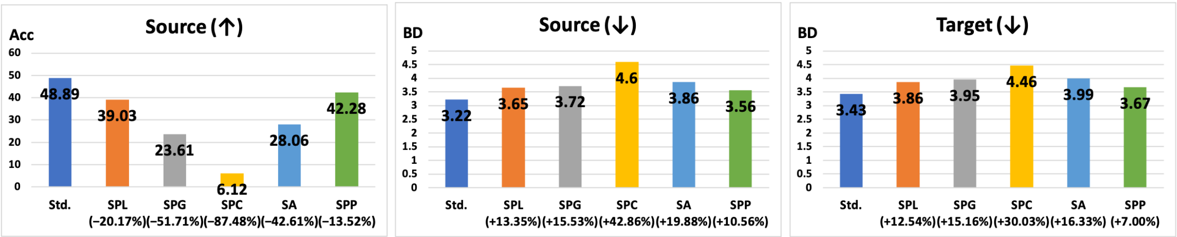


Figure 3: Relative Performance Degradation for the source (classification and regression) and target (regression) sub-tasks. ↑ (↓) denotes higher (lower) is better respectively. Here, SPP uses only one permutation and the degradation becomes more severe when consistency across a larger set of permutations is considered (Appendix A)

target location to move it to. While the target output is always a location $y \in \mathbb{R}^3$, for the source task the model can either predict a particular block $y \in \{1, 2, ..., 20\}$ (Tan and Bansal, 2018) or a particular source location $y \in \mathbb{R}^3$ (Bisk et al., 2016). Let $\mathcal{P}$ denote a perturbation space and (I', W') be the perturbed version of (I, W) under $\mathcal{P}$. Note that (I', W') can be chosen randomly or adversarially as the perturbation which maximizes the loss :

$$(I', W') = \underset{(I',W')\in\mathcal{P}}{\arg\max}\, \ell(f(I', W'), O).$$

Here ℓ denotes a loss function and O denotes the gold source/target location. If the perturbation space is discrete and finite we can simple search over all candidate (I', W') to find the one with the maximum loss. If it is continuous and infinite, we can use a first order method (eg: First Order Gradient Signed Method FGSM (Goodfellow et al., 2014)) to find the adversarial (I', W').

Now we characterize $\mathcal{P}$. Broadly, we have the following two types of perturbations:

(i) **Semantics Preserving (SP)**: Perturbations when applied to either I or W, do not change the meaning of either. Since the modified instruction I' or world state W' is semantically unchanged, the model should perform similarly on the perturbed input. Informally, we want $f(I, W) \approx f(I', W')$ since $I \approx I'$ and $W \approx W'$. SP perturbations can be of the following types:

- **Linguistic (SPL)**: Perturbations that do not change the overall meaning of the instruction. Consider Lexical Substitutions: We identify a list of synonyms for each of the shapes and spatial concepts ($\mathcal{C}$) in this domain.[2] For each test example which contains at least one of these concepts we adversarially pick the one with the highest loss over all combinations of substitutions from the synonyms in $\mathcal{C}$. Since the size of these synonym sets are small, an explicit search over all candidate substitutions is possible, although the search space grows combinatorially with the number of elements of $\mathcal{C}$ in I.
- **Geometric (SPG)**: These perturbations do not change the semantics of the board. Tiny changes in the block locations which preserve the overall semantics of W should not affect model predictions. We perturb each block location slightly in an adversarial direction[3] w.r.t W.
- **Count (SPC)**: We identify *distractor* blocks which do not affect the meaning of the instruction (Fig. 2(b)). Large distance from the source and target location acts as a proxy for this. $\mathcal{P}$ comprises of deleting k blocks where $k \in \{0, 1, 2, ..., N\}$ is chosen adversarially to generate W'. We set $N = 3$.[4]
- **Permutation (SPP)**: These are perturbations where the order in which the block locations are fed to the model, are permuted: $\Pi(B)$=$\{b_{\Pi(1)}, ..., b_{\Pi(20)}\}$. While semantically nothing changes in the input ($(I', W') \equiv (I, \Pi(W))$, where $\equiv$ denotes semantic equivalence), we see models still suffer a large performance drop, even for a random permutation Π.

(ii) **Semantic Altering (SA)**: These perturbations create a new (I', W') pair with different semantics, using a simple transformation that we want the model to be equivariant to. A horizontal mirroring of W with a corresponding change in I (flipping all the *left* concept words to *right* and vice versa)

[2] Examples of synonym sets for shapes in $\mathcal{C}$ are $\{tower(s), stack(s)\}$, $\{block(s), brick(s), box(es)\}$.

[3] according to a FGSM attack with $\epsilon = 0.05$

[4] Addition of such *distractor* blocks at locations far from the source and target locations, form a similar perturbation set that also leads to a significant performance drop for existing models (Appendix A).

$\mathcal{P}$	Model	$Source(Acc)\uparrow$			$Source(BD)\downarrow$			$Target(BD)\downarrow$		
		$Std.$	$Rob.$	$RI(\%)$	$Std.$	$Rob.$	$RI(\%)$	$Std.$	$Rob.$	$RI(\%)$
SPL	M_{std}	48.89	39.03	7.81	3.22	3.65	7.40	3.43	3.86	7.51
	M_{rob}	48.19	**42.08**		3.23	**3.38**		3.35	**3.57**	
SPG	M_{std}	48.89	23.61	97.08	3.22	3.72	4.57	3.43	3.95	6.58
	M_{rob}	48.47	**46.53**		3.17	**3.55**		3.31	**3.69**	
SPC	M_{std}	48.89	6.12	165.85	3.22	4.60	24.56	3.43	4.46	17.71
	M_{rob}	53.19	**16.27**		3.40	**3.47**		3.51	**3.67**	
SA	M_{std}	48.89	28.06	26.23	3.22	3.86	9.84	3.43	3.99	10.78
	M_{rob}	50.14	**35.42**		3.20	**3.48**		3.35	**3.56**	
SPP	M_{std}	48.89	42.28	4.28	3.22	3.56	10.11	3.43	3.65	1.92
	M_{rob}	49.03	**44.09**		3.17	**3.20**		3.56	**3.58**	

Table 1: Standard (Std.) and Robust (Rob.) performance of each model (M_{std}) and its robust counterpart (M_{rob}) for the different perturbations ($\mathcal{P}$). $\uparrow, (\downarrow)$ denotes higher (lower) is better respectively. RI denotes the relative improvement in robust evaluation of the robust model w.r.t. the standard model. BD denotes the block-distance measure and Acc. denotes classification accuracy. The bold numbers are the best robust performance for each $\mathcal{P}$.

, as in Fig: 2(a) should satisfy: if the error on $f(I, W)$ is small, the error on $f(I', W')$ should also be small.

2.1 Model Performance vs Our Expectations

The dataset from Bisk et al. (2016)[5] has 2493 training examples and 720 test examples. We evaluate the performance of our implementation of two models: from Bisk et al. (2016) and from Tan and Bansal (2018). One important difference between the two models is that while both models treat the target subtask T as a regression task (trained and evaluated using a normalized mean squared error called *block distance* BD), Tan and Bansal (2018) treats the source subtask as a classification task S_{cls} (trained using cross entropy loss as ℓ and evaluated using classification accuracy Acc.) while Bisk et al. (2016) treats it as a regression task S_{reg} (trained and evaluated using BD). We use both models for the source and the Bisk et al. (2016) model for the target subtask. We compare model performance on the original test set using standard evaluation and on the perturbed test set using a robust evaluation measure. The robust evaluation measure for S_{reg} and T is $\max(BD(f(I, W), O), BD(f(I', W'), O))$ and $\min(Acc(f(I, W), O), Acc(f(I', W'), O))$ for S_{cls}. This robust evaluation formulation is motivated by the requirement that models perform well on both the original and the perturbed examples. From Fig. 3 we see that models suffer a large performance drop of upto 87.48%, 42.86% and 30.03% for the source-classification,-regression and target subtasks respectively, over different $\mathcal{P}$.

2.2 Adversarial Data Augmentation

In this section we show that a simple data augmentation strategy improves model performance under robust evaluation on the perturbed test set. For each input (I, W) in the training data we add another example which is adversarial:

$$(I', W') = \underset{(I', W') \in \mathcal{P}}{\arg\max}\ \ell(f(I', W'), O).$$

This perturbation set $\mathcal{P}$ used in training is the same one that is used for robust test evaluation. When $\mathcal{P}$ is continuous (eg: SPG), we use the FGSM attack to solve this maximization and obtain (I', W'). When $\mathcal{P}$ is discrete (eg: SPL, SPC) we search over the perturbation space to find the perturbation with the highest loss. We train the model on a combined dataset consisting of both the original train-set and the adversarially augmented data. This is an extension of Adversarial Training (Madry et al., 2017) to the instruction following task for (i) both discrete and continuous perturbations (ii) both regression and classification tasks.

2.3 Results

In this section we show the benefits of adversarially augmented robust training. Consider the models M_{std} from Bisk et al. (2016) and Tan and Bansal (2018) which were shown to perform poorly under robust evaluation in Sec. 2.1. Here we compare their performance with their robustly trained variants M_{rob}. For all models we perform standard

[5] https://groundedlanguage.github.io

evaluation and robust evaluations for each perturbation type. This is done for the source (classification and regression) and target sub-tasks. In Table 1 we show the results under the different settings, averaged over 5 runs. For every perturbation category and for all sub-tasks, we see that the robust models (i) outperform their standard counterparts in terms of robust evaluation metric and (ii) in some cases even on standard evaluation. Thus, knowledge-free robust training framework can produce models which are less brittle to perturbations with competitive standard performance on the original test set.

3 Conclusion

In this paper we formulated the performance expectations for an instruction following system. Based on these expectations, we created several categories of perturbations and showed that existing models fail spectacularly on them. We then demonstrated the benefits of adversarial data augmentation on each perturbation category.

4 Acknowledgments

This work was supported by Contracts W911NF-15-1-0461 and FA8750-19-2-0201 with the US Defense Advanced Research Projects Agency (DARPA), the Army Research Office under Grant Number W911NF-20-1-0080, and by ONR Contract N00014-19-1-2620. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Department of Defense or the U.S. Government.

References

Oludare Isaac Abiodun, Aman Jantan, Abiodun Esther Omolara, Kemi Victoria Dada, Nachaat AbdElatif Mohamed, and Humaira Arshad. 2018. State-of-the-art in artificial neural network applications: A survey. *Heliyon*, 4(11):e00938.

Moustafa Alzantot, Yash Sharma, Ahmed Elgohary, Bo-Jhang Ho, Mani Srivastava, and Kai-Wei Chang. 2018. Generating natural language adversarial examples. *arXiv preprint arXiv:1804.07998*.

Yonatan Bisk, Kevin J Shih, Yejin Choi, and Daniel Marcu. 2018. Learning interpretable spatial operations in a rich 3d blocks world. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Yonatan Bisk, Deniz Yuret, and Daniel Marcu. 2016. Natural language communication with robots. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 751–761.

Satchuthananthavale RK Branavan, Harr Chen, Luke S Zettlemoyer, and Regina Barzilay. 2009. Reinforcement learning for mapping instructions to actions. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 1-Volume 1*, pages 82–90. Association for Computational Linguistics.

David L Chen and Raymond J Mooney. 2011. Learning to interpret natural language navigation instructions from observations. In *Twenty-Fifth 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.

Taco Cohen and Max Welling. 2016. Group equivariant convolutional networks. In *International conference on machine learning*, pages 2990–2999.

Matt Gardner, Yoav Artzi, Victoria Basmov, Jonathan Berant, Ben Bogin, Sihao Chen, Pradeep Dasigi, Dheeru Dua, Yanai Elazar, Ananth Gottumukkala, et al. 2020. Evaluating models' local decision boundaries via contrast sets. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings*, pages 1307–1323.

Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. 2014. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*.

Divyansh Kaushik, Eduard Hovy, and Zachary Lipton. 2019. Learning the difference that makes a difference with counterfactually-augmented data. In *International Conference on Learning Representations*.

Brenden Lake and Marco Baroni. 2018. Generalization without systematicity: On the compositional skills of sequence-to-sequence recurrent networks. In *International Conference on Machine Learning*, pages 2873–2882. PMLR.

Chuanrong Li, Lin Shengshuo, Zeyu Liu, Xinyi Wu, Xuhui Zhou, and Shane Steinert-Threlkeld. 2020a. Linguistically-informed transformations (lit): A method for automatically generating contrast sets. In *Proceedings of the Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 126–135.

Yang Li, Jiacong He, Xin Zhou, Yuan Zhang, and Jason Baldridge. 2020b. Mapping natural language instructions to mobile ui action sequences. *arXiv preprint arXiv:2005.03776*.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. 2017. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*.

Nikhil Mehta and Dan Goldwasser. 2019. Improving natural language interaction with robots using advice. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1962–1967.

Dipendra Misra, John Langford, and Yoav Artzi. 2017. Mapping instructions and visual observations to actions with reinforcement learning. *arXiv preprint arXiv:1704.08795*.

Anjali Narayan-Chen, Prashant Jayannavar, and Julia Hockenmaier. 2019. Collaborative dialogue in minecraft. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5405–5415.

Krunal Shah, Nitish Gupta, and Dan Roth. 2020. What do we expect from multiple-choice qa systems? In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings*, pages 3547–3553.

Hao Tan and Mohit Bansal. 2018. Source-target inference models for spatial instruction understanding. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Stefanie Tellex, Thomas Kollar, Steven Dickerson, Matthew R Walter, Ashis Gopal Banerjee, Seth Teller, and Nicholas Roy. 2011. Understanding natural language commands for robotic navigation and mobile manipulation. In *Twenty-fifth AAAI conference on artificial intelligence*.

Eric Wallace, Shi Feng, Nikhil Kandpal, Matt Gardner, and Sameer Singh. 2019. Universal adversarial triggers for attacking and analyzing nlp. *arXiv preprint arXiv:1908.07125*.

Terry Winograd. 1972. Understanding natural language. *Cognitive psychology*, 3(1):1–191.

A Appendix A: Additional Experiments

In this appendix, we show a few additional experiments to investigate the following claims:

- For the SPP perturbation, an even stricter evaluation that requires consistent predictions for a larger set of permutations over the block indices, further degrades performance of the existing models. Table 2 shows this for the case of two permutations corresponding to each instance. In all cases, adversarial data augmentation helps improve performance under the robust evaluation metric.

$\mathcal{P}$	Model	$Source(BD) \downarrow$		
		$Std.$	$Rob.$	$RI(\%)$
SPP-1	M_{std}	3.22	3.56	10.11
	M_{rob}	3.17	**3.20**	
SPP-2	M_{std}	3.22	3.69	12.74
	M_{rob}	3.18	**3.22**	
$\mathcal{P}$	Model	$Target(BD) \downarrow$		
		$Std.$	$Rob.$	$RI(\%)$
SPP-1	M_{std}	3.43	3.65	1.92
	M_{rob}	3.56	**3.58**	
SPP-2	M_{std}	3.43	3.75	4.27
	M_{rob}	3.57	**3.59**	

Table 2: **SPP Perturbation**: Standard (Std.) and Robust (Rob.) performance of the Bisk et al. (2016) model (M_{std}) and its robust counterpart (M_{rob}) for the different perturbations ($\mathcal{P}$): $SPP - i$ denotes the perturbation set contains i additional permutations of the original input. ↓ denotes lower is better . RI denotes the relative improvement in robust evaluation of the robust model w.r.t. the standard model. BD denotes the block-distance measure. The bold numbers are the best robust performance for each $\mathcal{P}$.

- For the SPC perturbation, a gradual degradation in model performance is observed as the number of *distractor* blocks (whose presence or absence do not affect the semantics of the instruction) removed, are increased. Further, addition of *distractor* blocks also leads to significant performance degradation in Table 3. In all cases, adversarial data augmentation helps improve performance under the robust evaluation metric and sometimes, even under the standard evaluation metric.

$\mathcal{P}$	Model	$Source(Acc) \uparrow$		
		$Std.$	$Rob.$	$RI(\%)$
R(1)	M_{std}	48.89	18.5	52.59
	M_{rob}	53.61	**28.23**	
R(2)	M_{std}	48.89	10.01	61.14
	M_{rob}	53.89	**16.13**	
R(3)	M_{std}	48.89	6.12	165.85
	M_{rob}	53.19	**16.27**	
A(1)	M_{std}	48.89	19.33	22.30
	M_{rob}	49.31	**23.64**	

Table 3: **SPC Perturbation**: Standard (Std.) and Robust (Rob.) performance of the Tan and Bansal (2018) model (M_{std}) and its robust counterpart (M_{rob}) for the different perturbations ($\mathcal{P}$): $A(i)$ and $R(i)$ denotes the addition and removal of i blocks respectively. ↑ denotes higher is better.