Interactive Vision and Language Learning

Amin PARVANEH

A thesis submitted for the degree of

DOCTOR OF PHILOSOPHY

The University of Adelaide

May 24, 2022

Contents

Abstract **xix**

Declaration of Authorship **xxi**

1 Introduction **1**

1.1 Motivations 1

1.1.1 Human Intent Understanding 4

1.1.2 Information Seeking 6

1.1.3 Generalisation 7

1.2 Main Contributions 9

1.3 Thesis Structure 11

2 A Negotiator with Online Value Look-Ahead **13**

2.1 Overview 13

2.2 Introduction 14

2.3 Related Work 18

2.3.1 Goal-Oriented Dialogue 18

2.3.2 Dialogue Systems Design 19

2.4 Price Negotiator 20

2.4.1 Problem Definition 20

2.4.2 Online Value Estimator 21

2.4.3 Hierarchical Recurrent Negotiation Encoder 23

2.4.4 Action Predictor 24

2.4.5 Price Adjuster 25

2.4.6 Language Decoder 26
2.4.7 Copy Mechanism 26
2.4.8 Overall Objective 28
2.4.9 Reinforcement Learning 28
2.5 Implementation Details 29
2.5.1 Dataset 29
2.5.2 Embeddings 29
2.5.3 Training Settings 30
2.5.4 Price Adjustment 30
2.5.5 Evaluation Metrics 31
2.6 Results 33
2.6.1 Value Estimation Experiments 33
2.6.2 Negotiation Dialogue Evaluation 34
2.7 Conclusion and Future Works 39

3 Active Learning by Feature Mixing **41**

3.1 Overview 41
3.2 Introduction 42
3.3 Related Work 46
3.4 Methodology 48
3.4.1 Problem Definition 48
3.4.2 Feature Mixing 49
3.4.3 Optimising the Interpolation Parameter α 51
3.4.4 Candidate Selection 53
3.4.5 Relations Between ALFA-Mix and Other Baselines 53
3.5 Experiments and Results 57
3.5.1 Baselines 57
3.5.2 Experiment Settings 60
3.5.3 Overall Results 63

3.5.4 Ablation Study 64

3.6 Conclusions and Limitations 71

4 Counterfactual Vision and Language Learning 85

4.1 Overview 85

4.2 Introduction 86

4.3 Related Work 89

4.3.1 Visual Question Answering 90

4.3.2 Counterfactuals 91

4.4 Counterfactual Vision and Language (CVL) 92

4.5 Counterfactual Distribution 94

4.5.1 Generating Counterfactuals 95

4.5.2 Counterfactual Loss 97

4.5.3 Further Analysis 98

4.6 Experiments 101

4.6.1 Unimodal Problems 101

4.6.2 Visual Question Answering 103

4.6.3 Vision and Language Navigation 107

4.7 Conclusion 109

5 Counterfactual Vision-and-Language Navigation 111

5.1 Overview 111

5.2 Introduction 112

5.3 Related Work 115

5.4 Background on Counterfactuals 116

5.4.1 Counterfactual Vision and Language Navigation 117

5.5 Methodology 118

5.5.1 Problem Definition 118

5.5.2 Counterfactual Formulation in VLN 120

5.5.3 Counterfactual Distribution Learning and Generation 122

5.6 Experiments 124
  5.6.1 Room-to-Room Navigation 125
  5.6.2 Embodied Question Answering 130
5.7 Conclusions 132

6 Conclusion **137**

Bibliography **139**

List of Figures

1.1 Interactive agents and the corresponding research areas and applications. The main focus of this thesis is interactive vision and language learning which lies in the intersection of all the three areas. 2

1.2 An example of an interactive vision and language learning problem and the corresponding challenges that the agent faces. 4

2.1 An example scenario of our *Price Negotiator* from supervised and RL training. Two agents play the role of sellers and buyers in a visually-grounded bargaining game over an item. The agents need to uncover the underlying value of the item and the attributes of their counterpart (e.g. their assertiveness) to succeed. 14

2.2 The diagram of our *price negotiator* that consists of five main modules: (1) online value estimator (OVE), (2) hierarchical recurrent negotiation encoder (HRNE), (3) action predictor, (4) price adjuster, and (5) language decoder. 16

2.3 The architecture of *Online Value Estimator* that finds K-similar items in an online store. Comparing their multi-modal contextual features $\{\boldsymbol{C}_{i,1}, \ldots, \boldsymbol{C}_{i,K}\}$ with those of the given item $\boldsymbol{C}_i$, it prognosticates a fair agreement price to be considered in the negotiation. See Figure 2.4 for more details about multi-modal embeddings $\Psi_0, \ldots, \Psi_3$. 21

2.4 Multimodal Embedding. The sum of embeddings of the textual resources (title and description) is concatenated with the down-sized visual features obtained from a pre-trained network (ResNet-152 (He et al. 2016)). The concatenated vector is passed through a 2-layer MLP with ReLU activation to create the multimodal embedding. . . 23

2.5 Examples from two models. The left-hand side dialogue is generated from a simple Seq-Seq model which makes a mistake in its offer (from $639 that was agreed upon moves to $720) and produces repetitive, dull responses. On the other hand, our *Price Negotiator* model, the right-hand side dialogue, creates a linguistically more diverse and price-wise reasonable dialogue and uses different negotiation strategies to persuade the opponent. 27

2.6 More samples from our *price negotiator* model. 38

3.1 We propose to form linear combinations (*i.e.* interpolations or mixing) of the features of an unlabelled instance (middle image) and of labelled ones (top and bottom images). The interpolated features are passed through the current classifier. We show that inconsistencies in the predicted labels indicate that the unlabelled instance may have novel features to learn from. 42

3.2 Visualization of sample selection behaviours of various AL methods in the latent space (see the Appendix for additional methods). The larger dots represent the selected samples to label; smaller dots represent unlabelled ones. Our approach finds a candidate set (demonstrated by stars in 3.2a) of unlabelled instances with inconsistencies in their label prediction when interpolated with labelled representations. It selects a diverse set of samples lying close to the all four borders for the labelling (with three zoom-in windows). The demonstration problem is that of identifying 4 classes from MNIST (illustrated above by 4 colours) using a MLP. An initial training set of 200 randomly selected points and their labels was provided, with each method given a budget of 200 additional labels. The features are projected to two-dimensions for visualization. 43

3.3 A comparative depiction of our approach (ALFA-Mix) vs. BADGE vs. adversarial in the latent space: Since ours considers interpolations in the direction of the anchor points and proportional to their distance, it better evaluates the consistency of the predictions in the latent space. When points are less consistent, it is more intuitive to consider them as candidates to be queried (*e.g.* z_2^u in this figure is inconsistent after the interpolation, and hence likely to be queried). 55

3.4 Test accuracy plots across some of the employed settings. Each experiment has been repeated 5 times. 58

3.5 Pairwise comparison (Ash et al. 2020) of different approaches. Lower values shown at each column reveal the better performances of that AL method across all the experiments. Maximum value of each cell is 30, which represents the number of experimental settings. 62

3.6 Ablations of our AL approach. Experiments are conducted on MNIST datasets using an MLP model and a small AL budget. 65

3.7 A summary of the performance of our proposed AL method (ALFA-Mix) compared with state-of-the-art across all the 30 settings considered. Each bar represents the percentage of AL rounds in which our approach outperforms others (lower indicates stronger baseline). It is worth noting that our approach (ALFA-Mix) under-performs others in close to zero cases. 66

3.8 Uncertainty and diversity of the selected samples for labelling. All experiments are done on MNIST dataset using LeNet-5 model and a small budget of size 100. 69

3.9 The t-SNE visualisation of the sample selection of our proposed method on MNIST dataset using LeNet-5. The model is trained based on 500 random labelled set (shown as triangles) and is provided with a budget of size 500 to (depicted as bold circles). 70

3.10 Pairwise comparison of different AL approaches based on the type of data. The maximum value of each cell for each setting is also provided in the captions. 71

3.11 Pairwise comparison of different AL approaches based on different model architectures. The maximum value of each cell for each setting is also provided in the captions. 72

3.12 Pairwise comparison of different AL approaches based on different sizes of budget. The maximum value of each cell for each setting is also provided in the captions. 73

3.13 Pairwise comparison of different AL approaches based on different sizes of budget. The maximum value of each cell for each setting is also provided in the captions. 74

3.14 Small Budget, ViT-Base, DomainNet-Real 75

3.15 Small Budget, ViT-Small, Mini-ImageNet 75

3.16 Small Budget, ViT-Small, CIFAR100 75

3.17 Small Budget, MLP, MNIST 76

3.18 Small Budget, MLP, MNIST, Continue 76

3.19 Small Budget, MLP, EMNIST 76

3.20 Small Budget, MLP, EMNIST, Continue 77

3.21 Small Budget, LeNet-5, MNIST 77

3.22 Small Budget, LeNet-5, MNIST, Continue 77

3.23 Small Budget, LeNet-5, EMNIST 78

3.24 Small Budget-ResNet-18, SVHN 78

3.25 Small Budget, ResNet-18, CIFAR10 78

3.26 Small Budget, ResNet-18, DomainNet-Real 79

3.27 Small Budget, ResNet-18, DomainNet-Real-10 79

3.28 Small Budget, ResNet-18, DomainNet-Real-20 79

3.29 Small Budget, DenseNet-121, SVHN 80

3.30 Small Budget, DenseNet-121, CIFAR10 80

3.31 Small Budget, DenseNet-121, DomainNet-Real 80

3.32 Small Budget, DenseNet-121, DomainNet-Real-10 81

3.33 Small Budget, DenseNet-121, DomainNet-Real-20 81

3.34 Large Budget, MLP, MNIST 81

3.35 Large Budget,MLP, EMNIST 82

3.36 Large Budget, LeNet-5, MNIST 82

3.37 Large Budget, LeNet-5, EMNIST 82

3.38 Large Budget, ResNet-18, SVHN 83

3.39 Large Budget, ResNet-18, CIFAR10 83

3.40 Large Budget, DenseNet-121, SVHN 83

3.41 Large Budget, DenseNet-121, CIFAR10 84

3.42 Small Budget, MLP, OpenML-6 84

3.43 Small Budget, MLP, OpenML-155 84

4.1 The training process with counterfactuals. We infer the posterior on the exogenous variables. Subsequently generate counterfactual samples using that variable and evaluate its output. 87
4.2 The difference between a typical VQA graphical model (in Fig. 5.2a), our corresponding causal model (in Fig. 5.2b) and an example of intervention in the question representation of this model (in Fig. 4.2c). In our model two exogenous variables $\mathbf{u}^q$ and $\mathbf{u}^v$ are incorporated to learn and reason about the intervention caused by these variables. . . 92
4.3 Counterfactual examples that can be generated for a given image. The fusion function h is used with the observational data as well as the counterfactual data to predict the answer. For the counterfactual loss, we need to consider the relationship between the predicted counterfactual answer and its observational counterpart. 94
4.4 Training metrics in CIFAR experiments. 102
4.5 CIFAR-10 results 103
4.6 CIFAR results 104
4.7 The performance of our approach vs. the baseline using fraction of the training data. 106
4.8 Given the image-question pair in the first column, the closest instances of the questions (in second column) and images (in the third column) are found from the VQA v2 test dataset corresponding to the generated counterfactuals (using the exogenous variables). 106

5.1 We seek to improve a VLN agent's capability to generalise to unseen environments at test time. Agents are typically trained by reinforcement and imitation learning, using ground-truth pairs of instructions/trajectories ("factual observations", left). We propose to generate alternative, *counterfactual* training observations with combinations of existing environments. We determine the minimum intervention on the factual data that causes the current model to produce different outputs. We then formulate our novel training objective to best exploit these additional examples and improve its generalisation capabilities (right). The generation process is formalised with a causal model of the data, in which we introduce the interpolation coefficients as an exogenous variable $\mathbf{u}$, effectively modelling an intervention on the environment. 114

5.2 Structural Causal Model (SCM) of the vision-and-language navigation (VLN). We incorporate an exogenous variable in the SCM that is learned and utilised for reasoning about interventions in the observation. 117

5.3 The effect of α on the results inside unseen environments. $\frac{\alpha}{(1-\alpha)} = 0$ means no counterfactual is used (conventional training). 130

5.4 Trajectory Sample. The baseline agent (left trajectory) follows the language instruction until step 4, where instead of moving towards the *massage table*, it goes into the next *hallway*. We argue that since there are limited *massage tables* in the training set, the baseline method does not consider this one as a variant of table and continues searching to stop at the end of the next hallway (6 meters away from the target position). On the other hand, our agent (at the right side), succeeds in identifying the *massage table* and ends up at the target location without any navigation error. 133

5.5 Trajectory Sample. Both agents follow the same path until step 3, where they need to identify and reach the *coffee table* that is surrounded by the *couch* and *armchairs*. In contrast to the baseline model that looks for typical tables in the environment and overlooks the couch from the back view, our model, recognises them and attends the target position successfully. . . . 134

5.6 Trajectory Sample. The baseline model neglects a part of the instruction, and it seems that it has presumed the *picture frame* in step 2 as the one mentioned in the guidance improperly. Biased by the great number of trajectories in the training set, it decides to go into the door at the opposite side, which costs the agent to end up unsuccessfully (7 meters away from the target). On the contrary, our approach executes the instruction precisely, finds the *eye chart picture frame* correctly, and stops at the vicinity of the goal location (1 meter error). 135

5.7 Trajectory Sample. From the pictures it is evident that the baseline approach cannot find the right path, that is identifiable with the *picture* clue in the instruction, and, consequently, ends up about 13 meters away from the target. On the other hand, our approach succeeds in correlating the language instruction to the correct path and reaching the target location (less than 1 meter error). 136

List of Tables

2.1 Value estimation results. Values in the table demonstrate the average divergence of the value estimations from the humans' agreed prices in the test set. 33

2.2 Evaluation metrics for language and pricing evaluation of the models. ↑ indicates higher is better, while ↓ shows lower is better. The model represented by * is our full *Price Negotiator* model and the one indicated by ** is our full *Price Negotiator* when using reinforcement learning. 35

2.3 Human study results. Turing test shows the rates at which dialogues generated from each model has been identified as human generated negotiations. Comparison results reveal the ratio of preferring the dialogues generated from each model over those from the other one. The last three columns demonstrate the average scores provided by humans after interactive negotiations with models. 36

2.4 The table shows the average distance of agreed prices by our **price negotiator** models from their understanding about the value of the item (from OVE). The smaller value indicates the model's ability to insist on achieving its initially estimated value. 36

3.1 A summary of diverse AL settings that we used in our image and non-image experiments. Overall, 30 different settings were utilised in our experiments to compare AL methods in various conditions. "Continue" represents the setting where the weights of the network initialise from those of the network trained in the previous round. Please note that datasets indicated by * are two small subsets of DomainNet-Real that has been used to compare AL methods on small datasets with high-resolution images. 59
3.2 Top-1 test accuracy of various AL methods on HMDB (Kuehne et al. 2011). Values on top of each column reveal the size of the labelled set at the end of each round. 63
3.3 Label acquisition run times of different methods. Our approach is significantly faster than BADGE and about 50x quicker than its Adversarial counterpart. 68

4.1 Accuracy (%) obtained by the testing methods using LSTM (with randomly initialized, trainable embeddings). Best results highlighted in Bold. T abbreviates TreeLSTM (Tai, Socher, and Manning 2015); +P and +C indicate posterior and Counterfactuals respectively. . . . 102
4.2 Test errors for the CIFAR experiments. 103
4.3 State-of-the-art results on VQA-CP test. **UpDn+C** indicates our approach based on UpDn baseline. **(Q+I)** denotes both question and images are intervened. 105
4.4 Performance of our approach on VQA v2 validation. **Pythia+C** is our counterfactual implementation of (Singh et al. 2019). 105

4.5 Evaluation metrics for EQA navigation. Spawning the agent 10, 30, or 50 steps away from the target location, d_0 shows the distance between these initial locations and the target location, while d_T reveals the distance of the final locations and the target ones by starting from these initial location and using the model for the maximum of 100 steps. Finally, $d_\Delta = d_T - d_0$ measures the overall progress of the agent towards the target. GRU+C is ours. 107

4.6 Evaluation metrics for R2R Navigation. **Navigation Length (NL)**: The distance (in meters) that the agent has travelled during an episode. **Navigation Error (NE)**: The difference (in meters) between the agent's final position and the target location. **Success Rate (SR)**: The percentage at which the agent ends up less than 3 meters away from the target location. **Success weighted by Path Length (SPL)**: The success rate weighted by the inverse ratio of the traversed trajectory length to the length of the shortest path.. ↑ shows higher is better, while ↓ means lower is better.. Rows indicated by * our results from Behaviour Cloning and Counterfactuals (BC+C) and Behaviour Cloning, Reinforcement Learning and Counterfactuals (BC+RL+C). 108

5.1 Evaluation metrics for R2R Navigation. Navigation Length (NL) and Navigation Error (NE) values are represented in meters, while Success Rate (SR) values are percentage. ↑ indicates higher is better, while ↓ shows lower is better. Results indicated by * are as reported in the official implementation: `https://github.com/airsplay/R2R-EnvDrop`.128

5.2 The comparison of our results with others in unseen environments. Test-unseen results are reported on the task's leaderboard in single-run setting. The methods indicated by * are taking advantage of self-supervised pre-training. Our method can be applied on top of these methods to result in a further improvement. 129

5.3 Evaluation metrics for EQA navigation. 131

Abstract

Effective and efficient interactions with humans in real environments is an appealing though challenging task for an artificial agent. Despite recent advances in deep learning, especially in the branch of vision and language learning, there are still unsolved issues in the way of reaching such an ambitious agent. Three critical aspects of the interactions between human and machine via natural language (*e.g.* to create intelligent assistants) are: (1) for the model to understand and anticipate human intents to consistently participate in conversations, (2) to learn from a small set of instances and seek information the model needs to accurately achieve its goals and (3) to generalise with those small number of observations obtained under the supervision of humans so that the agent can be practically used. As for human intent perception, we propose an inclusive model for the visual negotiation task, where the intelligent agent needs to anticipate human intent while communicating via natural language. Our model exploits online resources in search of similar items for the estimation of a fair agreement price humans might set as their goals. Considering the estimated agreement price of the advertised item as well as its visual and textual features (*i.e.* images and textual descriptions), we build competitive and consistent language and price generation policies that negotiate significantly better than other baselines. For the information-seeking aspect, we propose an effective active learning (AL) method that facilitates learning with less labelled data by seeking a small subset of unlabelled instances that, when labelled and used for the model training, the highest test accuracy can be achieved. We propose efficient interpolations in the feature space between unlabelled and labelled samples to identify unlabelled instances that have inconsistent class predictions in their neighbourhood. After requesting labels of the selected subset from the human expert, we achieve the highest performance boost in the retrained model in comparison to other AL methods. More specifically, our method achieves remarkable results in the low-data regimes on high-dimensional data, where the performances of other AL methods are unsatisfactory. Finally, regarding the generalisation, we equipped the agent with the capability of reasoning about counterfactual scenarios, which discourages the model's propensity for focusing on spurious features or memorising seen environments. For that, we let the model to intervene in the visual and textual features of the input in a causal model and create counterfactual samples that together with the real observations are used for the training of the model. Hence, the trained model is more resilient to the effect of spurious features and biases in the data and better generalises to unseen situations. Additionally, to increase the generalisation to unseen environments in more interactive applications, we propose a novel approach to generate counterfactual environments and enforce the agent to learn from both the observations and actions in those counterfactual environments. After formalising the supervised and reinforcement learning objectives to include both real and counterfactual environments, our trained agent generalises significantly better than other baselines to unseen environments in two challenging vision-and-language navigation tasks.

Declaration of Authorship

I certify that this work contains no material which has been accepted for the award of any other degree or diploma in my name, in any university or other tertiary institution and, to the best of my knowledge and belief, contains no material previously published or written by another person, except where due reference has been made in the text. In addition, I certify that no part of this work will, in the future, be used in a submission in my name, for any other degree or diploma in any university or other tertiary institution without the prior approval of the University of Adelaide and where applicable, any partner institution responsible for the joint-award of this degree.

I give permission for the digital version of my thesis to be made available on the web, via the University's digital research repository, the Library Search and also through web search engines, unless permission has been granted by the University to restrict access for a period of time.

I acknowledge the support I have received for my research through the provision of an Australian Government Research Training Program Scholarship.

Amin Parvaneh

March 2022

Preface

The main sections of this thesis are based on the following paper in which I am the primary author:

1. Parvaneh, Amin, Ehsan Abbasnejad, Damien Teney, Gholamreza Hafari, Anton van den and Javen Qinfeng Shi (2022). Active Learning by Feature Mixing. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR).

2. Parvaneh, Amin, Ehsan Abbasnejad, Qi Wu, Javen Qinfeng Shi and Anton van den Hengel (2021), "Show, Price and Negotiate: A Negotiator with Online Value Look-Ahead". In: IEEE Transactions on Multimedia.

3. Parvaneh, Amin, Ehsan Abbasnejad, Damien Teney, Javen Qinfeng Shi, and Anton van den Hengel (2020). “Counterfactual Vision-and-Language Navigation: Unravelling the Unseen”. In: Advances in Neural Information Processing Systems. Vol. 33. (*Spotlight presentation*)

Additionally, there is another related paper in which I have been the third author:

- Abbasnejad, Ehsan, Damien Teney, Amin Parvaneh, Javen Shi, and Anton van den Hengel (2020). “Counterfactual Vision and Language Learning”. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). (*Oral presentation*)

My contribution in this paper is 20%, which covers mainly the vision and language navigation (VLN) section. Additionally, I extended the paper to another VLN task (R2R) and included the extended version in this thesis.

Acknowledgements

I would like to thank everyone whose support made my PhD achievable. First and foremost, I like to thank my supervisory panel, Javen Qinfeng Shi and Ehsan Abbasnejad, who supported and guided me during the last couple of years. More specifically, I like to express my gratitude to my main supervisor, Javen Qinfeng Shi, for his support, understanding and technical advice during this long journey. It has been an honour for me to work with him in the last three years. Additionally, I would like to send my special thanks to Ehsan Abbasnejad, an exemplary researcher, who not only helped me to find my general research direction, but he also collaborated with me in all the challenges of my research. I also thank all my collaborators during these years; including but not limited to Damien Teney, Qi Wu, Gholamreza Haffari, and Anton Van den Hengel.

I would also like to thank my family for their support. Especially, I like to give my deepest appreciation to my wife, Ghazal Mirzavandi Moghaddam, for her unconditional and continuous love, support, companionship and encouragement during all the highs and lows of my PhD. Apart from that, I am profoundly grateful to my friends for making my life in Adelaide more fun during the past years.

Finally, I would like to thank the funding support from the school of computer science and the Australian Institute for Machine Learning (AIML).

Chapter 1

Introduction

1.1 Motivations

The emergence of Deep Learning has embarked the resurgence of hope for having an artificial agent able to communicate effectively and efficiently with humans in interactive environments. Since an agent needs the visual and linguistic capabilities to interact with humans in the real world, the study of interactive artificial intelligence lies in the intersection of computer vision, natural language processing (NLP) and interactive learning. Recent advances in deep neural networks have enabled a machine to *see* the visual inputs in the form of images (Zhao et al. 2019; Minaee et al. 2021), videos (Yao et al. 2020) and point clouds (Guo et al. 2021); *understand and generate natural language* (Devlin et al. 2018; Radford et al. 2019); and *interact* with either environments or humans (Andrychowicz et al. 2017; Fang et al. 2019; Kiran et al. 2021; Arora and Doshi 2021). Figure 1.1 demonstrates some of the prominent applications introduced in each of the aforementioned areas and provides examples of tasks defined in their intersections.

Combining vision and language in a single but complex task, researchers have taken initial steps towards building an agent capable of interacting with humans via natural language while utilising visual inputs. With the help of advanced attention mechanisms (Chaudhari et al. 2021), language grounding approaches (Anderson et al. 2018b; Huang et al. 2019), and transformer-based joint pre-training of vision and language models (Lu et al. 2019), magnificent improvements have been achieved

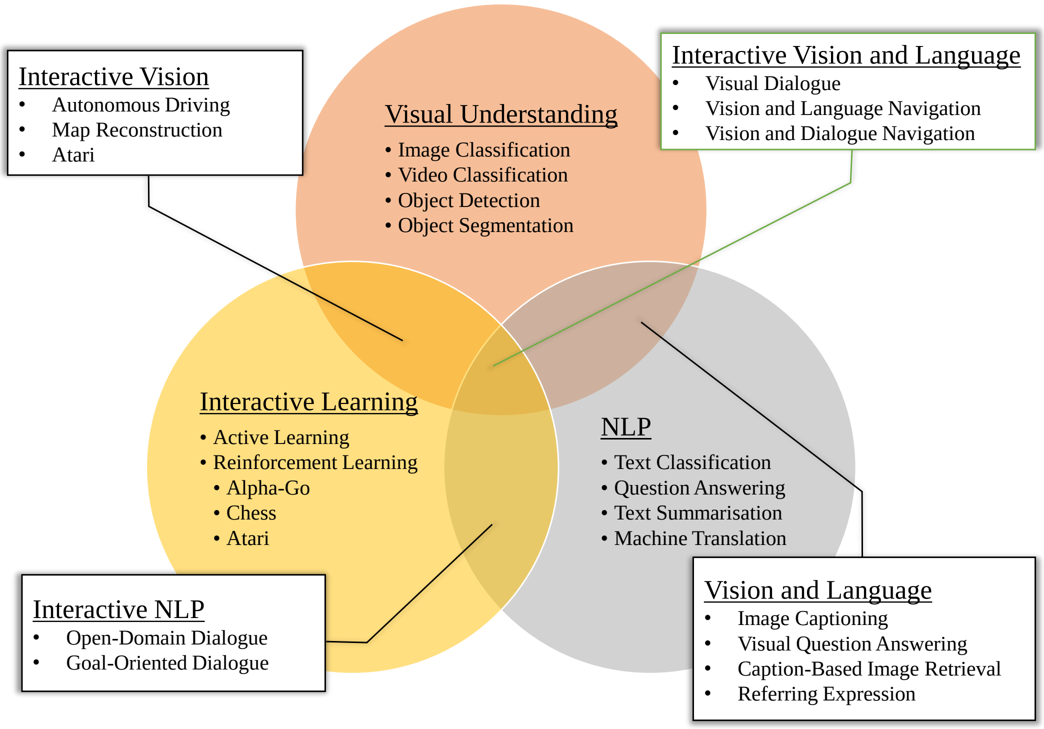


FIGURE 1.1: Interactive agents and the corresponding research areas and applications. The main focus of this thesis is interactive vision and language learning which lies in the intersection of all the three areas.

in some of the critical vision and language learning problems like image captioning (You et al. 2016), visual question answering (VQA) (Wu et al. 2017a), referring expressions (Kazemzadeh et al. 2014), and caption-based image retrieval (Hu et al. 2016). However, most of these advancements have been occurred in static environments or during single-round interactions between humans and machines. Promising progresses in learning a policy that enables an agent to perform in an interactive environment Kiran et al. 2021, especially in deep reinforcement learning (RL) Mnih et al. 2015; Kiran et al. 2021, have given the hope that the combination of vision and language approaches with the interactive learning methods can lead to agents that communicate with humans in interactive environments.

Inspired by animal behaviours (Thorndike and Bruce 2017), RL is a learning methodology (different from supervised and unsupervised learning) that finds an optimised control policy for mapping various situations in an environment to actions in a way that a reward signal is maximised (Sutton and Barto 2018). The introduction of deep neural networks to RL has facilitated the learning of policies for making sequential decisions in environments with high-dimensional input signals (e.g. Atari) at performance levels close to human ones (Mnih et al. 2015; Kiran et al. 2021). More specifically, deep learning has eased the optimisation of complex action-value functions, policy functions, value functions and world models based on high-dimensional inputs in various model-based RL approaches (Ha and Schmidhuber 2018; Kaiser et al. 2020) and model-free ones (Mnih et al. 2015; Andrychowicz et al. 2017; Mnih et al. 2016a; Schulman et al. 2017).

Despite endeavours towards developing interactive vision and language models using RL in new tasks such as visual dialogue systems (Vries et al. 2017; Das et al. 2017a; Kottur et al. 2019), vision-and-language navigation (Anderson et al. 2018c; Savva, Manolis et al. 2019; Hong et al. 2021), and vision-and-dialogue-navigation (Thomason et al. 2020; Nguyen and Daumé III 2019), there are still some critical challenges that have not been addressed well in the literature. Specifically, current methods suffer from inconsistencies in their predicted actions during a conversation with humans due to the misunderstanding of human intents, lack of efficient algorithms for extracting useful information from the human counterpart, and considerable drop in their performance when confronting unseen situations or performing in new environments. Figure. 1.2 demonstrates these challenges in an example where a conversational navigation agent has to perceive human behaviours and in some cases predict their underlying intentions to execute their commands consistently in an in-house environment. Additionally, the agent can efficiently ask its human partner for feedback and more information for the sake of better future performance (*e.g.* requesting desired labels for a small informative subset of collected samples). Moreover, not only should this agent be robust to the visual changes in the given house (*e.g.* the colour of the refrigerator or the decoration of the living room), but it also should perform well in unseen houses. In this thesis, by focusing on the aforementioned challenges we take some steps towards training ideal interactive agents that:

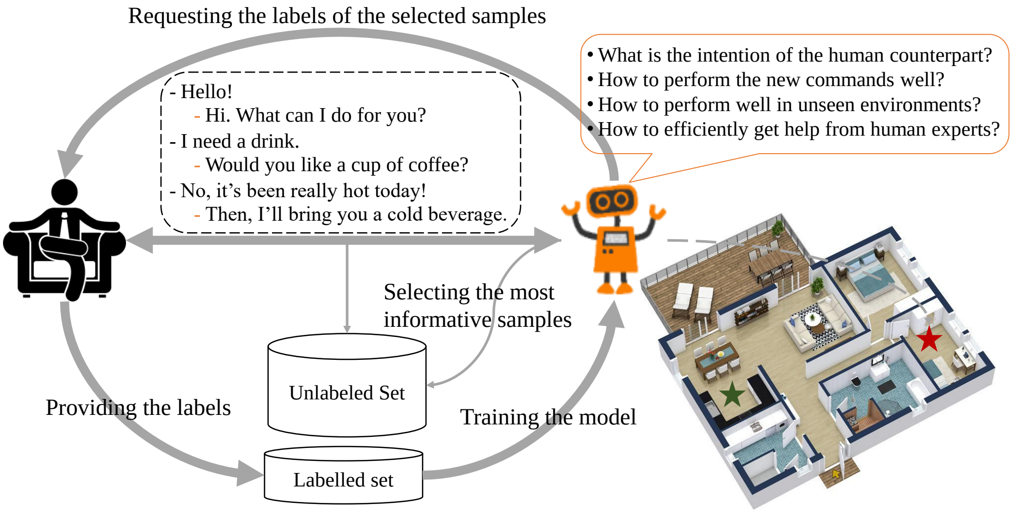

Requesting the labels of the selected samples
- Hello!
- Hi. What can I do for you?
- I need a drink.
- Would you like a cup of coffee?
- No, it's been really hot today!
- Then, I'll bring you a cold beverage.
• What is the intention of the human counterpart?
• How to perform the new commands well?
• How to perform well in unseen environments?
• How to efficiently get help from human experts?
Selecting the most informative samples
Unlabeled Set
Providing the labels
Training the model
Labelled set

FIGURE 1.2: An example of an interactive vision and language learning problem and the corresponding challenges that the agent faces.

1. Comprehend and anticipate human intents and perform consistently during a multi-round conversation.
2. Encourage learning with fewer examples by efficiently seeking information that contributes to the agent's performance improvement.
3. Generalise well to unseen visuals and new environments given a limited set of observations.

The rest of this section, describes the mentioned challenges and our proposed solutions to tackle each one.

1.1.1 Human Intent Understanding

To interact *reasonably* and *consistently* with humans, an artificial agent not only must understand human commands as well as the context of multi-round conversations, but it also should infer humans' underlying goals. Comprehending contextual information from various visual and textual sources as well as the history of the dialogue is an essential, though challenging skill that an artificial agent must possess to respond effectively in a conversation with human. While various modular and end-to-end learning frameworks have been suggested for dialogue understanding in the literature (Lewis et al. 2017; He et al. 2018; Xing et al. 2018), most of these approaches focus on textual resources and neglect the importance of visual inputs.

Another controversial aspect of human-machine interactions through the natural language is the apprehension of humans' underlying intents. This is a crucial and challenging task, especially in conversations to achieve a goal, where human intentions could be conveyed ambiguously or, in some cases, kept as secrets. Neglecting or misunderstanding the human intent during a conversation towards a goal leads to inconsistent actions, which is not desirable in practice.

To highlight this aspect of interactive vision and language learning, we focus on visual negotiation, one of the most challenging goal-oriented dialogue systems, where the anticipation of underlying intents of sellers and buyers is challenging as they attempt to conceal their budgets as well as their estimations about a fair agreement price. Additionally, extracting various negotiation features from multiple resources (images, texts, and numerals) is another challenging aspect of negotiations, which its omission hinders the agent's ability of consistent language generation and price suggestion. Generally, to address the mentioned challenges, we propose an effective way for multi-modal context understanding and human intent prediction, which leads to a model able to consistently and effectively take part in negotiations for selling or buying an advertised item. Specifically, we let the agent analyse online resources in search of similar advertised items, which assists the agent to have a clear estimation of the actual value of the item and the final goal of a human negotiator. Additionally, using a hierarchical recurrent encoder, we enable the agent to consider all the visual and textual information, prices, and the dialogue history for both language generation and offer adjustment during the negotiation. Our final negotiation model, trained in supervised and reinforcement learning settings, enjoys a competitive negotiation policy with fluent language generation and consistent price suggestion skills and surpasses all other baseline models in terms of various linguistic and pricing metrics.

1.1.2 Information Seeking

While the prime purpose of an interactive agent is to provide reliable services, the agent could also benefit from their interactions with humans by looking for useful information that, with the supervision of the human counterpart, can improve the agent's performance in achieving their goals. Reinforcement learning tackles the information–seeking aspect of interactive learning without any supervision from humans by encouraging the agent to explore the environment while learning the policy, which is known as the exploration–exploitation dilemma in the literature (Sutton and Barto 2018; Kiran et al. 2021). While exploration strategies in RL, which are in the form of adding noisy actions to the policy rollout during the training, are effective for the generalisation of the model, they may not be applicable to human interactions as asking random questions from humans is not desirable.

In two-way relationships between humans and machines, the agent can efficiently seek information from its human counterpart either by asking about ambiguous and/or controversial aspects of the task or by requesting feedback on good and bad decisions. While the former concept helps the agent improve its performance level by asking optimal questions (Misra et al. 2018), the latter one, which we cover in this thesis, facilitates the efficient improvement of the model using a limited set of examples labelled under the supervision of humans. One major issue regarding the latter concept is that, provided with a large pool of samples gathered during interactions, an agent cannot demand feedback for numerous instances due to the limited time and efforts humans might be willing to spend. Therefore, constructing a strategy with which the agent can select a small subset of informative and valuable samples from a large pool of unknown instances is essential for efficiently benefiting from the limited number of human guides.

Active Learning (AL) is a solution in which instead of passively spending the limited labelling budget, the agent can actively select a small, informative subset from a large pool of unlabelled ones and label them through a repetitive interaction with a human expert (*i.e.* oracle). Originally, deep AL has been proposed as an interactive solution for the data exhaustiveness of the deep neural networks (Ren et al. 2021). However, most AL strategies fail to perform well when applied to high-dimensional data (*e.g.* image and video) in low-data regimes. In this thesis, we provide a novel AL approach based on interventions in the latent representations of unlabelled instances. We propose an efficient interpolation mechanism to mix features of an unlabelled instance with general features of each class (extracted from the labelled set) to discover inconsistencies in label predictions in the vicinity of the unlabelled sample. We select a diverse set of unlabelled instances with observed inconsistencies during the interpolation process. After labelling these selected instances by oracle and adding them to the training set, the highest improvements in the test accuracy can be gained in comparison to other AL approaches. We demonstrate the efficiency and effectiveness of our approach in various AL settings, especially in low-data regimes, on a wide range of classification tasks on high-dimensional image, video and non-visual data.

1.1.3 Generalisation

Generalisation to new situations using a limited set of observations, is an essential though challenging aspect of interactive vision and language learning. Two critical issues that impede the generalisation of these interactive agents to unseen situations are (1) the model's tendency towards extracting some biases or spurious features from the small training set and (2) environmental shifts to which the model is not resilient.

Learning generalisable features from a small and sometimes biased training set is one of the major challenges in the context of vision and language learning. In spite of remarkable recent success in tasks involving data from multiple modalities (*e.g.* visual question answering and image captioning), it has been shown that a part of this success comes from the exploitation of biases and statistical regularities in the data (Johnson et al. 2017; Ramakrishnan, Agrawal, and Lee 2018; Hudson and Manning 2019; Agrawal et al. 2018). Consequently, rather than learning the actual reasoning required for the fulfilment of the task, the model relies on some spurious features or biases in the data that may not generalise well beyond the provided dataset.

In this thesis, by considering the basic causal model of the training data, we propose the usage of *counterfactuals* as a solution for the raised issue. More specifically, we empower the model to find the minimum alteration (*i.e.* intervention) to the inputs from different modalities that could change the outcome. By defining an additional loss for the intervened instances, we enforce the model to perform well on both factual and counterfactual cases. We demonstrate the effectiveness of our simple approach on a wide range of uni-modal and multi-modal vision and language tasks.

Additionally, complying with the variability of the visual signals from the surrounding environment is another challenging requirement that an agent able to interact with humans in the real world should meet. While domain adaptation approaches (Wang and Deng 2018) ease the adaptation of the model to distribution and domain shifts (*e.g.* expecting a model that has been trained with samples in the day to perform well at night or to generalise well to an unseen set of classes), they may not be directly applicable to a practical interactive agent. An ideal interactive agent requires to perform well in new environments without any requirement for further training and tuning. For instance, a domestic agent trained to take the in-house commands from humans not only should be robust to the changes in the furniture arrangements and wall textures in those seen houses, but they also should perform well when they are taken to an unseen house (Figure 1.2). Despite some interesting approaches proposed for the improvement of the robustness to environmental changes in interactive vision and language tasks (Fried et al. 2018; Tan, Yu, and Bansal 2019; Savva, Manolis et al. 2019), there is still a big gap between the model's performance in seen and unseen environments. The complexity and biases in the multi-modal input data and the intricacy of the proposed neural networks increases the model's propensity to either memorise the training samples instead of reasoning about them or focus on some spurious characteristics in the environment instead of generalisable features.

To address the environment generalisation aspect of interactive vision and language learning, we show that enforcing the agent to reason about counterfactual scenarios in interactive environments can play an influential role in learning more generalisable features. Having that, the model's susceptibility to over-fitting to the seen environments or focusing on spurious features decreases significantly. In our approach, in the context of vision-and-language navigation (VLN), we propose an effective method to build counterfactual environments on the fly in a simple mixing process of two training environments. Additionally, we formalise the training objectives (*i.e.* both imitation learning and reinforcement learning) to consider the agent's performance in both real environments and counterfactual ones. We demonstrate the significant improvements gained from our approach on two challenging VLN tasks.

1.2 Main Contributions

The main contributions of this thesis are:

- With regard to the human intent prediction, in a competitive negotiation setting, we introduce an effective *visual negotiator* that by solving various content understanding, human behaviour understanding, and intent anticipation challenges, interactively negotiates for selling or buying an advertised item. For that, firstly, the agent searches through the online shopping resources and finds similar items to be used as a reference for reasoning a fair agreement price. The predicted price resembles the final goal the human counterpart would have in mind and helps the agent to better understand the behaviour of the opponent during the upcoming challenge. Additionally, our method creates a comprehensive representation of the dialogue state that helps the agent to understand the intentions of the opponent during the dialogue and then act reasonably and consistently based on that. In general, not only our approach follows a consistent and competitive pricing strategy, but it also outperforms all the baselines in terms of language quality and agreement price.

- Concerning the information–seeking aspect of an interactive agent, we propose an innovative active learning method based on mixing features of the known and unknown instances to identify novel features in the unlabelled set of samples. We propose a closed-form solution to find optimum interventions in the latent space to efficiently find unlabelled samples with inconsistent predicted labels in their neighbourhood. By employing our AL approach, the agent can significantly reduce the labelling effort needed for the training in an interactive connection with the human expert and increase the model's performance using fewer labelled instances.

- To improve the model's generalisation, we introduce counterfactuals to tackle two major issues in the interactive vision and language learning:

  - To combat the model's tendency to biases and spurious features in the data, we suggest the usage of counterfactuals. For that, we formalise the vision and language task in a causal model and propose a simple way for the interventions in the input modalities. By encouraging the model to reason about both observational and counterfactual instances, we improve the generalisation of the model in a wide range of vision and language applications.

  - We propose an environment augmentation approach based on counterfactual reasoning that helps the agent to generalise well to unseen environments. We enforce the agent to imagine itself in a counterfactual environment, a mixture of two training environments, and learn from the execution of the command in those environments. Our experiments over two challenging vision-and-language navigation tasks reveals the effectiveness of our approach in improving the model's generalisation to unseen environments significantly.

1.3 Thesis Structure

Chapter. 2 focuses on the challenges of the human intent understanding in the context of visual negotiation, one the most challenging tasks in the area of interactive vision and language learning. **Chapter.** 3 presents our new active learning approach that effectively and efficiently reduces the model's demand for enormous labelled data. In **chapter.** 4, we provide our novel approach, based on the employment of counterfactuals, to improve the generalisation of vision and language models and avoid the model's reliance on spurious features. We extend the generalisation problem to **chapter.** 5, where we demonstrate the significance of counterfactuals for the resilience of interactive vision and language agents to the changes in the surrounding environment. We provide a summary of the thesis in **chapter.** 6 and discusses some of the future directions for having an interactive vision and language agent able to efficiently and effectively communicating with humans in real environments.

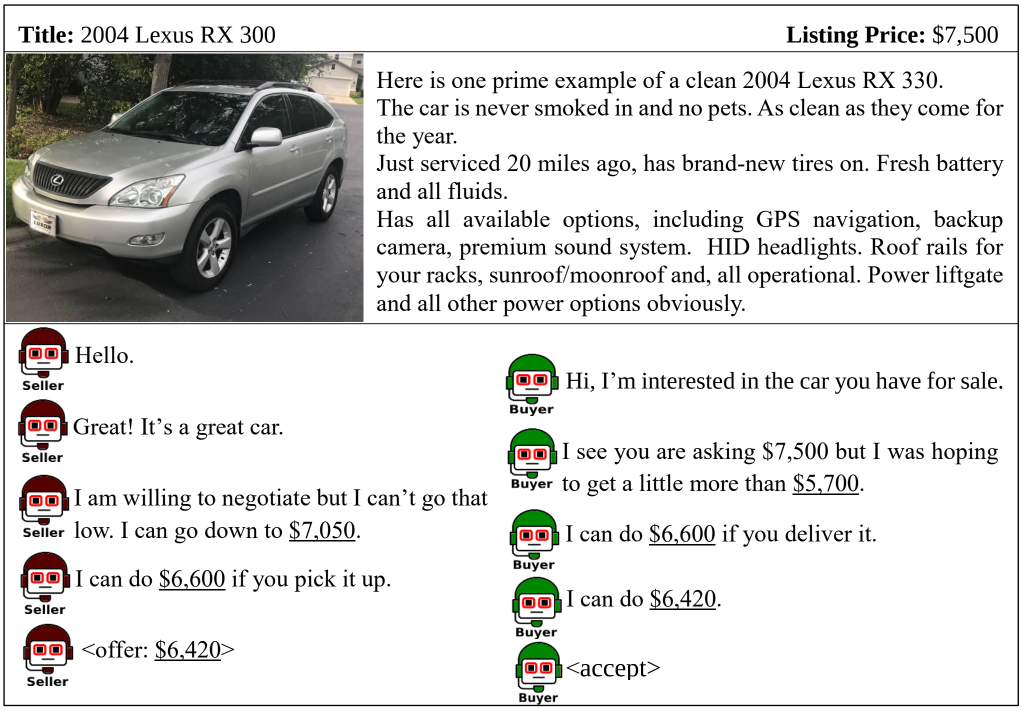
Chapter 2

Show, Price and Negotiate: A Negotiator with Online Value Look-Ahead

In this chapter, we focus on the human intent anticipation challenges of an interactive vision-and-language agent. In the context of visual negotiation, we propose our comprehensive model to handle some issues that have not been addressed well in other baselines.

2.1 Overview

Negotiation, as an essential and complicated aspect of online shopping, is still challenging for an intelligent agent. To that end, we propose the *Price Negotiator*, a modular deep neural network that addresses the unsolved problems in recent studies by (1) considering images of the items as a crucial, though neglected, source of information in a negotiation, (2) heuristically finding the most similar items from an external online source to predict the potential value and an acceptable agreement price, (3) predicting a general price-based “action" at each turn which is fed into the language generator to output the supporting natural language, and (4) adjusting the prices based on the predicted actions. Empirically, we show that our model, that is trained in both supervised and reinforcement learning setting, significantly improves negotiation on the CraigslistBargain dataset, in terms of the agreement price, price consistency, and dialogue quality.



Title: 2004 Lexus RX 300 **Listing Price:** $7,500

Here is one prime example of a clean 2004 Lexus RX 330.
The car is never smoked in and no pets. As clean as they come for the year.
Just serviced 20 miles ago, has brand-new tires on. Fresh battery and all fluids.
Has all available options, including GPS navigation, backup camera, premium sound system. HID headlights. Roof rails for your racks, sunroof/moonroof and, all operational. Power liftgate and all other power options obviously.

Seller Hello.

Buyer Hi, I'm interested in the car you have for sale.

Seller Great! It's a great car.

Buyer I see you are asking $7,500 but I was hoping to get a little more than $5,700.

Seller I am willing to negotiate but I can't go that low. I can go down to $7,050.

Buyer I can do $6,600 if you deliver it.

Seller I can do $6,600 if you pick it up.

Buyer I can do $6,420.

Seller <offer: $6,420>

Buyer <accept>

FIGURE 2.1: An example scenario of our *Price Negotiator* from supervised and RL training. Two agents play the role of sellers and buyers in a visually-grounded bargaining game over an item. The agents need to uncover the underlying value of the item and the attributes of their counterpart (e.g. their assertiveness) to succeed.

2.2 Introduction

Negotiation is an integral part of human interactions. It is a complex task that requires reasoning about the attitudes of the counterpart, mutual interests, and uttering convincing arguments and potentially appealing to sympathy. The prevalence of online shopping provides a test-bed for negotiation ability of artificial agents as human's advocate for the best deals. This artificial agent has to assess the photos of the advertised item, understand the textual content, estimate its true value compared to the others in the market, and conduct a dialogue with its counterpart to reach an agreement.

Recently, Lewis et al. 2017 pioneered negotiation as a specific form of dialogue systems in a DealOrNoDeal game where two artificial agents negotiate splitting of three items. Subsequently, He et al. 2018 used real human dialogues on Craigslist advertisements to learn a dialogue model of negotiations. In both cases, in par with other dialogue systems, various sequence-to-sequence (Seq-Seq) encoder-decoders are utilised to model negotiations. Seq-Seq models or more complex alternatives (Devlin et al. 2018; Luong, Pham, and Manning 2015) for that matter are effective tools for learning the correlation between words (e.g. co-occurrences) and potentially the goal. However, negotiation presents a unique set of challenges beyond word correlation that distinguishes it to that of the conventional dialogue systems. Subsequently, these methods struggle to attain some indispensable aspects of a negotiation including: (1) extracting and utilising information from multiple sources (e.g. photos, texts, and numerals), (2) predicting a suitable price for the products to reach the best possible agreement, (3) expressing the intention conditioned on the price in natural language, and (4) offering consistent prices.

In this chapter, we propose a *price negotiator* to address the aforementioned problems. Our negotiator, inspired by the modular needs of a negotiating agent, comprised of five main units particularly tailored for shopping: (1) *online value estimator (OVE)*, (2) *hierarchical recurrent negotiation encoder (HRNE)*, (3) *action predictor* controller, (4) *price adjuster* and (5) *language decoder* (see Figure 2.2 for details). For *OVE*, motivated by human behaviour, before starting negotiation we find similar items in online stores–simulating market evaluation. This is done by learning an embedding for the *textual* (title and description) and *visual* content of the listings and using a matching network to choose the most similar ones to the current item in the negotiation. Hence, an estimate of how much the item valued is prognosticated that allows the agent to uncover how demanding an item is and whether it's worth the listing price.

Subsequently, in *HRNE* the counterpart's dialogue is encoded conditioned on the content of the advertisement and the agent's belief of its value. This is a significant and distinguishing aspect of our approach since OVE and HRNE effectively disentangle the value of an item from the language model. The output of this step is a dialogue state representation (encoding a combination of dialogue history representation, last prices proposed by the agents, textual and visual inputs and the estimated

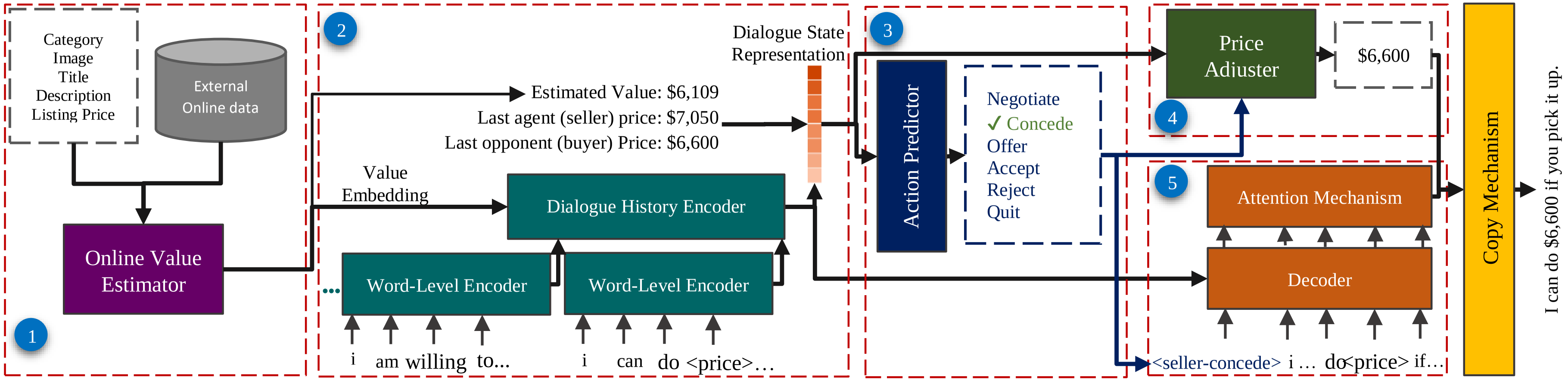


FIGURE 2.2: The diagram of our *price negotiator* that consists of five main modules: (1) online value estimator (OVE), (2) hierarchical recurrent negotiation encoder (HRNE), (3) action predictor, (4) price adjuster, and (5) language decoder.

value) from which *action predictor* decides on the next step for the negotiation. In a nutshell, action predictor decides on continuing with the intention of convincing the counterpart, conceding, offering a price, accepting their terms or quitting. If the decision is to change the offer, then our *price adjuster* proposes a new price. From the state representation and the predicted action, our *language decoder* generates the appropriate language to convey the intentions of the agent. In any case, we use *copy mechanism* (Luong et al. 2015) to combine the new offered price to that of the appropriate negotiating words to utter.

We evaluate our proposed model on CraigslistBargain (He et al. 2018) which provides human-generated negotiations in various scenarios using Craigslist advertisements. Our experiments show that not only the language quality of the generated utterances from our approach outperforms the baselines, the prices are consistent and the agreed price is more similar to that of humans. Moreover, we show *reinforcement learning* (Williams 1992)–that has become increasingly popular with dialogue systems–also improves our model's performance. We also run several human studies to evaluate our negotiator.

In summary, our main contributions are as follows:

1. We propose a novel AI agent that performs negotiation for the best price for either a seller or a buyer. It utilises both visual and textual content for decision making, follows a consistent and human-like pricing strategy and, as our experiments show, outperforms the baselines on both language quality and agreement price.
2. Our negotiator, unlike its counterparts, is able to find the relevant online items to accurately predict its potential agreement price. This enables scalable and commercially viable applications and reduces human bias and inconsistency.

2.3 Related Work

2.3.1 Goal-Oriented Dialogue

Goal-oriented dialogue systems have a long history in natural language processing (NLP). Recently, researchers suggested to define a goal in open-domain dialogues to improve the consistency and engagement of the agent (Tang et al. 2019). Additionally, multi-modal dialogue systems have gained strong interests in speech recognition (Potamianos et al. 2007) and computer vision communities (Liao et al. 2018; Saha, Khapra, and Sankaranarayanan 2018). Specifically, visual goal-oriented dialogue systems have got the popularity by introducing miscellaneous tasks including "Guess-What?!"(Vries et al. 2017; Lee et al. 2019), "Image Guessing" (Das et al. 2017b), "MNIST Counting Dialogue" (Lee, Heo, and Zhang 2018), "Visual Dialogue" (Das et al. 2017a) and "CLEVR-Dialogue" (Kottur et al. 2019). However, since the machine can play just one role (either questioner or answerer) in most applications, they are Visual Question Answering problems by nature rather than two-way, interactive dialogue systems (Wu et al. 2017b; Das et al. 2017a; Kottur et al. 2019). In this chapter, we focus on *Visual Negotiation* where the model is evaluated interactively in negotiations either with humans or with another model.

Generally, dialogue systems can be categorised into *collaborative* and *competitive* systems. In a collaborative dialogue environment, agents can help each other to reach a common goal. Applications include trip and accommodation reservation (El Asri et al. 2017; Wei et al. 2018), information seeking (Reddy, Chen, and Manning 2018; Abbasnejad et al. 2019; Ammicht, Fosler-Lussier, and Potamianos 2007), mutual friend searching (He et al. 2017), navigation (Dušek and Jurcicek 2016; Vries et al. 2018), fashion product recommendation (Liao et al. 2018), disease diagnosis (Xu et al. 2019), addressee detection (Tsai, Stolcke, and Slaney 2015), emotion detection (Majumder et al. 2018), and even donation persuasion (Wang et al. 2019b). In contrast, in a competitive dialogue environment, agents must negotiate to achieve an agreement based on their individual goals. Their goals are often opposite to each other. “Settlers of Catan” (Cuayáhuitl, Keizer, and Lemon 2015) and DealOrNoDeal (Lewis et al. 2017) are two frontier tasks defined as competitive dialogues. Very recently, a new negotiation dataset is introduced by crawling tangible negotiation scenarios from the Craigslist website and collecting seller-buyer dialogues for each scenario (He et al. 2018). Although our work is built on top of the same dataset, there are significant differences: (1) *we propose to use the photos of the item as an important source of knowledge which was neglected in (He et al. 2018)*; (2) *before the negotiation, we prognosticate an ideal agreement price by analysing other similar items on online stores*; (3) *we aim to estimate and refine the price in a consistent manner, and produce human-like dialogues.*

2.3.2 Dialogue Systems Design

Goal-oriented dialogue systems can be designed in a component-based fashion or end-to-end. In a component-based fashion, it typically has three separate modules: (1) natural language understanding (NLU) unit that maps an utterance into semantic slots to be understood and processed by the machine, (2) dialogue manager (DM) which selects the best action according to the output of NLU, and (3) natural language generator (NLG) which produces a meaningful response based on the action chosen by DM, either by looking at a set of possible responses for that action or by using a statistical machine learning language model (Chen et al. 2017; Zue and Glass 2000).

To overcome the complexity and bypass the reliance on human-crafted information retrieval rules in component-based approaches, end-to-end systems have been proposed in recent years (Wen et al. 2017; Bordes, Boureau, and Weston 2017; Dhingra et al. 2017; Li et al. 2017b; Li et al. 2017a; Sordoni et al. 2015; Xing et al. 2018; Dušek and Jurcicek 2016). These systems often use an Seq-Seq architecture consisting of an encoder which receives the previous utterance(s) and encode them into a latent representation based on which the decoder can predict and generate the next utterance. In the end-to-end model proposed by He et al. 2018 for the negotiation, prices are embedded similar to other words in the utterance. Since the range of the prices are broad and there is not any pre-trained embedding for them, their embedding is learned through the model training. In addition, the generated prices are inconsistent since they were produced based on the correlation with other words rather than the true underlying value of the item. Furthermore, this way of embedding the prices adds more complexity to the model and leads to weaker language model. In this research, we show that eliminating the prices from the dictionary of the model, can help the language model to generate better dialogues. We propose an end-to-end modular approach in which we predict the price and the supportive language separately from different heads of the network.

2.4 Price Negotiator

2.4.1 Problem Definition

The problem we consider is that of having two agents, namely a seller and a buyer, negotiating on the price of an item which is identified by an image, textual title and description. The items are classified into various categories as is the common practice in the online shopping websites. The seller advertises an item with a *listing* price and most likely agrees to offers closest to this value. The buyer on the other hand has a *target* price which is lower than the seller's listing. While the buyers know the listing price, their target price is not revealed to the seller. It should be noted that a negotiation may end without an agreement.

Each negotiation scenario consists of an advertised item i by providing its context information $\boldsymbol{C}_i = \{\boldsymbol{v}_i, k_i, \boldsymbol{x}_i, \boldsymbol{c}_i, p_{0,i}\}$, where $\boldsymbol{v}_i$ represents its visual cue/feature (i.e. photo), k_i is the category in which the item has been advertised, $\boldsymbol{x}_i$ is the title of the advertisement, $\boldsymbol{c}_i$ is the description provided for the item, and $p_{0,i}$ is the listing price suggested by the seller. Additionally, at each dialogue turn t, a sequence of utterances in previous turns is available as the dialogue history $\boldsymbol{z}_{i,t} = \{\boldsymbol{u}_{i,0}, \boldsymbol{u}_{i,1}, \ldots, \boldsymbol{u}_{i,t-1}\}$. It is noticeable that each utterance is a sequence of words

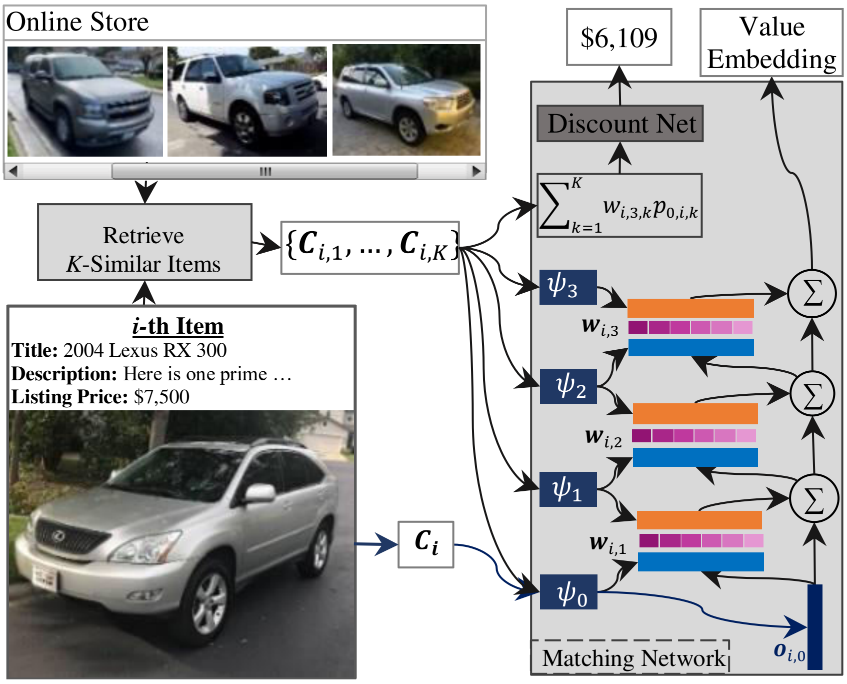


FIGURE 2.3: The architecture of *Online Value Estimator* that finds K-similar items in an online store. Comparing their multi-modal contextual features $\{\boldsymbol{C}_{i,1},\ldots,\boldsymbol{C}_{i,K}\}$ with those of the given item $\boldsymbol{C}_i$, it prognosticates a fair agreement price to be considered in the negotiation. See Figure 2.4 for more details about multi-modal embeddings $\Psi_0,\ldots,\Psi_3$.

(tokens) $\boldsymbol{u}_{i,t} = \{\boldsymbol{\omega}_{i,t,0}, \boldsymbol{\omega}_{i,t,1}, \ldots, \boldsymbol{\omega}_{i,t,L-1}\}$, where L represents the maximum length of each utterance, and each word is represented as a d-dimensional vector.

At t-th round of negotiation, the agent generates the j-th token conditioned on the context information $\boldsymbol{C}_i$, the dialogue history $\boldsymbol{z}_{i,t}$, and the previously generated tokens $\{\boldsymbol{\omega}_{i,t,0}, \boldsymbol{\omega}_{i,t,1}, \ldots, \boldsymbol{\omega}_{i,t,j-1}\}$. The objective is to as closely as possible mimic the behaviour of a human in negotiation. Consequently, the prices agreed upon by an agent has to be as similar as possible to that of the human using convincing arguments.

2.4.2 Online Value Estimator

One of the essential skills in negotiation is to have a good estimation of the real value of the item. Humans usually search through different shopping websites to find similar items and compare their attributes and listing prices with those of the given item. Motivated by this, we designed the *online value estimator* (OVE), a deep neural network that can make a precise value prediction (Figure 2.3).

Given the context information $\boldsymbol{C}_i$ of the item i, the OVE component predicts a scalar value for the agreement price. This estimation is based on both visual features of the item, extracted from its photo $\boldsymbol{v}_i$; its textual features extracted from its category k_i, title $\boldsymbol{x}_i$ and description $\boldsymbol{c}_i$; and its listing price $p_{0,i}$. Generally, the OVE component aims at minimising the difference between the predicted price $\bar{p}_i$ and the ground-truth real agreed price p_i. The ground-truth real price is calculated as the average of all agreed prices in human-human negotiations over the given item in the dataset. To predict a price we learn a deep neural network Φ parameterised by $\boldsymbol{\alpha}$ through the minimisation of the following loss:

$$\boldsymbol{\ell}_{ove} = \sum_{i=1}^{N} |\Phi(\boldsymbol{v}_i, k_i, \boldsymbol{x}_i, \boldsymbol{c}_i, p_{0,i}; \boldsymbol{\alpha}) - p_i|, \tag{2.1}$$

where N represents the number of items in the training set.

The price is predicted in a three-stage process. First, the extracted features of the items are used to find the K-similar items from an online source of advertised items. The similarity between two items is defined as a combination of cosine similarities between their visual and textual features and the normalised abstract similarity between their listing prices.

Second, the *matching network*, a deep neural network with a structure akin to memory networks (Sukhbaatar et al. 2015), takes these items and measures their importance in valuing the item. It worth mentioning that in contrast to (Sukhbaatar et al. 2015), where they only embed text inputs, we propose *multimodal embedding* (Figure 2.4) that embeds the visual and textual features of the given item i into a d-dimensional representation $\boldsymbol{e}_i$. Specifically, our proposed matching network consists of 3 attention layers and 4 multimodal embeddings. At each layer l the the correlation between the previous representation of the given item $\boldsymbol{o}_{i,l-1}$ and multimodal embeddings of related K similar items $\boldsymbol{m}_{i,l} = [\boldsymbol{e}_{i,l,1}, \ldots, \boldsymbol{e}_{i,l,K}]$, which

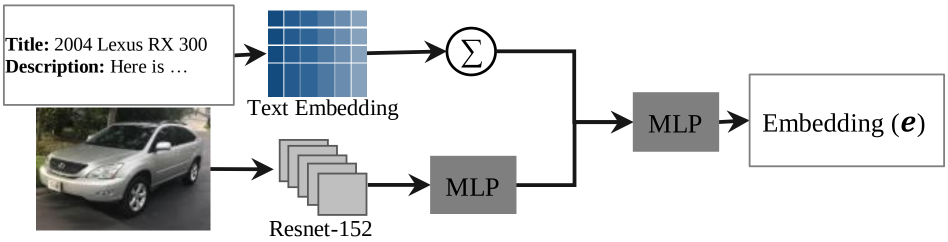

Title: 2004 Lexus RX 300
Description: Here is …
Text Embedding
$\sum$
Resnet-152
MLP
MLP
Embedding ($\boldsymbol{e}$)

FIGURE 2.4: Multimodal Embedding. The sum of embeddings of the textual resources (title and description) is concatenated with the down-sized visual features obtained from a pre-trained network (ResNet-152 (He et al. 2016)). The concatenated vector is passed through a 2-layer MLP with ReLU activation to create the multimodal embedding.

are extracted from $(l-1)$-th multimodal embedding Ψ_{l-1}, is calculated as follows:

$$\boldsymbol{w}_{i,l} = \text{Softmax}(\boldsymbol{o}_{i,l-1}^{\intercal}\boldsymbol{m}_{i,l}). \tag{2.2}$$

Afterwards, the output of the layer (the item representation $\boldsymbol{o}_{i,l}$), is calculated based on the following equation:

$$\boldsymbol{o}_{i,l} = \boldsymbol{o}_{i,l-1} + \sum_{k=1}^{K}(w_{i,l,k}\boldsymbol{e}_{i,l,k}). \tag{2.3}$$

Please note that the initial item representation $\boldsymbol{o}_{i,0}$ also comes from the first multimodal embedding.

Finally, the correlation weights from the last layer of the network are multiplied by the listing prices of the corresponding similar items to achieve an estimated value $\bar{p}_i$ for the given item. It worth to mention that since this value is calculated from the listing prices and the target is the agreement price, we pass the output through another fully connected layer, which we name it *discount net*, to estimate the final value for the item.

2.4.3 Hierarchical Recurrent Negotiation Encoder

One of the problems in conventional negotiation models is that they include price values (real numbers) in the vocabulary and treat them like ordinary words in the dialogue. This deters the intelligent agent from understanding the numerical meaning of the prices, and entangles the strategies for generating words and prices together. As a result, the prices generated in the dialogue, especially at final offering turn, are inconsistent in most cases.

In our price negotiator we devise a novel *hierarchical recurrent encoder* in which the prices in the utterances are replaced with a fix token (`<price>`) to be later replaced with the generated ones. In a hierarchical structure (Sordoni et al. 2015; Xing et al. 2018; Dušek and Jurcicek 2016), our model encodes utterances in two levels: a *word-level encoder* that is an RNN network ($f^{we}: \mathbb{R}^{L \times d} \rightarrow \mathbb{R}^d$) mapping the word embedding of t-th utterance (a sequence of maximum L words) into a d-dimensional vector ($\boldsymbol{h}^{we}_{i,t}$) as the word-level representation of the utterance; and a *dialogue history encoder* that is another RNN network ($f^{he}: \mathbb{R}^{(t-1) \times d} \rightarrow \mathbb{R}^d$) which at each turn t receives word-level representation of the previous utterances as the input and maps them to a d-dimensional vector ($\boldsymbol{h}^{he}_{i,t}$). Since this representation should be conditioned on the value estimation resulted from OVE, we feed the output of the last layer of the matching network into this RNN as the initial hidden state.

Apart from the dialogue history representation, the last prices suggested by the agent $p^a_{i,t-1}$ and the opponent $p^o_{i,t-1}$ and the estimated price $\bar{p}_i$ are embedded into a vector which represents the dialogue state $\boldsymbol{s}_{i,t}$ (more details in section 2.5.2). This vector will then be used by other components to decide about action and prices that should be considered.

2.4.4 Action Predictor

The action predictor module is a multi-layer perceptron (MLP) that predicts the next action $a_{i,t}$ should be taken by the agent according to the dialogue state $\boldsymbol{s}_{i,t}$ at round t of the negotiation. In contrast to (He et al. 2018) who tried to predict coarse intents based on intent encoding, we suggested to predict extremely simpler actions, which are based on the price. Actions defined in our framework are:

- *Negotiate* tells the agent that it should continue the negotiation without changing the price.

- *Concede* determines that the agent should make a concession on its previously proposed price. In other words, the buyer should increase its suggested price and the seller needs to decrease its asking price when this action is predicted.
- *Offer* suggests that the agent should propose a final offer and wait for the response from its counterpart.
- *Accept* means that the agent should accept the official offer suggested by the opponent and terminate the negotiation successfully.
- *Reject* clarifies that the agent should reject the proposed offer.
- *Quit* means the agent should abandon the negotiation.

In the supervised training setting, this neural network learns parameters $\boldsymbol{\delta}$ that better imitate human-like actions by minimising this loss function:

$$\boldsymbol{\ell}_{ap} = \sum_{i=1}^{N} \sum_{t=1}^{T_i} - \log p(a_{i,t} | \boldsymbol{s}_{i,t}; \boldsymbol{\delta}), \tag{2.4}$$

where N and T_i represent the number of training dialogues and the number of agent's turns in each dialogue respectively.

2.4.5 Price Adjuster

Proposing a reasonable price at each stage of the dialogue is fundamental for a negotiation agent. Our price adjuster module can make consistent price suggestions that lead the agent to reach the best possible agreement. This module is invoked only if the action predictor decides to concede or make an offer. In either case, the price adjuster, an MLP with parameters $\boldsymbol{\phi}$, predicts the ratio $r_{i,t}$ from which the agent should concede. This prediction is based on the current state of the dialogue $\boldsymbol{s}_{i,t}$ and the action predicted by the action predictor $\bar{a}_{i,t}$. We discretise the price change ratio into six categories (more details in section 2.5.4) and optimise the network using this loss function:

$$\ell_{pa} = \sum_{i=1}^{N} \sum_{t=1}^{T_i} -\log p(r_{i,t} | \boldsymbol{s}_{i,t}, \bar{a}_{i,t}; \boldsymbol{\phi}), \tag{2.5}$$

2.4.6 Language Decoder

Language decoder is an RNN that generates a sequence of words as the next utterance based on current dialogue state $\boldsymbol{s}_{i,t}$ and the predicted action $\bar{a}_{i,t}$. To that end, we initialise its hidden state with the last hidden state from the dialogue history encoder. Additionally, we condition the starting token on the selected action, by defining different tokens for different actions. We then train an agent able to play both seller and buyer roles by defining different start tokens for the buyer and the seller.

In order to encourage the output to pay more attention to the most important parts of various available information sources, a global attention mechanism (Luong, Pham, and Manning 2015; Gu et al. 2016) is applied to the outputs of the language decoder. This helps the system to ask or answer questions for different sources including the title, description and the outputs of word-level encoder for previous utterance.

To map the outputs of the model to a probability vector of our vocabulary size, a linear function (generative layer) and a LogSoftmax is applied to the output of the model. With language decoder we find the parameters $\boldsymbol{\theta}$ of the RNN to maximise the likelihood of each word,

$$\ell_{ld} = \sum_{i=1}^{N} \sum_{t=1}^{T_i} \sum_{j=1}^{L} -\log p(\boldsymbol{\omega}_{i,t,j} | \boldsymbol{C}_i, \boldsymbol{z}_t, \boldsymbol{\omega}_{i,t,0}, \ldots, \boldsymbol{\omega}_{i,t,j-1}; \boldsymbol{\theta}). \tag{2.6}$$

2.4.7 Copy Mechanism

We disentangle prices from other words during the encoding and decoding by replacing prices in utterances with a fix token (`<price>`). While we encode the current proposed prices separately, the decoder only predicts the price location in the generated utterance. Similar to *copy mechanism* utilised in machine translation and question

2009 Motobecane FLY Pro Hardtail MTB	Seq-Seq	Visual Negotiator
Listing Price **$800** Aluminum; lightweight bike, probably ~24 pounds. 16" frame or so; I'm 5'8" and just a bit too big, so probably best for someone in the 5'4"-5'7" range. Shimano Deore XT crankset Vuelta XRP Pro super lite wheels Ritchey Comp handlebars and stem Rock Shox racing forks	*seller*: hey there! *buyer*: hi, i am interested in your bike. how old is it? *seller:* it is in excellent condition. it is in great condition. *buyer:* i see. is it in good condition? *seller*: yes, it is in good condition. *buyer*: i see. i see it is in good condition. i would like to offer **$608**. *seller*: i 'm sorry, but i can't go that low. i can go down to **$639**. *buyer*: i can do that. *seller*: <offer: **$720**> *buyer*: <accept>	*seller*: hello, are you interested in the bike? *buyer:* yes i am. how old is it? *seller*: it's in great condition. i 'm asking **$800** for it. *buyer*: i see. i 'm interested in the bike. i 'm looking for a bike **for my son to birthday**. i was wondering if you could negotiate the price a bit? *seller*: i can do **$752**. *buyer*: i can do **$646** **if you can deliver it to me**. *seller*: i can do **$704**. *buyer:* i can do **$684** **if you can deliver it**. *seller*: <offer: **$684**> *seller*: <accept>

FIGURE 2.5: Examples from two models. The left-hand side dialogue is generated from a simple Seq-Seq model which makes a mistake in its offer (from $639 that was agreed upon moves to $720) and produces repetitive, dull responses. On the other hand, our *Price Negotiator* model, the right-hand side dialogue, creates a linguistically more diverse and price-wise reasonable dialogue and uses different negotiation strategies to persuade the opponent.

answering (Luong et al. 2015; Gu et al. 2016), we replace the price point predicted by the language decoder module with the value calculated by the price predictor module to create the final output.

2.4.8 Overall Objective

The final objective function for the price negotiator model is to minimise the combination of losses introduced for each component.

$$\boldsymbol{\ell} = \boldsymbol{\ell}_{ove} + \boldsymbol{\ell}_{ap} + \boldsymbol{\ell}_{pa} + \boldsymbol{\ell}_{ld}. \tag{2.7}$$

2.4.9 Reinforcement Learning

We use reinforcement learning to encourage our *Price Negotiator* agent to improve by employing self-play (i.e. two instances of our model play buyer and seller roles and negotiate with each other). Specifically, once supervised training of the network is done, the action predictor and the price adjuster are fine-tuned using REINFORCE algorithm. We assign a role to an agent (say seller) and let it negotiate against another (e.g. buyer) for a given scenario (i.e. image, title, description, listing price). At the end of negotiation, we evaluate the performance by providing a *reward* signal. Our reward signal measures how successful the agent was according to the distance between the agreed price and the estimated price $\bar{p}_i$ predicted by the OVE. The motivation for the reward signal is to intrigue the agent to mimic human's strategy and achieve the same agreement price. Thus, the action predictor network is updated by back propagating the following signal:

$$\sum_{i=1}^{N} \sum_{t=1}^{T_i} \log p(\bar{a}_{i,t} | \boldsymbol{s}_{i,t}; \boldsymbol{\delta}) G_i, \tag{2.8}$$

where G_i represents the total reward for negotiation i. The same update is applied on price adjuster component, which is eliminated for the brevity.

2.5 Implementation Details

2.5.1 Dataset

All the experiments are performed on the CraigslistBargain dataset (He et al. 2018). It contains 4,219 training dialogues, 471 evaluation dialogues and 500 test dialogues which are created based on 891, 103, and 134 different items respectively.

To make the training process of OVE simple and fast, we simulated an online external data source. To that end, we scraped $10,586$ items advertised on Craigslist website and made a local source accessible by our online value estimator. It is notable that we set the number of selected items from the online source k to 32 in all experiments.

2.5.2 Embeddings

In all the experiments, we use 300-dimensional vectors as the embedding for each word from pre-trained GloVe embedding (Pennington, Socher, and Manning 2014). In order to extract the features from the images, we utilised Resnet-152 (He et al. 2016) pre-trained on ImageNet dataset, which has shown exceptional performance in various object detection problems. We simply replaced its last fully-connected layer with another one to produce a 300-dimensional vector representing the image features.

The hierarchical recurrent negotiation encoder maps prices (either the agent price, the opponent price, or OVE estimated price) into a 7-dimensional one-hot vector which will be concatenated with the last hidden state of its dialogue history encoder to represent the dialogue state. To that end, similar to (He et al. 2018), prices are first normalised separately for each agent so that 1 is the agent's target price and 0 is their bottom-line price. Defined in the negotiation scenario, the bottom-line price for the seller is the lowest price he/she is supposed to sell the item while for the buyer it is the highest value they should pay for the item. It worth mentioning that these bottom-line prices are not strict and agents can propose and agree on values outside of this range. After the normalisation, price range between 0 and 1 is segmented equally to 5 parts representing 5 classes and other two classes belong to values lower than 0 and higher than 1. Please note that prices lower than the seller's bottom-line price or higher than buyer's bottom-line price are represented as negative values for them.

2.5.3 Training Settings

All RNNs used as encoder or decoder are 2-layer LSTMs with 300-dimensional hidden states. Action predictor and price decoder networks have the same network architecture, a 4-layer fully-connected network with ReLU activation functions. We also applied a dropout with a rate of 0.3 to all parts of our architecture.

Parameters of the models are optimised using Adam with the learning rate set to 1e-3 in first 20 epochs and then decayed to 1e-4 for another 320 epochs. The batch size is set to 128 in all experiments.

It worth to mention that since the number samples for each class in both action prediction and price decoding tasks are imbalanced, we used weighted Cross-Entropy loss function where $\frac{1}{\sqrt{frequency(C_i)}}$ are calculated for each class C_i and are used as the weights after normalisation.

Except for hierarchical dialogue encoder and the language decoder which are trained together, we trained other modules separately during the supervised learning process. Afterwards, for RL, we only optimised the action predictor and price decoder parameters for 5000 episodes using a learning rate of 1e-4.

2.5.4 Price Adjustment

In the experiments, the price adjuster branch of the negotiation model makes a multi-class decision about the ratio with which the agent should retreat. Specifically, the agent always begins from the listing price if it acts as the seller and 50%, 70%, or 90% of the listing price (depending on the scenario) if it acts as the buyer. Then, at each turn, the price decoder decides how much the current price should be changed.

Price changes are categorised into 6 classes representing the change ratio which is 0%, 20%, 40%, 60%, 80%, or 100% of either the listing price minus the target for the buyer or listing minus the 70% of list for the seller. If the price decoder decides on altering the price, and the generated utterance contains a price token, it decreases (if being a seller) or increases (if being a buyer) the current price by the predicted ratio.

2.5.5 Evaluation Metrics

In this task, a negotiation is successful if the agents reach an agreement (by accepting a final offer from the opponent). However, the successful rate is not a suitable evaluation metric in this case because the 'deal' might be a bad one even it was accepted. Instead, we defined various dialogue evaluation metrics which can be categorised into three groups: (1) metrics that evaluate the language quality (human-likeness) of the generated dialogue, (2) metrics that evaluate the pricing strategy of the model, such as the difference between the machine agreed price and the ground truth price, and (3) human studies.

Language Metrics. In addition to BLEU score that measures how similar are the sequence of words generated in a machine-machine negotiation with those from human dialogues, we introduce Intent-BLEU (IBLEU), a new metric to measure the similarity of the intents taken by a machine with those taken by humans. More specifically, we extract the intents of each dialogue turn in a machine generated dialogue using the information retrieval approach introduced by (He et al. 2018). For instance, the intent of saying "hello." is `intro` and the intent of saying "I can't go that low. I can go down to $7,050." is `counter-price`. Then, the extracted intents are concatenated to create a sequence of intents that will be compared with those extracted from human generated dialogues. This is basically done by calculating the clipped n-gram precision (for a maximum order of 4) of the generated sequence. The higher values of this metric shows the higher level of similarity with human.

Apart from IBLEU, we applied various word-level and sentence-level metrics to show the richness of the generated dialogues. We calculate the number of distinct sentences produced by the model and scale them by the total number of sentences as another new metric to show the language quality of the dialogue model. We also calculate the same metric at word-level to show the diversity of the lexical used by the model. And shorter dialogue length normally means the agent can make the deal in a shorter time.

Pricing Metrics. Two important metrics that measure the mistake ratio of the pricing are price inconsistency and offer inconsistency. When the seller proposes a price that is higher than the price previously suggested by themselves or is lower than the price offered by the seller, we consider it as an inconsistent pricing (see Figure 2.5). We also calculate the average distance between the agreed prices from the human's agreed prices as another important measure to evaluate the pricing strategy of the model.

Human Studies. Apart from automatic evaluation, we measured the performance of our price negotiator model using human evaluations. We designed three experiments to evaluate the human-likeness, language richness, and pricing quality of the negotiation models based on both third-party and interactive human evaluation.

- *Turing test:* In this study, given a random dialogue generated from a negotiation model, the participant is asked to clarify whether or not the dialogue is generated by humans.

- *Comparative test:* During this test, two dialogues generated based on the same scenario, one from our price negotiator and another from the baseline model, are shown to the contributor. The participants are tasked with choosing the best negotiation according to the language and pricing qualities.

- *Interactive test:* Similar to (He et al. 2018), we put our price negotiator model along with the baseline model online and asked human volunteers to have a negotiation with a randomly chosen agent. At the end, they are asked to assess the quality of the agent in terms of human-likeness, language fluency and coherency, and pricing competency.

Category	Averaging	AVE	O-KNN	OVE
Bike	$475	$422	$55	**$27**
Car	$3,452	$3,887	$547	**$495**
Electronics	$114	$69	$14	**$6**
Furniture	$191	$167	$26	**$20**
Housing	$433	$458	$205	**$129**
Phone	$112	$125	$21	**$20**
Overall	$993	$898	$155	**$123**

TABLE 2.1: Value estimation results. Values in the table demonstrate the average divergence of the value estimations from the humans' agreed prices in the test set.

In our experiments, we gathered 400 Turing test, 400 comparative, and 20 interactive chat evaluations from 20 participants. It is worth noting that the scenarios and generated chats are randomly selected from the test set.

2.6 Results

2.6.1 Value Estimation Experiments

To have a better understanding about the value estimation ability of our model, we calculate the average normalised divergence of the value estimations from humans' agreed prices in the test set, for each product category. Table 2.1 shows the results and several baseline models that are implemented for comparison. The simplest method is to use the average value of each category as the estimated value. We call this approach *averaging*. We also train an attention neural network named *attention value estimator* (AVE) that takes the visual and textual features of the item and outputs the predicted value. The third baseline is the average price of the k-similar items found from external online source and we name it online k-nearest neighbours (O-KNN). Since O-KNN approach is using listing prices and to have a fair comparison with our model which enjoys a discount network, we applied the average discount ratio from the discount network (around 11%) over the estimated values of this approach.

From the Table 2.1, we can see the model proposed for online value estimation (OVE) can prognosticate an accurate agreement price for an item and beats all other baselines significantly. Using external sources and comparing item features with other items is useful, as the divergence of the predicted prices with the real agreement ones drop significantly in O-KNN approach. More importantly, the divergences drop extremely when similar items are matched with the given item using OVE model.

2.6.2 Negotiation Dialogue Evaluation

In order to compare our negotiator with other baseline models, we train three state-of-the-art methods that treat the prices as words. The first two models are trained to match the method proposed in (He et al. 2018) on CraigslistBargain. The first one is a simple sequence-to-sequence model, SL(word), and the second one is a modular approach (SL(act)+rule) which has applied various human-crafted rules to repeat utterances produced by humans. Additionally, a Hierarchical Recurrent Encoder-Decoder (HRED), as a widely-used end-to-end approach for dialogue systems, has been trained as another baseline model. Moreover, we have done several ablation studies to show the importance of each module in our proposed model.

Language Evaluation Results. Table 4.5 demonstrates the fact that price elimination from the language vocabulary improves the language quality. Especially, compared with SL(word) and HRED which are not based on human-crafted rules, dialogues generated from both *price negotiator* models enjoy remarkably more language diversity both in word level and sentence level, as the ratio of repetitive sentences, which has been encountered as a common problem in dialogue generation, has increased significantly in both variations of the proposed framework. Additionally, the dialogue and utterance length of the dialogues generated from these models is large enough to show the richness of the generated dialogues. Although it can be inferred from the results that SL(act)+rule is generating linguistically better dialogues as the sentence and vocabulary diversity of this model is larger than the proposed model, it should be mentioned that this diversity is due to heuristic rules that select templates from the dataset that are different from the previously selected ones.

Model	**Language Metrics**					**Pricing Metrics**		
	IBLEU↑	BLEU↑	Sentence Diversity↑	Vocabulary Diversity↑	Dialogue Length↓	Inconsistency in Pricing↓	Inconsistency in Offering↓	Human Divergence↓
SL(act)+rule (He et al. 2018)	20.21	2.59	**0.498**	**0.0467**	18	1%	9%	$383
SL(word) (He et al. 2018)	35.76	3.74	0.310	0.0385	**7**	6%	6%	$375
HRED (Sordoni et al. 2015)	36.50	4.56	0.316	0.0336	10	6%	17%	$325
OVE+HRNE+LD	37.32	4.24	0.353	0.0357	9	9%	27%	$293
OVE+HRNE+AP+LD	39.12	4.74	0.375	0.0358	11	**0%**	**0%**	$152
OVE+HRNE+AP+PA+LD*	41.68	**4.85**	0.442	0.0430	9	**0%**	**0%**	$132
OVE+HRNE+AP+PA+LD+RL**	**42.90**	4.65	0.463	0.0432	9	**0%**	**0%**	**$125**

TABLE 2.2: Evaluation metrics for language and pricing evaluation of the models. ↑ indicates higher is better, while ↓ shows lower is better. The model represented by * is our full *Price Negotiator* model and the one indicated by ** is our full *Price Negotiator* when using reinforcement learning.

Model	**Turing Test**	**Comparison Test**			**Interactive Test**		
		Human-likeness	Language	Pricing	Human-likeness	Language	Pricing
SL(word)	35%	37%	34%	31%	2.2	3.2	2.6
Price Negotiator+RL	**49%**	**63%**	**66%**	**69%**	**3.8**	**4.0**	**4.3**

TABLE 2.3: Human study results. Turing test shows the rates at which dialogues generated from each model has been identified as human generated negotiations. Comparison results reveal the ratio of preferring the dialogues generated from each model over those from the other one. The last three columns demonstrate the average scores provided by humans after interactive negotiations with models.

Category	**Price Negotiator**	**Price Negotiator+RL**
Bike	$43	**$40**
Car	$712	**$607**
Electronics	$8	**$7**
Furniture	$23	**$22**
Housing	$135	**$117**
Phone	$20	**$19**
Overall	$151	**$131**

TABLE 2.4: The table shows the average distance of agreed prices by our **price negotiator** models from their understanding about the value of the item (from OVE). The smaller value indicates the model's ability to insist on achieving its initially estimated value.

Furthermore, a brief look at the IBELU scores demonstrates the superior performance of *price negotiator* model in comparison to all other ones. It means that this model acts most similarly to humans in different situations. Interestingly, a noticeable improvement in IBLEU score by applying reinforcement learning illuminates that RL pushes the agent to take more human-like decisions in different situations to persuade the opponent and receive the best reward.

Last but not least, the language assessment metrics in Table 2.3–including the Turing test and both human-likeness and language rates for both comparison and interactive tests–indicate that our proposed model has been accepted as a considerably more fluent and human-like negotiator by human participants.

Pricing Evaluation Results. Table 4.5 demonstrates results of calculating the pricing metrics. It is noticeable that both versions of the *price negotiator* models learn to propose consistent prices while maintaining the language quality. Besides, these models never make a mistake in offering prices which are in conflict with the prices discussed and agreed upon during the dialogue.

More importantly, the proposed *price negotiator* model understands the suitable agreement price for an item precisely. Table 4.5 shows remarkable decrease in agreed price divergence (the difference between the prices agreed by the model with those agreed by humans) resulted from *price negotiator* in comparison to those from other models. In other words, the proposed model can learn the value of items by an online value estimation and reach agreement on prices very close to those agreed by humans by taking human-like actions both in generating utterances and in proposing prices.

Finally, human pricing assessment results in Table 2.3 show our proposed *price negotiator* model consistently performs well in comparison to its counterpart.

Results from RL. In Table 4.5 and 2.4 we observe that using RL generally improves the performance. In particular, in Table 2.4 when RL is used, our approach learns to insist on the prior value estimated by OVE more effectively and utilise the language better to achieve its goal (i.e. buying/selling with minimal compromise to that estimated by OVE).

Ablation Study Results. Table 4.5 summarises the effect of each module independently. Simply adding *online value estimator* to the HRED model (OVE+HRNE+LD) results in a slight improvement in both IBLEU and human divergence metrics implying the agent's ability to better imitate human behaviour. Once the *action predictor (AP)* is added we observe significant improvement in negotiation, both linguistically and in agreement prices. It is worth noting that action predictor controls the decisions of the agent and leads to reaching agreements that are remarkably closer to those by humans. Furthermore, having *price adjuster (PA)* and putting all of the modules together, not only the language quality is enhanced, but crucially the gap between machine and human agreed prices decreased considerably. This is due to the agent's capability to use a specialised module that handles the prices. Finally, applying *reinforcement learning (RL)* helps the agent to make considerably better performance linguistically and price-wise. It should be stated that since the *language decoder (LD)* is an obligatory component and has the same architecture as other baselines, the ablation study over this component is not been considered.

EXTREMER CRUISER BIKE!! $250 I HAVE A EXTREMER CRUISER BIKE, HAS 26" WHEELS, 6 SPEED, IN GOOD CONDITION! IF INTERESTED OR HAVE ANY QUESTIONS PLEASE DON'T EMAIL. CALL OR TEXT #	*seller*: hi, are you interested in the bike? *buyer*: yes i am. how old is it? *seller*: it is in great condition. i 've only had it for about a year. i bought it for my daughter who was just used it. it's in great shape. *buyer*: i see that you are asking $250, but i 'd like to buy it for $225. *seller*: i can do $235 and i 'll throw in a free helmet. *buyer*: i can do $229, but that's a bit too high for me. *seller*: i can do $229. *buyer*: <offer: $229> *seller*: <accept>	Parker Apartments … $975 Parker is that thrill you feel when inspiration strikes. Perched proudly on Shattuck Avenue in Berkeley, this collection of 155 carefully crafted, sustainably designed apartments offer immediate…	*seller*: hello, are you interested in the apartment? *buyer*: yes i am. is it in good condition? *seller*: yes, it is in great condition. *buyer*: great! i would like to offer $915 for it. *seller*: i 'm sorry, but i can't go that low. i can do $920 and throw in a 1 year lease. *buyer*: i understand. i can do $920 and you have a deal. *seller*: {offer: $920} *buyer*: {accept}
2012 Toyota Prius $12,500 The car never been in accident. The car has 109000 k freeway mileage. The has been driving by one person only. The car is automatic with excellent condition.	*seller*: hello, are you interested in the car? *buyer*: yes, i am. how many miles does it have? *seller*: it hasn't been in a accident. it has a clean title and a clean title. *buyer*: i see. i am looking to spend around $11699 for it. *seller*: i can do $11750. *buyer* i see. i see that the car is in great condition, but i 'm willing to pay $11750 for it. *seller*: {offer: $11750} *buyer*: {accept}	Vintage Advent Heritage tower speakers $100 These speakers are in great condition. My grandfather used these up until a few years ago when he passed away. They've been sitting in the house not being used since. ..	*buyer*: hello, i am interested in your speakers. *seller*: great! i 'm asking $100 for it. *buyer*: i see that, but the price is a little high for me. would you be willing to go down to $60? *seller*: i 'm willing to negotiate. *buyer*: i can do $68 if you can deliver it. *seller*: i can do $94. *buyer*: i can do $75. *seller*: $88 and you have a deal *buyer*: {offer: $88} *seller*: {accept}

FIGURE 2.6: More samples from our *price negotiator* model.

2.7 Conclusion and Future Works

In this chapter, we proposed a visual goal-oriented dialogue model for the seller-buyer negotiation. Our model, *Price Negotiator* is a modular framework for negotiation that utilises insights from human's behaviour for disentangling various parts. In particular, we are the first model to incorporate a matching network for evaluating the underlying value of an item by consulting online stores in negotiation. Experiments on CraigslistBargain dataset show the superior performance of the proposed model both linguistically and in reaching a human-like agreement price in various scenarios.

For future we consider improving the current approach by: (a) adding external knowledge about the cost and the availability of side-offers, like free delivery; and (b) applying pre-trained language models, such as BERT (Devlin et al. 2018), that may improve the understanding and generation performance.

Chapter 3

Active Learning by Feature Mixing

The main focus of the chapter is active learning (AL), a framework with which an artificial agent interactively seeks more information by selecting the most informative subset of unlabelled samples and asking their labels from human experts. This can efficiently improve the agent's performance using a small labelled set of examples. We describe our proposed AL method that effectively decreases the labelling costs for training a deep neural model to achieve a certain level of performance.

3.1 Overview

The promise of active learning (AL) is to reduce labelling costs by selecting the most valuable examples to annotate from a pool of unlabelled data. Identifying these examples is especially challenging with high-dimensional data (*e.g.* images, videos) and in low-data regimes. In this chapter, we propose a novel method for batch AL called ALFA-Mix. We identify unlabelled instances with sufficiently-distinct features by seeking inconsistencies in predictions resulting from interventions on their representations. We construct interpolations between representations of labelled and unlabelled instances then examine the predicted labels. We show that inconsistencies in these predictions help discovering features that the model is unable to recognise in the unlabelled instances. We derive an efficient implementation based on a closed-form solution to the optimal interpolation causing changes in predictions. Our method outperforms all recent AL approaches in 30 different settings on 12

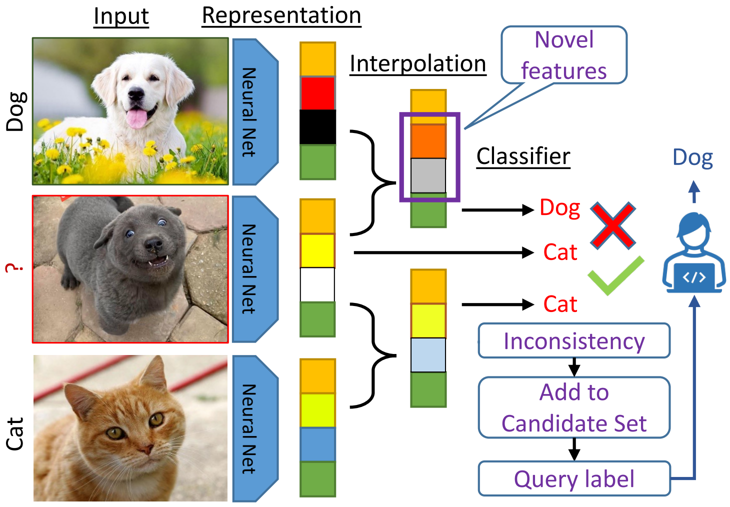


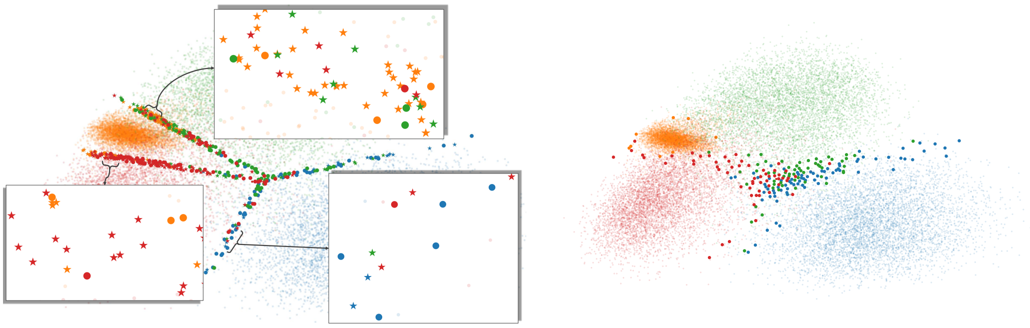
FIGURE 3.1: We propose to form linear combinations (*i.e.* interpolations or mixing) of the features of an unlabelled instance (middle image) and of labelled ones (top and bottom images). The interpolated features are passed through the current classifier. We show that inconsistencies in the predicted labels indicate that the unlabelled instance may have novel features to learn from.

benchmarks of images, videos, and non-visual data. The improvements are especially significant in low-data regimes and on self-trained vision transformers, where ALFA-Mix outperforms the state-of-the-art in 59% and 43% of the experiments respectively.

3.2 Introduction

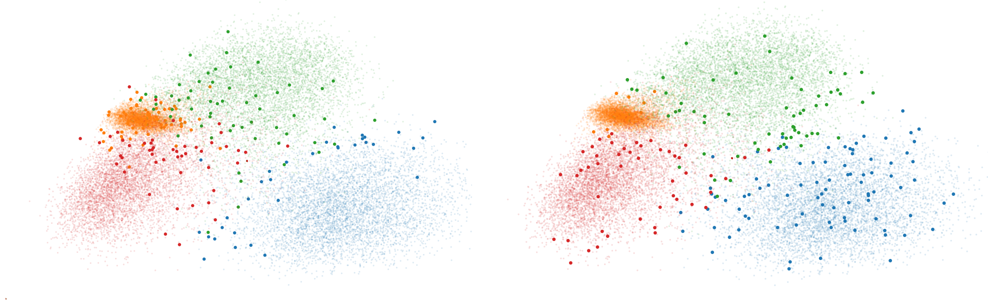
The success of machine learning applications depends on the quality and volume of the annotated datasets. High quality data annotations can be slow and expensive. Active learning (AL) aims to actively select the most valuable samples to be labelled in the training process iteratively, to boost the predictive performance. A popular setting called *batch* AL (Settles 2009) fixes a budget on the size of the batch of instances to be sent to an oracle for labelling. The process is repeated over multiple rounds, allowing the model to be updated iteratively. The core challenge is therefore to identify the most valuable instances to be included in this batch at each round, depending on the current model.

Various AL strategies have been proposed differing in predicting (1) how informative a particular unlabelled instance will be (*i.e.* uncertainty estimation (Roth and



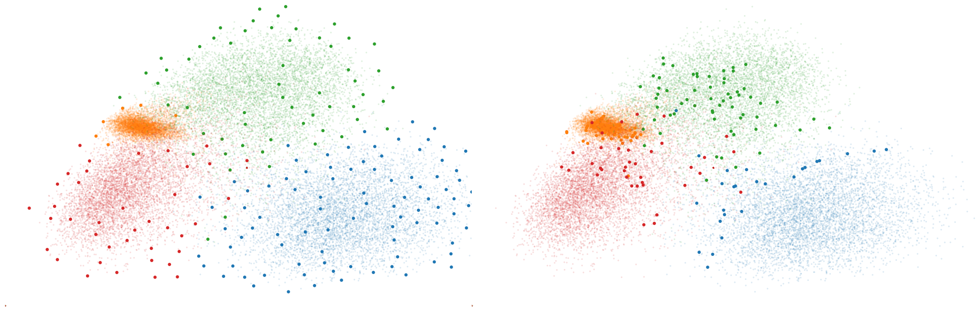
(A) **ALFA-Mix** (ours)

(B) CDAL (ECCV 2020)



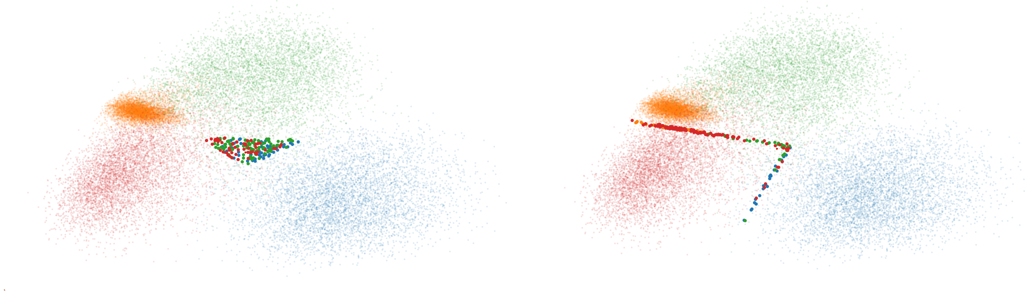
(C) BADGE (ICLR 2020)

(D) GCNAL (CVPR 2021)



(E) CoreSet (ICLR 2018)

(F) BALD (ICML 2017)



(G) Entropy

(H) DFAL

FIGURE 3.2: Visualization of sample selection behaviours of various AL methods in the latent space (see the Appendix for additional methods). The larger dots represent the selected samples to label; smaller dots represent unlabelled ones. Our approach finds a candidate set (demonstrated by stars in 3.2a) of unlabelled instances with inconsistencies in their label prediction when interpolated with labelled representations. It selects a diverse set of samples lying close to the all four borders for the labelling (with three zoom-in windows). The demonstration problem is that of identifying 4 classes from MNIST (illustrated above by 4 colours) using a MLP. An initial training set of 200 randomly selected points and their labels was provided, with each method given a budget of 200 additional labels. The features are projected to two-dimensions for visualization.

Small 2006; Wang and Shang 2014; Gal, Islam, and Ghahramani 2017; Ducoffe and Precioso 2018)) or (2) how varied a set of instances will be (*i.e.* diversity estimation (Yang et al. 2015; Sener and Savarese 2018)), or both (Huang, Jin, and Zhou 2010; Hsu and Lin 2015; Agarwal et al. 2020). Recent deep learning based AL techniques include, for example, the use of an auxiliary network to estimate the loss of unlabelled instances (Yoo and Kweon 2019), the use of generative models like VAEs to capture distributional differences (Sinha, Ebrahimi, and Darrell 2019; Kim et al. 2021), and the use of graph convolutional networks to relate unlabelled and labelled instances (Caramalau, Bhattarai, and Kim 2021).

Despite much progress made, current AL methods still struggle when applied to deep neural networks, with high-dimensional data, and in a low-data regime. We hypothesised that the representations learned within deep neural networks may be leveraged to reason about the model's uncertainty while alleviating the challenges associated with high-dimensional data. Some existing methods only consider the model's output, but we believe that this cannot convey a complete picture of the model's current state. Assessing the uncertainty in the model is particularly important in a low-data regime since the number of available training examples is small. This motivation has led to methods like BADGE (Ash et al. 2020) which uses gradients through the classifier layer of the network. Besides its relatively poor performance in lo-data regimes (Ash et al. 2020), the drawback is a high computational cost due to the high dimensionality of the gradient embeddings, making the method impractical for deep models with latent representations of high dimensions, large datasets, and large numbers of classes.

In this chapter, we present a novel and efficient AL method, named Active Learning by FeAture Mixing (ALFA-Mix), based on the manipulation of latent representations of the data. We identify informative unlabelled instances by evaluating the variability of the labels predicted for perturbed versions of these instances. These perturbed versions are instantiated in feature space as convex combinations of unlabelled and labelled instances (see Figure 3.1). This approach effectively explores the neighbourhood surrounding an unlabelled instance by interpolating its features with those of previously-labelled ones. Convex combinations of features have been already used in other contexts such as data augmentation, using random interpolations (Zhang et al. 2018a; Verma et al. 2019a; Verma et al. 2019b; Zhou et al. 2021) or actual solutions to an optimisation problem (Abbasnejad et al. 2020; Parvaneh et al. 2020).

We provide a theoretical support for the method. In particular, under a norm-constraint on the interpolation ratio, we show that the interpolation is equivalent to considering (1) the difference between the features of the unlabelled instance and the labelled ones and (2) the gradient of the model w.r.t the features at the unlabelled point. Discovering new features considering (1) and (2) leads us to finding an optimal interpolated point deterministically, at a minimal computing cost. Rather than using all the labelled data for these interpolations, we choose a subset we call anchors to capture the common features for each class. Subsequently, we construct a candidate set by choosing the instances from the unlabelled set that when mixed with these anchors lead to a change in the model's prediction for those instances. Then, to ensure selected instances are diverse, we perform a simple clustering in the candidate set and choose their centroids as the points to be queried.

The contributions of this chapter are as follows.

- Instead of interrogating an unlabelled instance directly, we interpolate its representation features from the labelled instances to uncover its hidden traits. To the best of our knowledge, it is the first of its kind in AL. Unlike existing methods that reply solely on the predicted output, we harness useful information from the feature representations as an indication of which features are novel for the model.
- We show that optimal interpolation/mixing for each instance that underscores the novel features with which the model could change prediction, has a closed-form solution making our approach efficient and scalable.
- We show that our approach outperforms its counterparts over 9 image, 2 OpenML, and one video datasets in various settings of architecture, network initialisation, and budget choice. Our approach consistently achieves higher accuracy than existing

methods, with particularly significant gains in a low-data regime.

- We provide the first investigation into using AL in vision transformers: we demonstrate the effectiveness of ALFA-Mix on a self-trained vision transformer (Caron et al. 2021), performing better than random selection in all tests, and 43% better than the state-of-the-art. In addition, our approach performs significantly better that its counterparts for video classification using transformers (Fan et al. 2021).

3.3 Related Work

Active learning strategies can be broadly categorised into three types: diversity-based, uncertainty-based, and hybrid sampling, according to the nature of their acquisition function. Diversity-based approaches aim to select samples that best represent the whole of the available unlabelled set. A variety of approaches have been proposed that cluster the unlabelled samples based on feature representations (Yang et al. 2015), or construct a core-set over the latent features to identify a suitably diverse set of samples (Sener and Savarese 2018).

Uncertainty-based methods seek to identify the unlabelled samples that are most ambiguous to the current model that has been trained over the present labelled set based on the target objective function. The assumption here is that having these uncertain samples labelled will add the most value to the next model training round. Entropy and the confidence of the predictions (Wang and Shang 2014), the margin between the confidence of the highest and second highest predicted classes (Roth and Small 2006), the information gain in the model parameters in a Bayesian framework (Gal, Islam, and Ghahramani 2017), and the variance between the predicted probabilities within the ensemble (Beluch et al. 2018) have all been proposed as measures of uncertainty. These methods favour points that lie close to the decision boundary, but as they rely entirely on the predicted class likelihoods they ignore the value of the feature representation itself. The closest method to that which we propose here is the deep fool attack learning (DFAL) approach (Ducoffe and Precioso 2018) where the distance to the decision boundary is approximated by perturbation, using techniques originally developed for adversarial attacks (Moosavi-Dezfooli, Fawzi, and Frossard 2016). Adversarial examples may expose vulnerability of the network architecture to particular patterns in the input rather than the distribution of the labels over latent space. That may lead to incorrect selection of instances that have patterns that are easily manipulated rather than helping to shape a more consistent decision boundary. Random perturbations are unlikely to lie within the true data distribution, and thus risk wasting labelling cost on feature values that can never arise in practice. Rather than repeatedly adding random noise in the input space, the method we propose here (ALFA-Mix) interpolates in latent space. ALFA-Mix is not only faster, it also significantly outperforms the DFAL approach.

Recently, a series of model-based active learning have been developed whereby a separate model is trained for active instance selection. Various objectives, either task-agnostic (*e.g.* variational adversarial active learning (Sinha, Ebrahimi, and Darrell 2019), graph convolutional active learning (Caramalau, Bhattarai, and Kim 2021)) or task-aware (*e.g.* target loss prediction (Yoo and Kweon 2019)), have been proposed as for training these models. Additionally, (Choi et al. 2021) has married model-based algorithms with conventional ones by combining a variational Bayes network with feature representations from the target model. In addition to sensitivity to hyper-parameters and additional computational cost, these AL methods do not consider the diversity of the selected samples and are prone to selecting samples with repetitive patterns. Moreover, our experiments show their poor performances in low-data regime.

Hybrid AL methods exploit both diversity and uncertainty in their sample selection methodologies. A mini-max strategy was proposed in (Huang, Jin, and Zhou 2010), for example, that maximises both the informativeness and representativeness of the samples. Interestingly, a method that learns to combine different AL strategies was presented in (Hsu and Lin 2015). Additionally, (Agarwal et al. 2020) exploits the predicted probabilities in images to select samples from diverse contexts (*i.e.* images of objects with varied backgrounds). Recently, (Ash et al. 2020) proposed to cluster the gradients of the final output layer of the target model as the features of the unlabelled samples that implicitly encompass the uncertainty information. Despite their state-of-the-art results on some image and non-image datasets, their approach is not scalable to larger tasks with numerous number of classes. Our approach not only consistently outperforms their method by a large margin in different settings, but it also is extremely efficient and scalable to large tasks.

3.4 Methodology

3.4.1 Problem Definition

Without loss of generality, we consider our learning objective to be training a supervised multiclass classification problem with K classes. A learner is actively trained in iterations of interactions with an oracle. At each iteration, this active learner has access to a small set of labelled data

$$\mathcal{D}^l = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^M \qquad \boldsymbol{x}_i \in \mathcal{X}, \quad y_i \in \{1, \ldots, K\},$$

where $\boldsymbol{x}_i$ represents the input (*e.g.* an image or a video clip) and y_i stands for the associated class label. The learner also has access to a set of unlabelled data $\mathcal{D}^u$ from which B number of instances are chosen to be labelled by the oracle. The labelled samples are then added to $\mathcal{D}^l$ to update the model. The performance of the model is evaluated on an unseen test dataset.

The learner is a deep neural network $f = f_c \odot f_e$ parameterised by $\boldsymbol{\theta} = \{\boldsymbol{\theta}_e, \boldsymbol{\theta}_c\}$. Here, $f_e : \mathcal{X} \rightarrow \mathbb{R}^D$ is the backbone which encodes the input to a D-dimensional representation in a latent space, *i.e.* $\boldsymbol{z} = f_e(\boldsymbol{x}; \boldsymbol{\theta}_e)$. Further, $f_c : \mathbb{R}^D \rightarrow \mathbb{R}^K$ is a classifier *e.g.* multi-layer perceptron (MLP) that maps the instances from their representations to their corresponding logits which can be converted to class likelihoods by:

$$p(y \mid \boldsymbol{z}; \boldsymbol{\theta}) = \mathrm{softmax}(f_c(\boldsymbol{z}; \boldsymbol{\theta}_c)).$$

We optimise the parameters end-to-end by minimising the cross-entropy loss over the labelled set: $\mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}^l}[\ell(f_c \odot f_e(\boldsymbol{x};\boldsymbol{\theta}), y)]$. The prediction of the label (*i.e.* pseudo-label) for an unseen instance is $y^*_{\boldsymbol{z}} = \arg\max_y f^y_c(\boldsymbol{z};\boldsymbol{\theta}_c)$ where $\boldsymbol{z} = f_e(\boldsymbol{x};\boldsymbol{\theta}_e)$ and f^y_c is the logit output for class y. Additionally, the logit of the predicted label is denoted as $f^*_c(\boldsymbol{z}) := f^{y^*_{\boldsymbol{z}}}_c(\boldsymbol{z})$[*]. We also denote $\boldsymbol{Z}^u = \{f_e(\boldsymbol{x}), \forall \boldsymbol{x} \in \mathcal{D}^u\}$ the set for representations of the unlabelled data and $\boldsymbol{Z}^l$ its labelled counterpart. We compute the average representation $\boldsymbol{z}^\star$ of the labelled samples per class, and call it anchor. The anchors for all classes form the anchor set $\boldsymbol{Z}^\star$, and serve as representatives of the labelled instances.

3.4.2 Feature Mixing

The characteristics of the latent space plays a crucial role in identifying the most valuable samples to be labelled. Our intuition is that the model's incorrect prediction is mainly due to novel "features" in the input that are not recognisable. Thus, we approach the AL problem by first probing the features learned by the model. To that end, we use a convex combination (*i.e.* interpolation) of the features as a way to explore novel features in the vicinity of each unlabelled point. Formally, we consider our interpolation between the representations of the unlabelled and labelled instances, $\boldsymbol{z}^u$ and $\boldsymbol{z}^\star$ respectively (we use the labelled anchor here for efficiency) as

$$\tilde{\boldsymbol{z}}_{\boldsymbol{\alpha}} = \boldsymbol{\alpha}\boldsymbol{z}^\star + (1-\boldsymbol{\alpha})\boldsymbol{z}^u, \qquad \boldsymbol{\alpha} \in [0,1)^D, \tag{3.1}$$

where $\boldsymbol{\alpha}$ represents the interpolation ratio. This process can be seen as a way of sampling a new instance without explicitly modelling the joint probability of the labelled and unlabelled instances (Zhang et al. 2018a; Lesniak, Sieradzki, and Podolak 2019; Abbasnejad et al. 2020; Parvaneh et al. 2020), *i.e.*

$$\boldsymbol{z} \sim p(\boldsymbol{z} \mid \boldsymbol{z}^u, \boldsymbol{Z}^\star, \boldsymbol{\alpha}) \equiv \boldsymbol{\alpha}\boldsymbol{z}^\star + (1-\boldsymbol{\alpha})\boldsymbol{z}^u, \;\; \boldsymbol{z}^\star \sim \boldsymbol{Z}^\star. \tag{3.2}$$

---

[*] For brevity, when the parameters $\boldsymbol{\theta}_e$ and $\boldsymbol{\theta}_c$ are clear from the context, we refrain from explicitly including them.

We consider interpolating an unlabelled instance with all the anchors representing different classes to uncover the sufficiently distinct features by considering how the model's prediction changes. For that, we investigate the change in the pseudo-label (*i.e.* y^*) for the unlabelled instance and the loss incurred with the interpolation. We expect that a small enough interpolation with the labelled data should not have a consequential effect on the predicted label for each unlabelled point.

Using a first-order Taylor expansion[†] w.r.t. $\boldsymbol{z}^u$, the model's loss for predicting the pseudo-label of an unlabelled instance at its interpolation with a labelled one can be written as:

$$\ell\left(f_c\left(\tilde{\boldsymbol{z}}_{\boldsymbol{\alpha}}\right), y^*\right) \approx \ell\left(f_c(\boldsymbol{z}^u), y^*\right) + \left(\tilde{\boldsymbol{z}}_{\boldsymbol{\alpha}} - \boldsymbol{z}^u\right)^{\intercal} . \nabla_{\boldsymbol{z}^u} \ell\left(f_c\left(\boldsymbol{z}^u\right), y^*\right) . \tag{3.3}$$

We also know that considering $\tilde{\boldsymbol{z}}_{\boldsymbol{\alpha}} = \boldsymbol{\alpha}\boldsymbol{z}^{\star} + (1-\boldsymbol{\alpha})\boldsymbol{z}^u$, we will have

$$\begin{aligned} \tilde{\boldsymbol{z}}_{\boldsymbol{\alpha}} - \boldsymbol{z}^u &= (\boldsymbol{\alpha}\boldsymbol{z}^{\star} + (1-\boldsymbol{\alpha})\boldsymbol{z}^u) - \boldsymbol{z}^u \\ &= \boldsymbol{\alpha}\boldsymbol{z}^{\star} + \boldsymbol{z}^u - \boldsymbol{\alpha}\boldsymbol{z}^u - \boldsymbol{z}^u \\ &= \boldsymbol{\alpha}\boldsymbol{z}^{\star} - \boldsymbol{\alpha}\boldsymbol{z}^u \\ &= \boldsymbol{\alpha}(\boldsymbol{z}^{\star} - \boldsymbol{z}^u) . \end{aligned} \tag{3.4}$$

By replacing this in Eq. (3.3), the loss of the model for predicting the pseudo-label at an interpolation point between a pair of unlabelled and labelled instances can be re-written as[‡]

$$\ell\left(f_c\left(\tilde{\boldsymbol{z}}_{\boldsymbol{\alpha}}\right), y^*\right) \approx \ell\left(f_c(\boldsymbol{z}^u), y^*\right) + \left(\boldsymbol{\alpha}(\boldsymbol{z}^{\star} - \boldsymbol{z}^u)\right)^{\intercal} . \nabla_{\boldsymbol{z}^u} \ell\left(f_c\left(\boldsymbol{z}^u\right), y^*\right) , \tag{3.5}$$

[†] We use the first-order Taylor expansion for its simplicity and efficiency. In practice, one can use higher orders to get a more accurate approximation.

[‡] This statement is true for any given instance and any convex combination of points in the latent space. For AL, we particularly focus on unlabelled instances though.

which for a sufficiently small $\boldsymbol{\alpha}$, *e.g.* $\|\boldsymbol{\alpha}\| \leq \epsilon$ is almost exact. Consequently, for the full labelled set, by choosing the max loss from both sides we have:

$$\begin{aligned} \max_{\boldsymbol{z}^{\star} \sim \boldsymbol{Z}^{\star}} \left[\ell\left(f_c\left(\tilde{\boldsymbol{z}}_{\boldsymbol{\alpha}}\right), y^{*}\right)\right] - \ell\left(f_c(\boldsymbol{z}^{u}), y^{*}\right) \approx \\ \max_{\boldsymbol{z}^{\star} \sim \boldsymbol{Z}^{\star}} \left[\left(\boldsymbol{\alpha}(\boldsymbol{z}^{\star} - \boldsymbol{z}^{u})\right)^{\intercal} . \nabla_{\boldsymbol{z}^{u}} \ell\left(f_c\left(\boldsymbol{z}^{u}\right), y^{*}\right)\right]. \end{aligned} \tag{3.6}$$

Intuitively, when performing interpolation, the change in the loss is proportionate to two terms: (a) the difference of features of $\boldsymbol{z}^{\star}$ and $\boldsymbol{z}^{u}$ proportionate to their interpolation $\boldsymbol{\alpha}$, and (b) the gradient of the loss w.r.t the unlabelled instance. The former determines which features are novel and how their value could be different between the labelled and unlabelled instance. On the other hand, the later determines the sensitivity of the model to those features. That is, if the features of the labelled and unlabelled instances are completely different but the model is reasonably consistent, there is ultimately no change in the loss, and hence those features are not considered novel to the model.

The choice of $\boldsymbol{\alpha}$ is input specific and determines the features to be selected. As such, in Sec 3.4.3 we introduce a closed form solution for finding a suitable value for $\boldsymbol{\alpha}$. Finally, we note that the interpolations utilised here have some interesting properties that are further discussed in the supplements.

3.4.3 Optimising the Interpolation Parameter $\boldsymbol{\alpha}$

Since manually choosing a value for $\boldsymbol{\alpha}$ is non-trivial, we devise a simple optimisation approach to choose the appropriate value for a given unlabelled instance. To that end, we note that, as observed from Eq. (3.6), the worst case of maximum change in the loss is when we choose $\boldsymbol{\alpha}$ that maximises the loss at the interpolation point. However, using the right-hand-side of the Eq. (3.6), we devise the objective for choosing $\boldsymbol{\alpha}$ as:

$$\boldsymbol{\alpha}^{*} = \underset{\|\boldsymbol{\alpha}\| \leq \epsilon}{\arg\max} \left(\boldsymbol{\alpha}(\boldsymbol{z}^{\star} - \boldsymbol{z}^{u})\right)^{\intercal} . \nabla_{\boldsymbol{z}^{u}} \ell(f_c(\boldsymbol{z}^{u}), y^{*}), \tag{3.7}$$

where ϵ is a hyper-parameter governing the magnitude of the mixing. Intuitively, this optimisation chooses the hardest case of $\boldsymbol{\alpha}$ for each unlabelled instance and anchor.

Using a 2-norm constraint on $\boldsymbol{\alpha}$ in Eq. 3.7, we approximate the optimum interpolation ratio as

$$\boldsymbol{\alpha}^* = \underset{\|\boldsymbol{\alpha}\|_2 \leq \epsilon}{\arg\max}\, (\boldsymbol{\alpha}(\boldsymbol{z}^\star - \boldsymbol{z}^u))^\intercal . \nabla_{\boldsymbol{z}^u} \ell(f_c(\boldsymbol{z}^u), y^*). \tag{3.8}$$

By multiplying both sides of the constraint in Eq. 3.8 by $\|(\boldsymbol{z}^\star - \boldsymbol{z}^u)\|_2$, we have

$$\|\boldsymbol{\alpha}\|_2 \, \|(\boldsymbol{z}^\star - \boldsymbol{z}^u)\|_2 \leq \epsilon \|(\boldsymbol{z}^\star - \boldsymbol{z}^u)\|_2.$$

Based on Cauchy-Schwartz inequality, we know that $\|\boldsymbol{\alpha}(\boldsymbol{z}^\star - \boldsymbol{z}^u)\|_2 \leq \|\boldsymbol{\alpha}\|_2 \, \|(\boldsymbol{z}^\star - \boldsymbol{z}^u)\|_2$. Thus, we can infer

$$\|\boldsymbol{\alpha}(\boldsymbol{z}^\star - \boldsymbol{z}^u)\|_2 \leq \epsilon \|(\boldsymbol{z}^\star - \boldsymbol{z}^u)\|_2 = \epsilon'.$$

Therefore, we can change the optimisation problem to

$$\boldsymbol{\alpha}^* = \underset{\|\boldsymbol{\alpha}(\boldsymbol{z}^\star - \boldsymbol{z}^u)\|_2 \leq \epsilon'}{\arg\max} \quad (\boldsymbol{\alpha}(\boldsymbol{z}^\star - \boldsymbol{z}^u))^\intercal . \nabla_{\boldsymbol{z}^u} \ell \left(f_c(\boldsymbol{z}^u), y^*\right).$$

We can use the dual norm (Boyd and Vandenberghe 2004) of the above equation to approximate the optimum value for $\boldsymbol{u} = \boldsymbol{\alpha}(\boldsymbol{z}^\star - \boldsymbol{z}^u)$, which is

$$\boldsymbol{u}^* = \epsilon' \frac{\nabla_{\boldsymbol{z}^u} \ell \left(f_c(\boldsymbol{z}^u), y^*\right)}{\|\nabla_{\boldsymbol{z}^u} \ell \left(f_c(\boldsymbol{z}^u), y^*\right)\|_2}. \tag{3.9}$$

After replacing the actual values for $\boldsymbol{u}$ and ϵ', we have

$$\boldsymbol{\alpha}^* \approx \epsilon \frac{\|(\boldsymbol{z}^\star - \boldsymbol{z}^u)\|_2 \nabla_{\boldsymbol{z}^u} \ell(f_c(\boldsymbol{z}^u), y^*)}{\|\nabla_{\boldsymbol{z}^u} \ell(f_c(\boldsymbol{z}^u), y^*)\|_2} \oslash (\boldsymbol{z}^\star - \boldsymbol{z}^u), \tag{3.10}$$

where $\oslash$ represents element-wise division. This approximation makes the optimisation of the interpolation parameter efficient and our experiments show that it will not have significant detrimental effects on the final results compared to directly optimising for $\boldsymbol{\alpha}$ to maximise the loss.

3.4.4 Candidate Selection

For AL it is reasonable to choose instances to be queried whose loss substantially change with interpolation according to Eq. (3.6). This corresponds to those instances for which the model's prediction change and have novel features. Intuitively, as depicted in Figure. 3.2a, these samples are placed close to the decision boundary in the latent space. Alternatively, we expect a small interpolation should not affect the model's loss when it is reasonably confident in its recognition of the features of the input. We, then, create our candidate set as:

$$\mathcal{I} = \left\{ \boldsymbol{z}^u \in \boldsymbol{Z}^u \,\middle|\, \exists \boldsymbol{z}^\star \in \boldsymbol{Z}^\star,\ f_c^*(\tilde{z}_{\boldsymbol{\alpha}}) \neq y^*_{\boldsymbol{z}^u} \right\}. \tag{3.11}$$

The size of the selected set $\mathcal{I}$ could potentially be larger than the budget B. In addition, ideally we seek *diverse* samples since most instances in $\mathcal{I}$ could be chosen from the same region (*i.e.* they might share the same novel features). To that end, we propose to cluster the instances in $\mathcal{I}$ into B groups based on their feature similarities and further choose the closest samples to the centre of each cluster to be labelled by oracle. This ensures the density of the space represented by $\mathcal{I}$ samples, is reasonably approximated using B instances. We simply use k-MEANS which is widely accessible. Similar strategy is also used by (Ash et al. 2020) to encourage diversity. Our approach is summarised in Algorithm 1.

3.4.5 Relations Between ALFA-Mix and Other Baselines

Using gradients in BADGE: From Eq. (3) in the main text we can understand that when the prediction is accurate and confident, small movements of the latent representation towards different directions (declared by anchors) should not change the prediction. Otherwise, as per right-hand-side of the equation, either the surface has

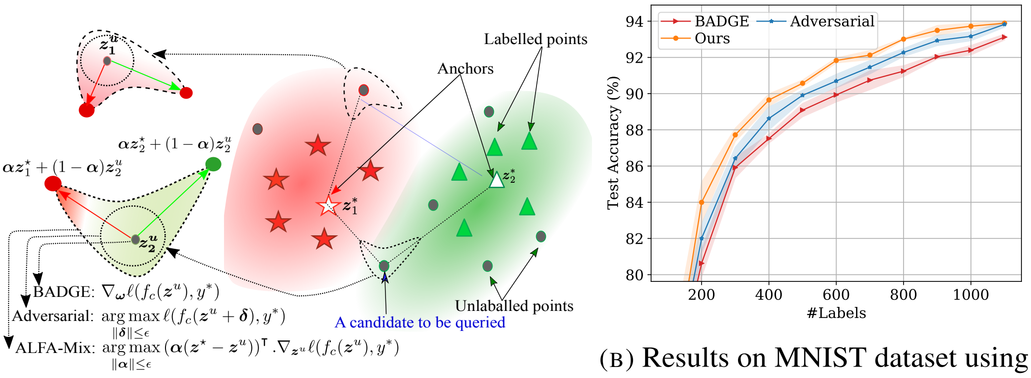
Algorithm 1: Our active learning algorithm.

```
Inputs: initial labelled set $\mathcal{D}^l$; unlabelled pool $\mathcal{D}^u$; labelling budget at each round $B$; mixing parameter $\epsilon$;
for $i = 1$ to max_rounds do
    Train the model $f$ using the labelled data $\mathcal{D}^l$.
    Initialise $\boldsymbol{Z}^\star$ based on the representations of $\mathcal{D}^l$.
    $\mathcal{I} = \{\}$.
    for $\boldsymbol{x}^u \in \mathcal{D}^u$ do
        $\boldsymbol{z}^u = f_e(\boldsymbol{x}^u)$.
        for $\boldsymbol{z}^\star \in \boldsymbol{Z}^\star$ do
            Calculate $\boldsymbol{\alpha}^*$ using $\epsilon$ and Eq. 3.10.
            $\tilde{\boldsymbol{z}} = \boldsymbol{\alpha}^* \boldsymbol{z}^\star + (1 - \boldsymbol{\alpha}^*) \boldsymbol{z}^u$.
            if $\arg\max_y (f_c^y(\boldsymbol{z}_u)) \neq \arg\max_y (f_c^y(\tilde{\boldsymbol{z}}))$ then
                $\mathcal{I} = \mathcal{I} \cup (\boldsymbol{x}^u, \boldsymbol{z}^u)$.
                Break
    Cluster the samples in $\mathcal{I}$ into $B$ clusters.
    Select samples at the centre of each cluster ($\mathcal{C}$).
    $Y^{new} = \text{Query}(\mathcal{C})$.
    $\mathcal{D}^l = \mathcal{D}^l \cup (\mathcal{C}, Y^{new}),\ \mathcal{D}^u = \mathcal{D}^u \setminus \mathcal{C}$.
```

changed dramatically or the unlabelled features is far from the labelled representations (*i.e.* the features of the unlabelled instance are novel). This is one of the major differences of our approach when compared with BADGE that only relies on the gradients of the unlabelled instances (Figure. 3.3).

Adversarial perturbation of features: To show the importance of the feature interpolations with labelled representations in our approach, we also considered using adversarial noise as an alternative perturbation mechanism. For that, we examined adding small values of noise $\boldsymbol{\delta}$ to the latent representations of each unlabelled point (instead of using interpolations with anchors) to find inconsistencies in their predicted labels. Following Eq. (3) and Eq. (4) in the main text, we set the objective for finding the optimum noise vector $\boldsymbol{\delta}^*$ as:

$$\boldsymbol{\delta}^* = \arg\max_{\|\boldsymbol{\delta}\| \leq \epsilon} \ell(f_c(\boldsymbol{z}^u + \boldsymbol{\delta}), y^*). \tag{3.12}$$



(A) Sampling strategies.

(B) Results on MNIST dataset using an MLP and a small budget of size 100 at each round.

FIGURE 3.3: A comparative depiction of our approach (ALFA-Mix) vs. BADGE vs. adversarial in the latent space: Since ours considers interpolations in the direction of the anchor points and proportional to their distance, it better evaluates the consistency of the predictions in the latent space. When points are less consistent, it is more intuitive to consider them as candidates to be queried (*e.g.* z_2^u in this figure is inconsistent after the interpolation, and hence likely to be queried).

Similarly, using a first-order Taylor expansion w.r.t. z^u and its dual norm, we can approximate the optimum noise as

$$\delta^* \approx \epsilon \frac{\nabla_{z^u}\ell(f_c(z^u), y^*)}{\|\nabla_{z^u}\ell(f_c(z^u), y^*)\|_2}. \tag{3.13}$$

After constructing a candidate set of unlabelled samples whose predicted labels are not consistent after the adversarial feature perturbation, we conduct clustering to sample a diverse set from the candidate set (similar to ALFA-Mix). Interestingly, as depicted in Figure. 3.3b, although the adversarial approach shows better performance in comparison to BADGE, it falls behind considerably when compared to ALFA-Mix. We believe that the main advantage of ALFA-Mix is the consideration of both the novelty of the features and the extent of gradient at each unlabelled point. It is worth mentioning that ALFA-Mix is able to identify more inconsistencies all over the decision boundary (Figure. (6c) in the main text).

Distribution matching. Denote $\Delta = \mathbb{E}_{p(z^l|\mathcal{D}^l)}\left[z^l\right] - \mathbb{E}_{p(z^u|\mathcal{D}^u)}\left[z^u\right]$ if we had the distributions in the latent space. We know that based on the definition of the interpolation between a pair of labelled and unlabelled samples (*i.e.* $\tilde{z}_{\alpha} = \alpha z^l + (1-\alpha) z^u$), we can have

$$\boldsymbol{z}^u = \frac{1}{1-\boldsymbol{\alpha}} \left(\tilde{\boldsymbol{z}}_{\boldsymbol{\alpha}} - \boldsymbol{\alpha}\boldsymbol{z}^l\right).$$

By taking the expectation from both side of the above equation for all the labelled samples we have

$$\boldsymbol{z}^u = \underset{p(\boldsymbol{z}^l|\mathcal{D}^l)}{\mathbb{E}} \left[\frac{1}{1-\boldsymbol{\alpha}} \left(\tilde{\boldsymbol{z}}_{\boldsymbol{\alpha}} - \boldsymbol{\alpha}\boldsymbol{z}^l\right)\right].$$

After replacing this in the definition of Δ, it is easy to show that:

$$\Delta = \frac{1}{(1-\boldsymbol{\alpha})} \left(\underset{p(\boldsymbol{z}^l|\mathcal{D}^l)}{\mathbb{E}} \left[\boldsymbol{z}^l\right] - \underset{p(\boldsymbol{z}^u|\mathcal{D}^u)}{\mathbb{E}} \left[\underset{p(\boldsymbol{z}^l|\mathcal{D}^l)}{\mathbb{E}} \left[\tilde{\boldsymbol{z}}_{\boldsymbol{\alpha}}\right]\right]\right).$$

That is, the interpolation operation we used here only affects difference of the expectation of distributions with a constant factor. When seen in light of Eq. (1) in the main text, it acts as a simple surrogate for a divergence measure. In fact, this relates our approach to other AL methods that their focus is on finding the distributional difference between labelled and unlabelled samples Sinha, Ebrahimi, and Darrell 2019; Caramalau, Bhattarai, and Kim 2021.

Gradient-based interpolation optimisation. Following Abbasnejad et al. 2020; Parvaneh et al. 2020, we could have utilised iterative gradient-based optimisation to find the optimum interpolation ratios (instead of the closed-form solution used in ALFA-Mix). For that, motivated by the condition in the Eq. (6) in the main text where we are interested in instances whose predictions flip with an interpolation in the latent space, we can choose $\boldsymbol{\alpha}$ as a solution to the following:

$$\boldsymbol{\alpha}^* = \underset{\boldsymbol{\alpha}\,\in\,[0,\alpha_{\max}]^D}{\arg\max}\ \ell(f_c(\boldsymbol{\alpha}\boldsymbol{z}^\star + (1-\boldsymbol{\alpha})\boldsymbol{z}^u), y^*), \tag{3.14}$$

$$\text{s.t.}\quad y^* = \underset{k\in\{1,\ldots,K\}}{\arg\max}\ f_c^k(\boldsymbol{z}^u), \quad \forall \boldsymbol{z}^u \in \boldsymbol{Z}^u, \quad \boldsymbol{z}^\star \in \boldsymbol{Z}^\star,$$

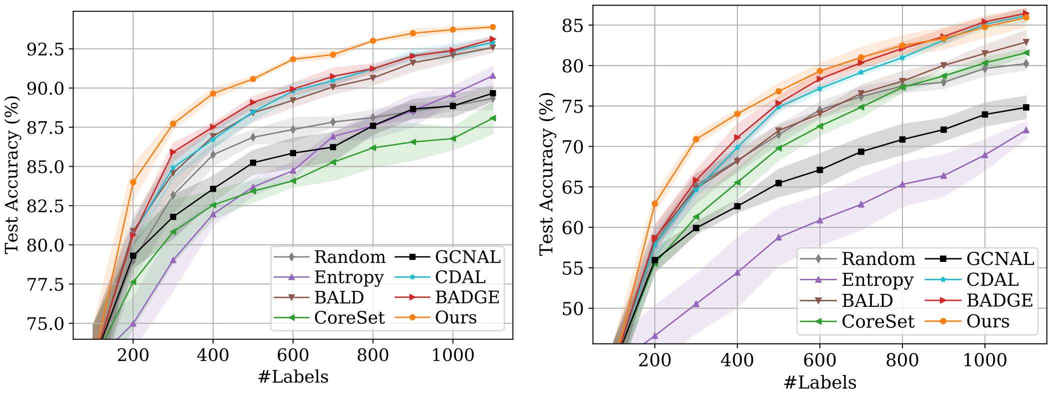
where α_{max} is a hyper-parameter governing the feature mixing ratios. Intuitively, this optimisation chooses the hardest case of $\boldsymbol{\alpha}$ for each unlabelled instance and anchor. We perform few iterations of projected gradient descent to optimise $\boldsymbol{\alpha}$. Our empirical studies reveal similar performances when using this objective in comparison to the closed-form one. However, the time required for the iterative gradient-based approach is much more than the closed-form one (*i.e.* when using 5 iterations of gradient update, it is 5x slower than ALFA-Mix).

3.5 Experiments and Results

3.5.1 Baselines

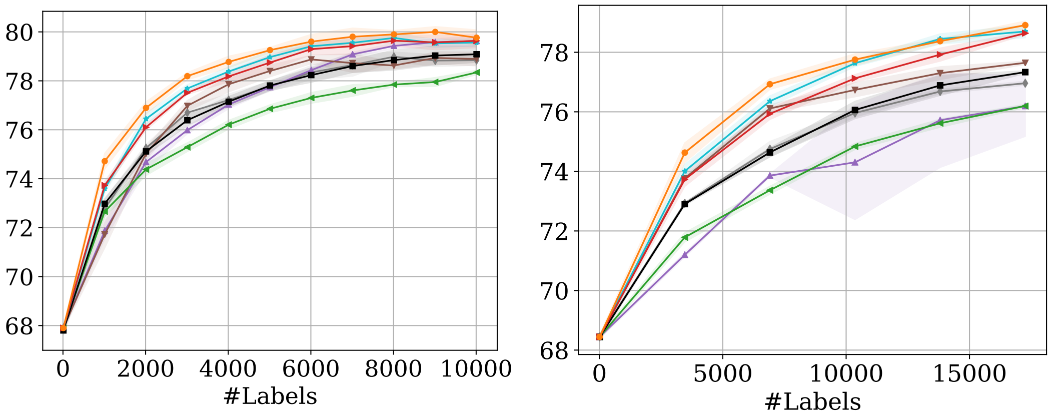
We compare ALFA-Mix with the following AL baselines:

- **Random**: a simple baseline that randomly selects B samples from the unlabelled pool at each round.
- **Entropy** (Wang and Shang 2014): A conventional AL approach that picks unlabelled instances with highest entropy.
- **BALD** (Gal, Islam, and Ghahramani 2017): An uncertainty model relying on Bayesian approaches that selects a set of samples with the highest mutual information between label predictions and posterior of the model approximated using dropout (Figure 3.2f).
- **Coreset** (Sener and Savarese 2018): An approach based on the core-set technique that chooses a batch of diverse representative samples of the whole unlabelled set (Figure. 3.2e).
- **Adversarial Deep Fool** (Ducoffe and Precioso 2018): An uncertainty method that utilises deep fool attacks to select a batch of unlabelled samples whose predictions flip with small perturbations.



(A) MNIST (MLP)

(B) OpenML-6 (MLP)



(C) MiniImageNet (ViT-Small)

(D) DomainNet-Real (ViT-Base)

FIGURE 3.4: Test accuracy plots across some of the employed settings. Each experiment has been repeated 5 times.

- **GCNAL** (Caramalau, Bhattarai, and Kim 2021): A model-based approach that learns a graph convolutional network to measures the relation between labelled and unlabelled instances (Figure. 3.2d)[§].

- **BADGE** (Ash et al. 2020): A hybrid approach that queries the centroids obtained from the clustering of the gradient embeddings (Figure. 3.2c).

- **CDAL** (Agarwal et al. 2020): A hybrid approach that exploits the contextual information in the predicted probabilities to choose samples with varied contexts (Figure. 3.2b)

[§] We employed CoreGCN variation in our experiments as results reported in (Caramalau, Bhattarai, and Kim 2021) show its superiority over the UncertainGCN version.

Dataset	Pool Size	Label Size	Input	Initial Instances	Budgets	Architectures	Initialisations
MNIST (Lecun et al. 1998)	$50,000$	10	28×28	100	$100, 1000$	MLP, LeNet-5	Random, Continue
EMNIST (Cohen et al. 2017)	$124,800$	26	28×28	260	$260, 2650$	MLP, LeNet-5	Random, Continue
SVHN (Netzer et al. 2011)	$50,000$	10	32×32	100	$100, 1000$	ResNet-18, DenseNet-121	Random
CIFAR10 (Krizhevsky 2009)	$50,000$	10	32×32	100	$100, 1000$	ResNet-18, DenseNet-121	Random
DomainNet-Real-10*	$4,673$	10	224×224	100	100	ResNet-18, DenseNet-121	Pre-trained
DomainNet-Real-20*	$8,615$	20	224×224	200	200	ResNet-18, DenseNet-121	Pre-trained
CIFAR100 (Krizhevsky 2009)	$50,000$	100	32×32	1000	1000	ViT-Small	Pre-trained
Mini-ImageNet (Sachin Ravi 2017)	$48,000$	100	84×84	1000	1000	ViT-Small	Pre-trained
DomainNet-Real (Peng et al. 2019)	$122,563$	345	224×224	3450	3450	ViT-Base, ResNet-18, DenseNet-121	Pre-trained
OpenML_6	$18,000$	26	16	100	100	MLP	Random
OpenML_155	$50,000$	9	10	100	100	MLP	Random

TABLE 3.1: A summary of diverse AL settings that we used in our image and non-image experiments. Overall, 30 different settings were utilised in our experiments to compare AL methods in various conditions. "Continue" represents the setting where the weights of the network initialise from those of the network trained in the previous round. Please note that datasets indicated by * are two small subsets of DomainNet-Real that has been used to compare AL methods on small datasets with high-resolution images.

3.5.2 Experiment Settings

Setting and Datasets: We conducted a comprehensive set of experiments in 30 different settings on multiple datasets to evaluate how ALFA-Mix compares to its counterparts. We define an AL setting as a combination of a specific dataset, a limited set of initial labelled samples, a particular type of deep neural network, a limited number of AL rounds, and a fixed labelling budget (batch) for each round.

Specifically, we experimented on MNIST (Lecun et al. 1998), EMNIST (Cohen et al. 2017), CIFAR10 (Krizhevsky 2009), CIFAR100 (Krizhevsky 2009), Mini-ImageNet (Sachin Ravi 2017), DomianNet-Real (Peng et al. 2019) as well as two subsets of DomainNet-Real for image datasets. Additionally, we extended our experiments to two more non-visual datasets from the OpenML[¶] repository. Furthermore, to reveal the effectiveness of each AL method in different data regimes, we utilised both small ($10 \times K$) and large ($100 \times K$) budget sizes. More importantly, the network architecture and its initial parameters are two more factors that we considered in our experiments. As for the choice of the architecture, we employed different common deep neural networks; including Multi-Layer Perceptron (MLP) (Ash et al. 2020), ResNet-18 (He et al. 2016), DenseNet-121 (Huang et al. 2017), as well as Vision Transformers (Dosovitskiy et al. 2021). Regarding the network initialisation, we considered three scenarios where at the start of each AL round[‖], the parameters are initialised randomly, from the model trained in the previous round (denoted as "Continue" in the results), or using pre-trained models (either from supervised or self-supervised (Caron et al. 2021) pre-training on ImageNet (Deng et al. 2009)). Please find for more details in Table 3.1.

We followed the supervised training setting proposed in (Ash et al. 2020) and optimised the network using all the labelled set (without any validation set) based on a cross-entropy loss and an Adam optimiser with a learning rate of $1e-3$ and $1e-4$

[¶] `https://www.openml.org`

[‖] After a new batch of samples are selected by AL method and added to the labelled set and before the model training.

for image and non-image datasets, respectively. Similarly, we continued the training using a batch size of 64 until the model reaches a certain early stopping condition (*i.e.* reaching a training accuracy above 99% (Ash et al. 2020)).

We set the number of rounds for each setting to 10, except for DomainNet-Real where we continue for 5 rounds. Additionally, to eliminate the effect of randomness in the results, we repeated each experiment 5 times with different random seeds.

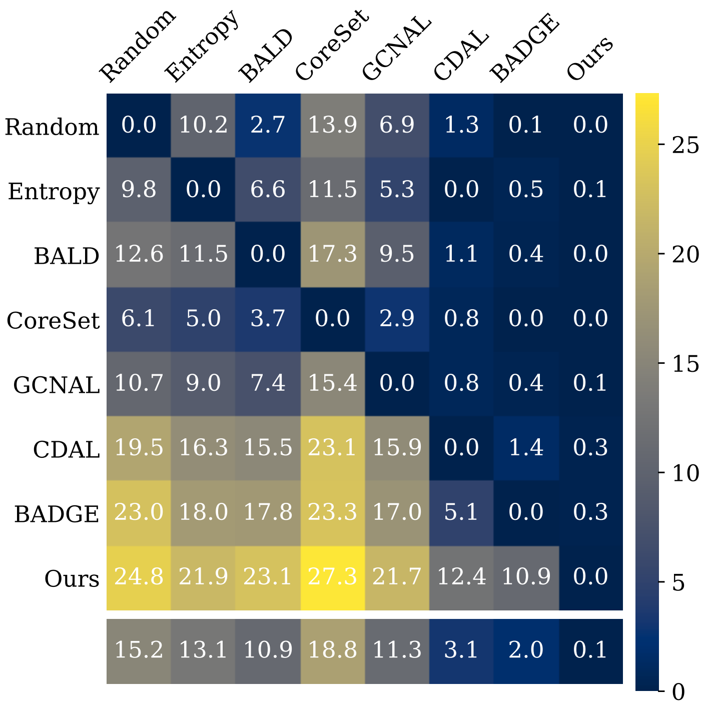
Comparison matrix We demonstrate the performance comparison between every pair of AL methods over various settings in a penalty matrix proposed in (Ash et al. 2020). Each cell of the matrix reveals the number of settings in which the method shown in the column is outperformed by the ones indicated in the row. It should be noted that each setting consists of conducting R rounds of AL with a specific labelling budget size B and using a particular model architecture on a single dataset. Since we repeat each setting with 5 different random seeds, at each round r in the setting we use t-score of the difference between the test performances ($d^r_{i,j} = a^r_i - a^r_j$) of each pair of AL methods (i, j) over the 5 repeats:

$$c^r_{i,j} = \frac{\sqrt{5}\mu^r}{\sigma^r}, \tag{3.15}$$

$$\mu^r = \frac{1}{5}\sum_{m=1}^{5} d^r_{i,j}, \quad \sigma^r = \sqrt{\frac{1}{5}\sum_{m=1}^{5}(d^r_{i,j} - \mu^r)^2},$$

where a^r_i and a^r_j are the test performances of methods i and j respectively at AL round r. Similar to (Ash et al. 2020), we also used a threshold of 2.776 for this score to decide if method i wins over method j. After clarifying the winner at each round of the setting, we calculate $C_{i,j} = \sum_{r=1}^{R} \mathbb{1}_{c^r_{i,j}>2.776}/R$ as the final victory score of AL method i over method j in that specific setting. Additionally, to compute the matrix over multiple settings, we simply report the element-wise sum of all the individual matrices.

Video classification: Since video classification is a more challenging task with higher annotation cost, we compare the AL performance on video classification tasks. All



	Random	Entropy	BALD	CoreSet	GCNAL	CDAL	BADGE	Ours
Random	0.0	10.2	2.7	13.9	6.9	1.3	0.1	0.0
Entropy	9.8	0.0	6.6	11.5	5.3	0.0	0.5	0.1
BALD	12.6	11.5	0.0	17.3	9.5	1.1	0.4	0.0
CoreSet	6.1	5.0	3.7	0.0	2.9	0.8	0.0	0.0
GCNAL	10.7	9.0	7.4	15.4	0.0	0.8	0.4	0.1
CDAL	19.5	16.3	15.5	23.1	15.9	0.0	1.4	0.3
BADGE	23.0	18.0	17.8	23.3	17.0	5.1	0.0	0.3
Ours	24.8	21.9	23.1	27.3	21.7	12.4	10.9	0.0
	15.2	13.1	10.9	18.8	11.3	3.1	2.0	0.1

FIGURE 3.5: Pairwise comparison (Ash et al. 2020) of different approaches. Lower values shown at each column reveal the better performances of that AL method across all the experiments. Maximum value of each cell is 30, which represents the number of experimental settings.

the experiments are conducted on HMDB (Kuehne et al. 2011), a widely used dataset consisting of 5,412 training videos belonging to 51 classes representing different actions. For each video, we randomly sampled a video clip with 32 frames of size 224×224 using a temporal stride of 2. Regarding the network architecture, we employed the state-of-the-art Multi-Scale Vision Transformer (MViT) backbone pre-trained on Kinetics-600 (Carreira and Zisserman 2017). Starting with a labelled set consisting of two labelled instances from each class (a total of 102 video clips), we provide each AL method with budget of the varied sizes ($2 \times K$, $4 \times K$, $7 \times K$ and $15 \times K$) in the next AL rounds. At each AL round, we train the network for 50 epochs with a batch size of 8 using AdamW (Loshchilov and Hutter 2018) optimiser with a base learning rate of $1e - 4$ that warms up linearly for the first 30 epochs and then decays to $5e - 5$ for the rest of the iterations using a cosine scheduler (Loshchilov and Hutter 2017). We repeated each experiment twice to cancel out the effect of random selection of the initial labelled set.

Interpolation optimisation: In our approach, we set $\epsilon = \frac{0.2}{\sqrt{D}}$, where D is the

	AL Rounds			
Method	**204**	**408**	**765**	**1530**
MViT (initial accuracy with 102 instances: 50.9±1.2)				
Random	56.7±1.4	64.1±1.2	72.0±1.1	75.3±0.4
Entropy (Wang and Shang 2014)	55.5±0.6	65.5±0.3	70.2±2.0	76.5±0.7
BALD (Gal, Islam, and Ghahramani 2017)	56.7±0.4	65.5±0.6	72.4±1.3	76.6±1.8
CoreSet (Sener and Savarese 2018)	59.3±1.3	65.8±1.2	72.8±1.6	78.5±0.7
GCNAL (Caramalau, Bhattarai, and Kim 2021)	54.9±1.4	63.3±2.2	70.8±1.4	77.0±1.3
CDAL (Agarwal et al. 2020)	60.9±0.1	67.2±0.4	74.6±0.2	78.4±0.5
BADGE (Ash et al. 2020)	60.6±1.3	67.3±0.2	73.2±1.1	**78.7**±0.2
Ours	**62.5**±0.6	**69.4**±0.7	**75.1**±0.3	78.3±0.1

TABLE 3.2: Top-1 test accuracy of various AL methods on HMDB (Kuehne et al. 2011). Values on top of each column reveal the size of the labelled set at the end of each round.

dimentionality of $\boldsymbol{\alpha}$ vector. Considering the norm condition in Eq. 3.7, we relate the scale of ϵ to D to easily utilise the same hyper-parameter across different networks with representations of variable dimensions.

Implementation Details: All the experiments for small datasets were carried out on a NVIDIA GEFORCE GTX 1080 Ti, while for larger datasets we used an NVIDIA QUADRO RTX 8000. It is worth mentioning that for the video experiments, we utilised two NVIDIA V100 GPUs in parallel.

We borrowed the implementations of the baselines from their publicly provided codes. The MLP network we employed in our experiments follows the architecture proposed in (Ash et al. 2020): a two-layer Perceptron with ReLU activations and an embedding dimension of size 256 for image datasets (*i.e.* MNIST and EMNIST). Similarly, we expanded the embedding dimensionality to 1024 for OpenML datasets.

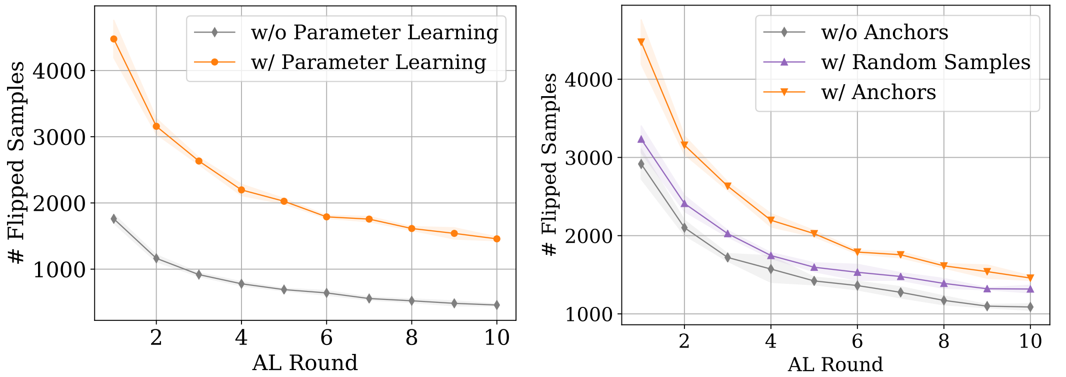
3.5.3 Overall Results

Image and non-image results. In Figure. 3.5 we summarise all the results across various datasets, budget sizes and architectures (30 different settings in total) for image and non-image tasks into a matrix C. While each element $C_{i,j}$ in the matrix reveals in how many experiments the method shown in row i outperforms the one in column j in terms of accuracy of an unseen test set (higher is better for the approach shown in the row). The last row indicates the average number of experiments in which the method in the column has been outperformed by others (lower is better). The maximum value for each cell in the matrix is 30. This matrix clearly shows the superior performance of our approach compared to the baselines. In particular, we outperform CDAL (Agarwal et al. 2020) and BADGE (Ash et al. 2020) in a significant number of experiments (12.3 and 10.6 out of 30, respectively) but ours under-performed in only 0.3 of the times. Generally as shown in the last column, our approach is rarely outperformed (lower than 0.3). In other words, except in 3 AL rounds, for the rest of 282 ones (around 99% of the rounds), our approach is capable of matching or outperforming the best-performing baselines (BADGE and CDAL). In addition to visualising the performance of each AL method in some of the settings in Figure. 3.4, we include the accuracy curves over the unseen test set for all the settings at the end of this chapter.

Video Classification results. Table. 3.2 summarises the results for applying various AL methods for the activity recognition in videos where our approach outperforms the baselines. Interestingly, compared to the Random sampling, we are able to improve the Top-1 test accuracy by more than 5% in the first two AL rounds and 3% in the last ones. This signifies the effectiveness of our proposed approach in reducing the labelling cost which is particularly an obstacle for video classification tasks. Moreover, ALFA-Mix outperforms all other strong baselines with a large margin (more than 2%) in the first three AL rounds. Interestingly, this is similar to what we observe from our experiments on other data types and show the effectiveness of our approach when applied to pre-trained transformers and/or in low-data regimes.

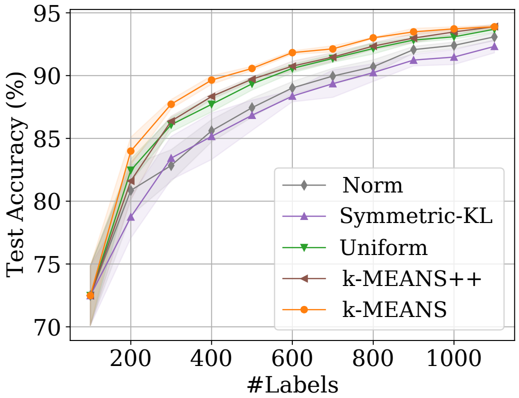
3.5.4 Ablation Study

Learning Ablations. Figure. 3.7 demonstrates the percentage of AL rounds where ALFA-Mix performs better than the baselines considering input data type, network architecture, network parameter initialisation and the budget size. The results indicate our approach, irrespective of other factors, consistently outperforms other AL baselines. Interestingly, when employing pre-trained networks, which is a common



(A) Number of unlabelled samples whose predictions flip with and without learning the interpolation parameter $\boldsymbol{\alpha}$.

(B) The impact of anchors on identifying samples whose labels flip during the interpolation.



(C) Diversity impact of the sample selection from the candidate set ($\mathcal{I}$). k-MEANS is our proposed full model.

FIGURE 3.6: Ablations of our AL approach. Experiments are conducted on MNIST datasets using an MLP model and a small AL budget.

practice for transferring learnt representations to new tasks, ALFA-Mix 99% of occasions assists the model to generalise better than random sampling. Additionally, in these settings, our approach surpasses the strongest baselines (CDAL and BADGE) in more than 40% of the rounds. Above all, using Vision Transformer networks pre-trained in a self-supervised manner, ALFA-Mix not only outperforms Random, BALD, CoreSet and GCNAL in all the AL settings, it also significantly improves on BADGE and CDAL in 60% and 43% of the rounds respectively.

Interestingly, we observe a significant advantage from our proposed AL method when it is applied on small budget setting (Figure. 3.7). In fact, the test performance of our approach exceeds BADGE (the best performing baseline) in 46% of the small budget experiments. Moreover, we observe a more evident gap between our approach

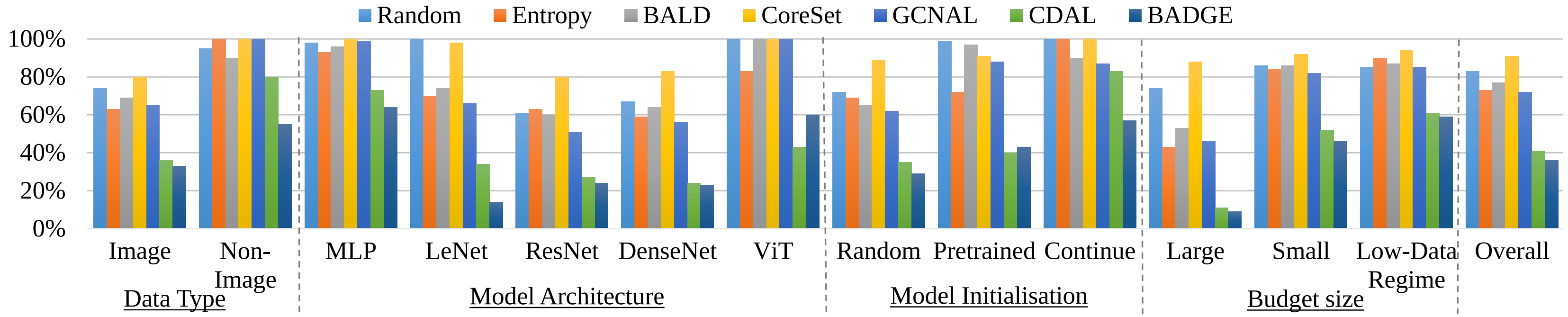


FIGURE 3.7: A summary of the performance of our proposed AL method (ALFA-Mix) compared with state-of-the-art across all the 30 settings considered. Each bar represents the percentage of AL rounds in which our approach outperforms others (lower indicates stronger baseline). It is worth noting that our approach (ALFA-Mix) under-performs others in close to zero cases.

and others when it comes to AL in low-data regime. For that, we consider the performance in the first 5 rounds of AL using a small budget; *i.e.* starting from $10 \times K$ randomly selected labelled samples, each method queries for the maximum of $50 \times K$ unlabelled samples overall during 5 AL iterations. Figure. 3.7 demonstrates the dominance of our approach in this setting as it eclipses all other approaches in at least 60% of the experiments. When using a large budget, our approach is able to slightly surpass BADGE which previously has shown great success in this setting.

Diversification. Figure. 3.6c illustrates the effectiveness of the batch diversification on selecting final instances from the set of samples whose predictions have been changed ($\mathcal{I}$) during the interpolation process. In addition to *uniformly* sampling instances from the candidate set, we consider two heuristics: (1) the *norm* of the interpolation parameter $||\boldsymbol{\alpha}||_2$ in which a lower norm indicates with smaller intervention the model changed prediction; and, (2) the *symmetric KL-Divergence* between the predicted label distributions from the unlabelled instance $p(y|\boldsymbol{z}^u; \boldsymbol{\theta}_c)$ and that of the interpolated variant $p(y|\tilde{\boldsymbol{z}}_{\boldsymbol{\alpha}}; \boldsymbol{\theta}_c)$. The latter evaluates the distributions change in the output (*i.e.* prefers samples with highest values of symmetric KL-Divergence). Interestingly, both heuristics show poor performances even in comparison with the uniform selection from the candidate set. While this highlights how hard the candidate selection could be, one explanation is that these approaches might use a considerable proportion of the budget on samples that reside in a small region of the space. Consequently, the selected batch does not carry the whole information obtained through the interpolation process.

In addition to the heuristic measures, we considered k-MEANS++, a simpler variation of k-MEANS that has shown better performance in (Ash et al. 2020), as another contender. In contrast to what found in (Ash et al. 2020), in our experiments, k-MEANS outperforms k-MEANS++ considerably as it better representations found using interpolation.

Learning the Interpolation Parameter. As it is evident in Figure. 3.6a, skipping the learning process for the interpolation parameter $\boldsymbol{\alpha}$ (see section 3.4.3) significantly

Method	**Time** (seconds)	
	MNIST (MLP)	**SVHN** (DenseNet)
Entropy (Wang and Shang 2014)	1±0	169±44
BALD (Gal, Islam, and Ghahramani 2017)	16±4	1723±445
Coreset (Sener and Savarese 2018)	7±2	185±49
DFAL (Ducoffe and Precioso 2018)	242±69	–
GCNAL (Caramalau, Bhattarai, and Kim 2021)	12±4	187±65
CDAL (Agarwal et al. 2020)	5±2	179±52
BADGE (Ash et al. 2020)	50±13	523±135
Ours	5±7	210±50

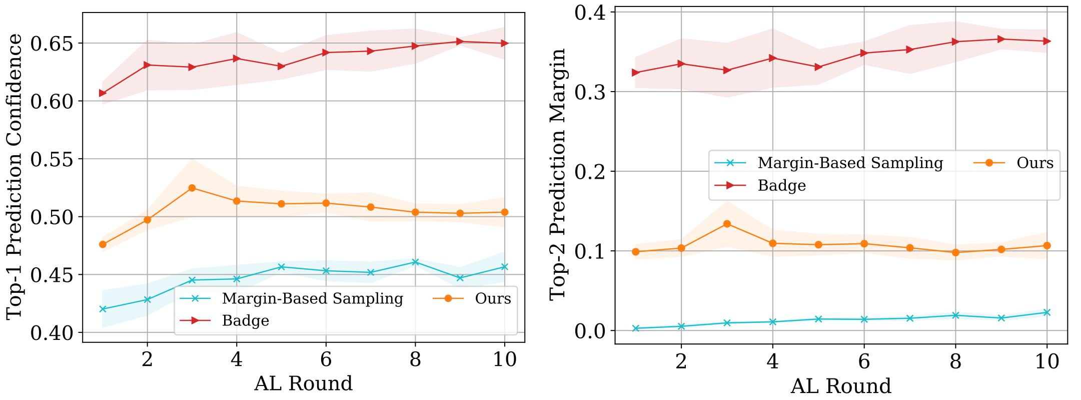
TABLE 3.3: Label acquisition run times of different methods. Our approach is significantly faster than BADGE and about 50x quicker than its Adversarial counterpart.

reduces the number of samples chosen in the candidate set. This can have detrimental consequence on the diversity of samples that are selected during the clustering.

Anchors. Figure. 3.6b shows the impact of using different anchors $Z^\star$. Evidently, the proposed method based on anchors outperforms other plausible alternatives including picking random samples from the labelled set and removing $z^\star$ during the interpolation. The latter resembles adding noise to the sample and is similar to applying adversarial attack in the latent space.

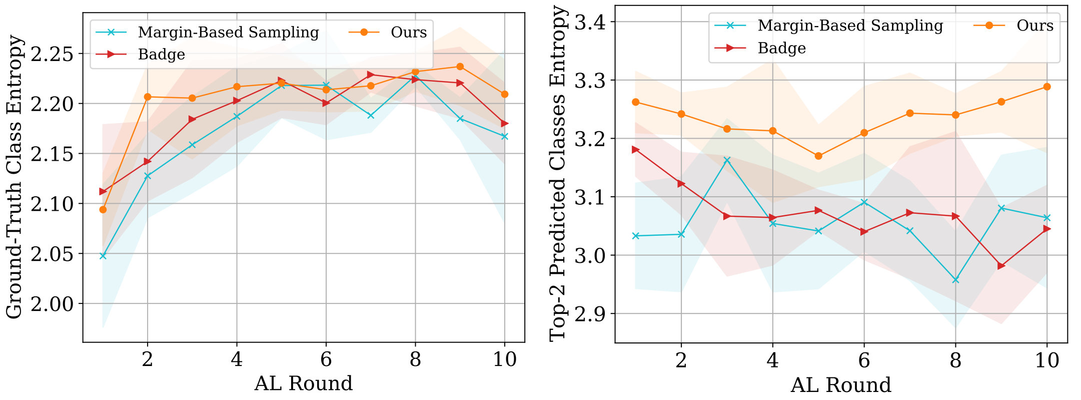
Acquisition Time. We measured the time required to choose instances for labelling during each AL round. As demonstrated in Table 3.3, using a simple MLP network or a deep DenseNet-121, our approach performs competitive with the fastest baselines. This is mainly because of the fact that we only back-propagates to a latent representation layer (not the whole network). Additionally, our approaches reduces the time required for BADGE (the best performing baseline) by a factor of more than 2 when applied to datasets with a small number of classes. We should note that running BADGE on large-scale datasets with numerous classes requires a considerable time and computing resources. The main reason is the large dimensionality of the gradient embedding in tasks with large number of classes and instances. More importantly, Table 3.3 shows the time needed for DFAL method for MNIST dataset, which makes it impossible to apply to deep models and large datasets in a reasonable time.

Sampling Diversity and Uncertainty To have a better understanding with regards to the effectiveness of our approach in selecting an uncertain and diverse set of samples



(A) The confidence of the predicted Top-1 class.

(B) The margin (distance) between the predicted probabilities of the Top-2 classes.



(C) The entropy of the revealed ground-truth labels.

(D) The entropy of the predicted Top-2 classes (ignoring the order of them).

FIGURE 3.8: Uncertainty and diversity of the selected samples for labelling. All experiments are done on MNIST dataset using LeNet-5 model and a small budget of size 100.

for labelling, we compare some characteristics of the selected batch of instances at each AL round comparing our method with those from BADGE Ash et al. 2020 and Margin-Based Sampling[**] Roth and Small 2006 (Figure 3.8).

Comparing the confidence and Top-2 prediction margins of the selected unlabelled samples, depicted in Figures 3.8a and 3.8b respectively, we can see that the uncertainty level of the selected samples by our method is closer to the highest possible value in comparison to BADGE sampling. Please note that in contrast to what Margin-Based Sampling is doing, we do not explicitly enforce our approach to select samples close to the decision boundaries. On the other hand, considering the higher entropy values

---

[**] Margin-Based Sampling is another AL method based on uncertainty. It selects samples with the lowest distance between the predicted probabilities for the Top-2 classes (called margin). It should be noted that BADGE has shown significantly better performance compared to Margin-Based Sampling in prior works Ash et al. 2020.

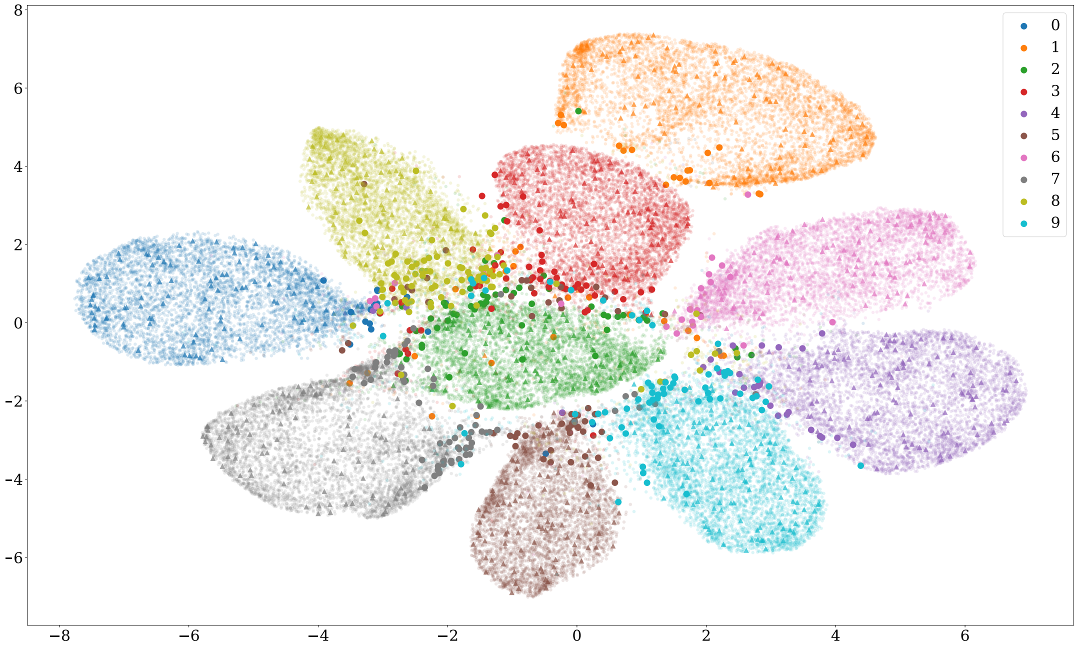
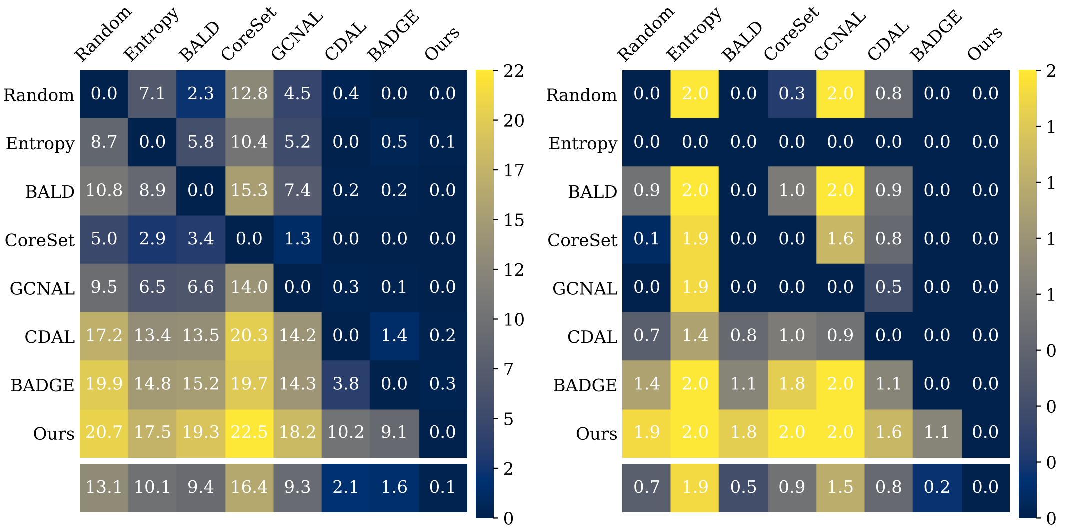


FIGURE 3.9: The t-SNE visualisation of the sample selection of our proposed method on MNIST dataset using LeNet-5. The model is trained based on 500 random labelled set (shown as triangles) and is provided with a budget of size 500 to (depicted as bold circles).

in the ground-truth labels of the selected set and their Top-2 predicted classes, we can realise the capability of our proposed method in selecting a diverse set of unlabelled samples in terms of their true class labels and their position with regard to the decision boundaries. All in all, as depicted in Fig. 3.9, our method is able to exploit both uncertainty and diversity concepts to select a diverse set of samples that lie close to decision boundaries, which leads to significantly higher performances.

More Ablations In addition to providing the percentage with which our approach outperforms others in each setting (Figure. 3.7), we report the pairwise comparison of all the AL methods across various choices of data (Fig. 3.10), budget size (Fig. 3.12), model architecture (Fig. 3.11) and network initialisation method (Fig. 3.13. Further, in Figure 3.12c, we provide the pairwise comparisons in low-data regimes. Considering the values in the rows and columns corresponding to our approach, we can infer that our approach consistently outperforms all other baselines regardless of the architecture, dataset selection, network initialisation and budget size and is rarely beaten by others.



	Random	Entropy	BALD	CoreSet	GCNAL	CDAL	BADGE	Ours
Random	0.0	7.1	2.3	12.8	4.5	0.4	0.0	0.0
Entropy	8.7	0.0	5.8	10.4	5.2	0.0	0.5	0.1
BALD	10.8	8.9	0.0	15.3	7.4	0.2	0.2	0.0
CoreSet	5.0	2.9	3.4	0.0	1.3	0.0	0.0	0.0
GCNAL	9.5	6.5	6.6	14.0	0.0	0.3	0.1	0.0
CDAL	17.2	13.4	13.5	20.3	14.2	0.0	1.4	0.2
BADGE	19.9	14.8	15.2	19.7	14.3	3.8	0.0	0.3
Ours	20.7	17.5	19.3	22.5	18.2	10.2	9.1	0.0
	13.1	10.1	9.4	16.4	9.3	2.1	1.6	0.1

(A) Image (maximum value: 28)

	Random	Entropy	BALD	CoreSet	GCNAL	CDAL	BADGE	Ours
Random	0.0	2.0	0.0	0.3	2.0	0.8	0.0	0.0
Entropy	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
BALD	0.9	2.0	0.0	1.0	2.0	0.9	0.0	0.0
CoreSet	0.1	1.9	0.0	0.0	1.6	0.8	0.0	0.0
GCNAL	0.0	1.9	0.0	0.0	0.0	0.5	0.0	0.0
CDAL	0.7	1.4	0.8	1.0	0.9	0.0	0.0	0.0
BADGE	1.4	2.0	1.1	1.8	2.0	1.1	0.0	0.0
Ours	1.9	2.0	1.8	2.0	2.0	1.6	1.1	0.0
	0.7	1.9	0.5	0.9	1.5	0.8	0.2	0.0

(B) OpenML (maximum value: 2)

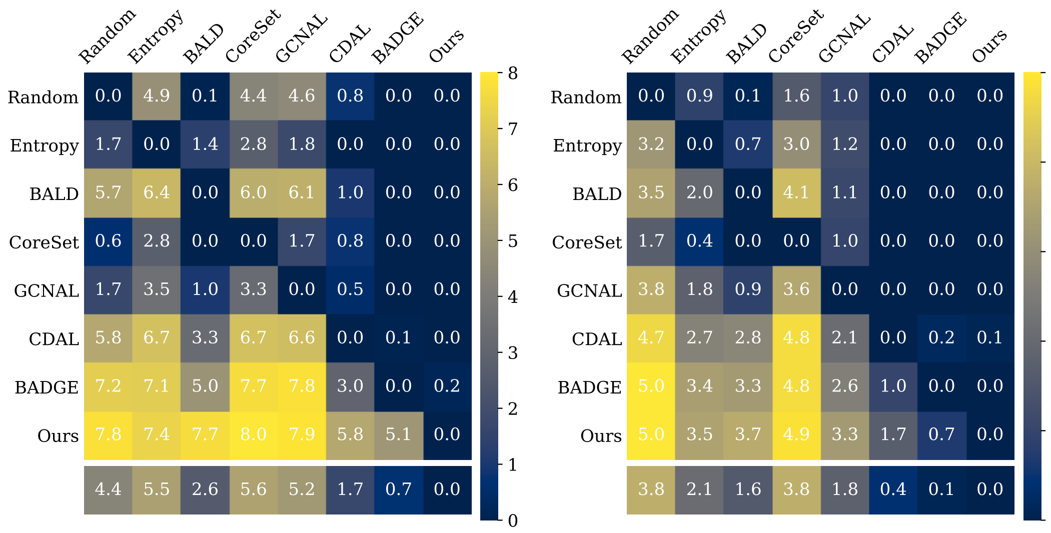
FIGURE 3.10: Pairwise comparison of different AL approaches based on the type of data. The maximum value of each cell for each setting is also provided in the captions.

3.6 Conclusions and Limitations

In this chapter, we proposed a simple AL method based on the interpolation between labelled and unlabelled samples. We effectively applied ALFA-Mix to a wide variety of image, non-image and video datasets and demonstrate its state-of-the-art results across various settings. Attractively, when the labelled set is small and the budget is limited, our approach is able to gain the most performance boost–it surpassed all other baselines in around 60% of all evaluated rounds.

Further, the feature representations are not generally disentangled (Locatello et al. 2020; Engelcke et al. 2020) and interpolation in the high dimensional space may yield representations for unexpected inputs. Nevertheless, our approach indicates such interpolations highlight reasonable variations in the input that may otherwise remain unexplored. For future, we consider using disentangled representations to explore novel factors of variations.

Limitations: AL consciously selects a small subset of a large pool of unlabelled samples to be labelled and used to train a model. AL will be essential in catastrophes, like pandemics, where the time to reach a model at a particular level of accuracy becomes vital and would directly impact the lives of people. In spite of that, its a common practice to evaluate AL in a simulated environment mainly due to financial

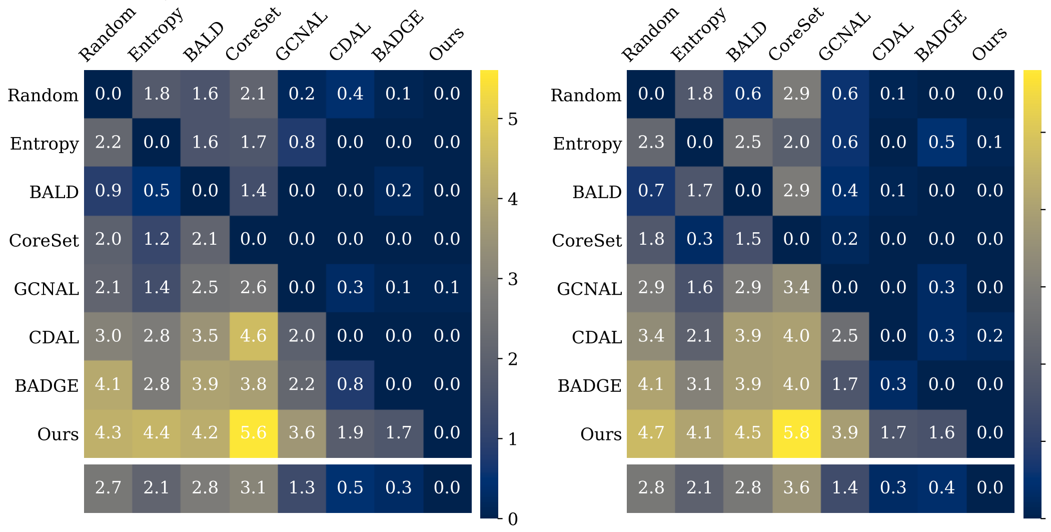


	Random	Entropy	BALD	CoreSet	GCNAL	CDAL	BADGE	Ours
Random	0.0	4.9	0.1	4.4	4.6	0.8	0.0	0.0
Entropy	1.7	0.0	1.4	2.8	1.8	0.0	0.0	0.0
BALD	5.7	6.4	0.0	6.0	6.1	1.0	0.0	0.0
CoreSet	0.6	2.8	0.0	0.0	1.7	0.8	0.0	0.0
GCNAL	1.7	3.5	1.0	3.3	0.0	0.5	0.0	0.0
CDAL	5.8	6.7	3.3	6.7	6.6	0.0	0.1	0.0
BADGE	7.2	7.1	5.0	7.7	7.8	3.0	0.0	0.2
Ours	7.8	7.4	7.7	8.0	7.9	5.8	5.1	0.0
	4.4	5.5	2.6	5.6	5.2	1.7	0.7	0.0

	Random	Entropy	BALD	CoreSet	GCNAL	CDAL	BADGE	Ours
Random	0.0	0.9	0.1	1.6	1.0	0.0	0.0	0.0
Entropy	3.2	0.0	0.7	3.0	1.2	0.0	0.0	0.0
BALD	3.5	2.0	0.0	4.1	1.1	0.0	0.0	0.0
CoreSet	1.7	0.4	0.0	0.0	1.0	0.0	0.0	0.0
GCNAL	3.8	1.8	0.9	3.6	0.0	0.0	0.0	0.0
CDAL	4.7	2.7	2.8	4.8	2.1	0.0	0.2	0.1
BADGE	5.0	3.4	3.3	4.8	2.6	1.0	0.0	0.0
Ours	5.0	3.5	3.7	4.9	3.3	1.7	0.7	0.0
	3.8	2.1	1.6	3.8	1.8	0.4	0.1	0.0

(A) Two-layer MLP (maximum value: 8)

(B) LeNet-5 (maximum value: 5)

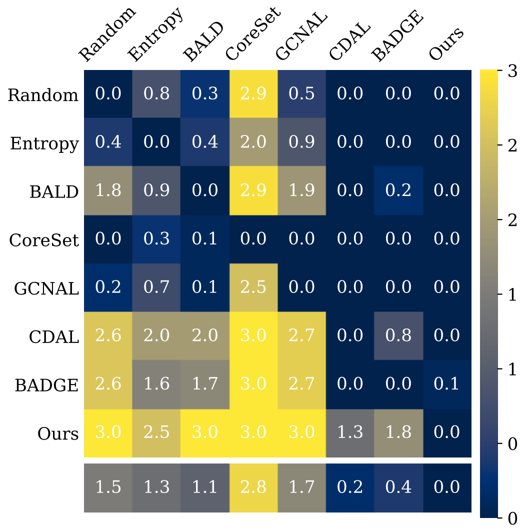


	Random	Entropy	BALD	CoreSet	GCNAL	CDAL	BADGE	Ours
Random	0.0	1.8	1.6	2.1	0.2	0.4	0.1	0.0
Entropy	2.2	0.0	1.6	1.7	0.8	0.0	0.0	0.0
BALD	0.9	0.5	0.0	1.4	0.0	0.0	0.2	0.0
CoreSet	2.0	1.2	2.1	0.0	0.0	0.0	0.0	0.0
GCNAL	2.1	1.4	2.5	2.6	0.0	0.3	0.1	0.1
CDAL	3.0	2.8	3.5	4.6	2.0	0.0	0.0	0.0
BADGE	4.1	2.8	3.9	3.8	2.2	0.8	0.0	0.0
Ours	4.3	4.4	4.2	5.6	3.6	1.9	1.7	0.0
	2.7	2.1	2.8	3.1	1.3	0.5	0.3	0.0

	Random	Entropy	BALD	CoreSet	GCNAL	CDAL	BADGE	Ours
Random	0.0	1.8	0.6	2.9	0.6	0.1	0.0	0.0
Entropy	2.3	0.0	2.5	2.0	0.6	0.0	0.5	0.1
BALD	0.7	1.7	0.0	2.9	0.4	0.1	0.0	0.0
CoreSet	1.8	0.3	1.5	0.0	0.2	0.0	0.0	0.0
GCNAL	2.9	1.6	2.9	3.4	0.0	0.0	0.3	0.0
CDAL	3.4	2.1	3.9	4.0	2.5	0.0	0.3	0.2
BADGE	4.1	3.1	3.9	4.0	1.7	0.3	0.0	0.0
Ours	4.7	4.1	4.5	5.8	3.9	1.7	1.6	0.0
	2.8	2.1	2.8	3.6	1.4	0.3	0.4	0.0

(C) ResNet-18 (maximum value: 7)

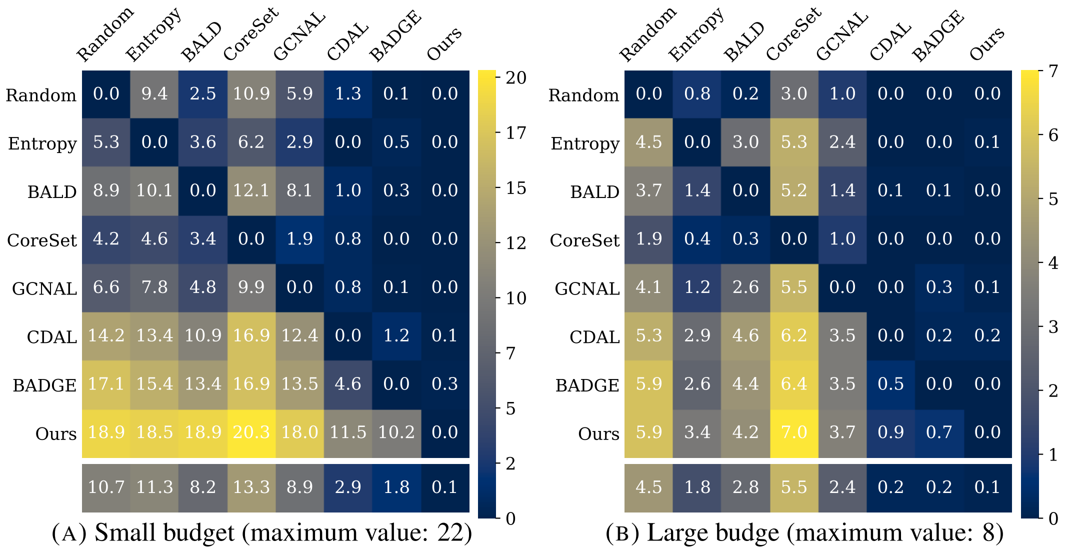
(D) DenseNet-121 (maximum value: 7)



	Random	Entropy	BALD	CoreSet	GCNAL	CDAL	BADGE	Ours
Random	0.0	0.8	0.3	2.9	0.5	0.0	0.0	0.0
Entropy	0.4	0.0	0.4	2.0	0.9	0.0	0.0	0.0
BALD	1.8	0.9	0.0	2.9	1.9	0.0	0.2	0.0
CoreSet	0.0	0.3	0.1	0.0	0.0	0.0	0.0	0.0
GCNAL	0.2	0.7	0.1	2.5	0.0	0.0	0.0	0.0
CDAL	2.6	2.0	2.0	3.0	2.7	0.0	0.8	0.0
BADGE	2.6	1.6	1.7	3.0	2.7	0.0	0.0	0.1
Ours	3.0	2.5	3.0	3.0	3.0	1.3	1.8	0.0
	1.5	1.3	1.1	2.8	1.7	0.2	0.4	0.0

(E) ViT (maximum value: 3)

FIGURE 3.11: Pairwise comparison of different AL approaches based on different model architectures. The maximum value of each cell for each setting is also provided in the captions.

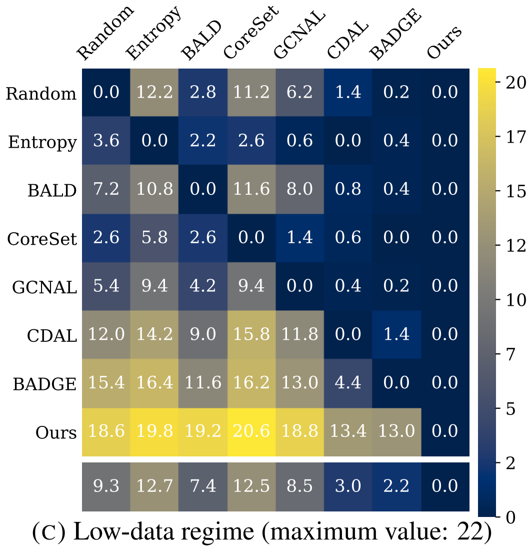


	Random	Entropy	BALD	CoreSet	GCNAL	CDAL	BADGE	Ours
Random	0.0	9.4	2.5	10.9	5.9	1.3	0.1	0.0
Entropy	5.3	0.0	3.6	6.2	2.9	0.0	0.5	0.0
BALD	8.9	10.1	0.0	12.1	8.1	1.0	0.3	0.0
CoreSet	4.2	4.6	3.4	0.0	1.9	0.8	0.0	0.0
GCNAL	6.6	7.8	4.8	9.9	0.0	0.8	0.1	0.0
CDAL	14.2	13.4	10.9	16.9	12.4	0.0	1.2	0.1
BADGE	17.1	15.4	13.4	16.9	13.5	4.6	0.0	0.3
Ours	18.9	18.5	18.9	20.3	18.0	11.5	10.2	0.0
	10.7	11.3	8.2	13.3	8.9	2.9	1.8	0.1

(A) Small budget (maximum value: 22)

	Random	Entropy	BALD	CoreSet	GCNAL	CDAL	BADGE	Ours
Random	0.0	0.8	0.2	3.0	1.0	0.0	0.0	0.0
Entropy	4.5	0.0	3.0	5.3	2.4	0.0	0.0	0.1
BALD	3.7	1.4	0.0	5.2	1.4	0.1	0.1	0.0
CoreSet	1.9	0.4	0.3	0.0	1.0	0.0	0.0	0.0
GCNAL	4.1	1.2	2.6	5.5	0.0	0.0	0.3	0.1
CDAL	5.3	2.9	4.6	6.2	3.5	0.0	0.2	0.2
BADGE	5.9	2.6	4.4	6.4	3.5	0.5	0.0	0.0
Ours	5.9	3.4	4.2	7.0	3.7	0.9	0.7	0.0
	4.5	1.8	2.8	5.5	2.4	0.2	0.2	0.1

(B) Large budge (maximum value: 8)

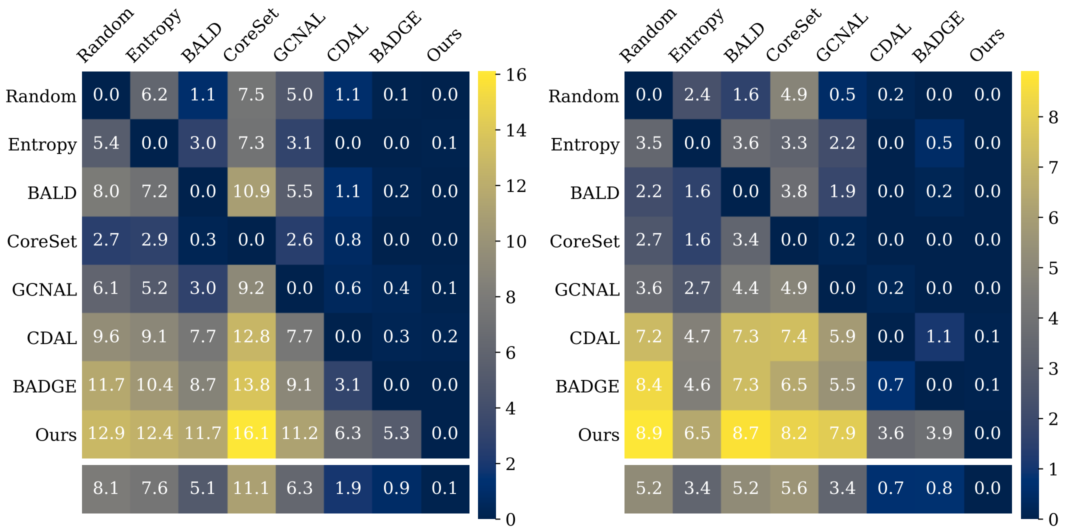


	Random	Entropy	BALD	CoreSet	GCNAL	CDAL	BADGE	Ours
Random	0.0	12.2	2.8	11.2	6.2	1.4	0.2	0.0
Entropy	3.6	0.0	2.2	2.6	0.6	0.0	0.4	0.0
BALD	7.2	10.8	0.0	11.6	8.0	0.8	0.4	0.0
CoreSet	2.6	5.8	2.6	0.0	1.4	0.6	0.0	0.0
GCNAL	5.4	9.4	4.2	9.4	0.0	0.4	0.2	0.0
CDAL	12.0	14.2	9.0	15.8	11.8	0.0	1.4	0.0
BADGE	15.4	16.4	11.6	16.2	13.0	4.4	0.0	0.0
Ours	18.6	19.8	19.2	20.6	18.8	13.4	13.0	0.0
	9.3	12.7	7.4	12.5	8.5	3.0	2.2	0.0

(C) Low-data regime (maximum value: 22)

FIGURE 3.12: Pairwise comparison of different AL approaches based on different sizes of budget. The maximum value of each cell for each setting is also provided in the captions.

constraints. However, AL community at large and our approach in particular could heavily benefit from real-world evaluations.

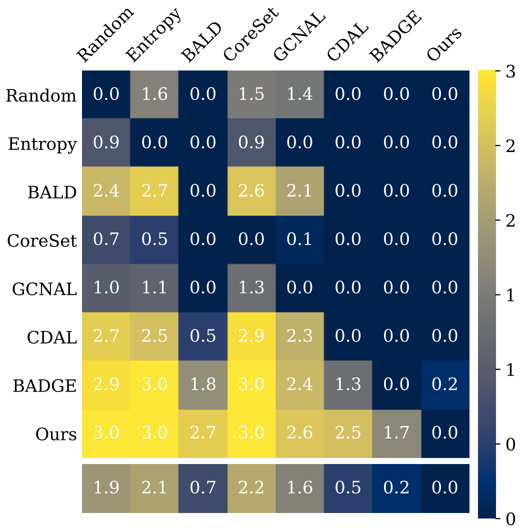


	Random	Entropy	BALD	CoreSet	GCNAL	CDAL	BADGE	Ours
Random	0.0	6.2	1.1	7.5	5.0	1.1	0.1	0.0
Entropy	5.4	0.0	3.0	7.3	3.1	0.0	0.0	0.1
BALD	8.0	7.2	0.0	10.9	5.5	1.1	0.2	0.0
CoreSet	2.7	2.9	0.3	0.0	2.6	0.8	0.0	0.0
GCNAL	6.1	5.2	3.0	9.2	0.0	0.6	0.4	0.1
CDAL	9.6	9.1	7.7	12.8	7.7	0.0	0.3	0.2
BADGE	11.7	10.4	8.7	13.8	9.1	3.1	0.0	0.0
Ours	12.9	12.4	11.7	16.1	11.2	6.3	5.3	0.0
	8.1	7.6	5.1	11.1	6.3	1.9	0.9	0.1

(A) Random (maximum value: 18)

	Random	Entropy	BALD	CoreSet	GCNAL	CDAL	BADGE	Ours
Random	0.0	2.4	1.6	4.9	0.5	0.2	0.0	0.0
Entropy	3.5	0.0	3.6	3.3	2.2	0.0	0.5	0.0
BALD	2.2	1.6	0.0	3.8	1.9	0.0	0.2	0.0
CoreSet	2.7	1.6	3.4	0.0	0.2	0.0	0.0	0.0
GCNAL	3.6	2.7	4.4	4.9	0.0	0.2	0.0	0.0
CDAL	7.2	4.7	7.3	7.4	5.9	0.0	1.1	0.1
BADGE	8.4	4.6	7.3	6.5	5.5	0.7	0.0	0.1
Ours	8.9	6.5	8.7	8.2	7.9	3.6	3.9	0.0
	5.2	3.4	5.2	5.6	3.4	0.7	0.8	0.0

(B) Pre-Training (maximum value: 9)



	Random	Entropy	BALD	CoreSet	GCNAL	CDAL	BADGE	Ours
Random	0.0	1.6	0.0	1.5	1.4	0.0	0.0	0.0
Entropy	0.9	0.0	0.0	0.9	0.0	0.0	0.0	0.0
BALD	2.4	2.7	0.0	2.6	2.1	0.0	0.0	0.0
CoreSet	0.7	0.5	0.0	0.0	0.1	0.0	0.0	0.0
GCNAL	1.0	1.1	0.0	1.3	0.0	0.0	0.0	0.0
CDAL	2.7	2.5	0.5	2.9	2.3	0.0	0.0	0.0
BADGE	2.9	3.0	1.8	3.0	2.4	1.3	0.0	0.2
Ours	3.0	3.0	2.7	3.0	2.6	2.5	1.7	0.0
	1.9	2.1	0.7	2.2	1.6	0.5	0.2	0.0

(C) Continue (maximum value: 3)

FIGURE 3.13: Pairwise comparison of different AL approaches based on different sizes of budget. The maximum value of each cell for each setting is also provided in the captions.

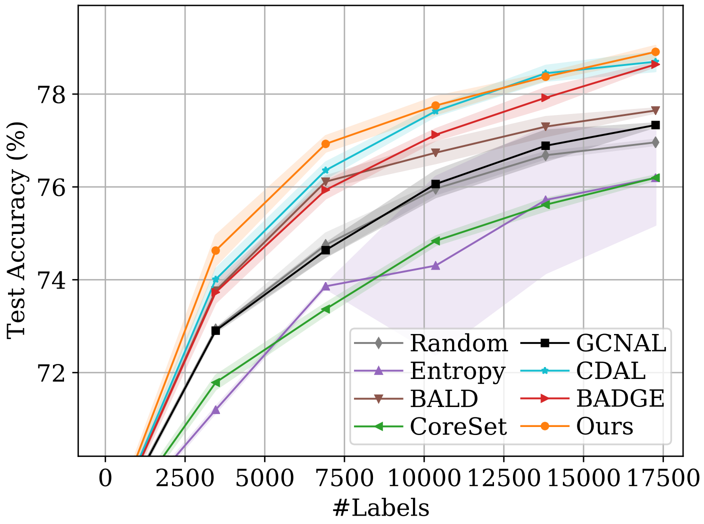


FIGURE 3.14: Small Budget, ViT-Base, DomainNet-Real

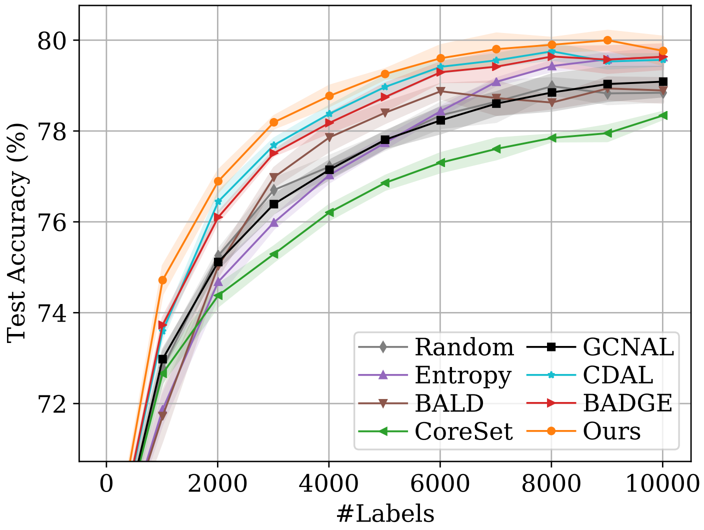


FIGURE 3.15: Small Budget, ViT-Small, Mini-ImageNet

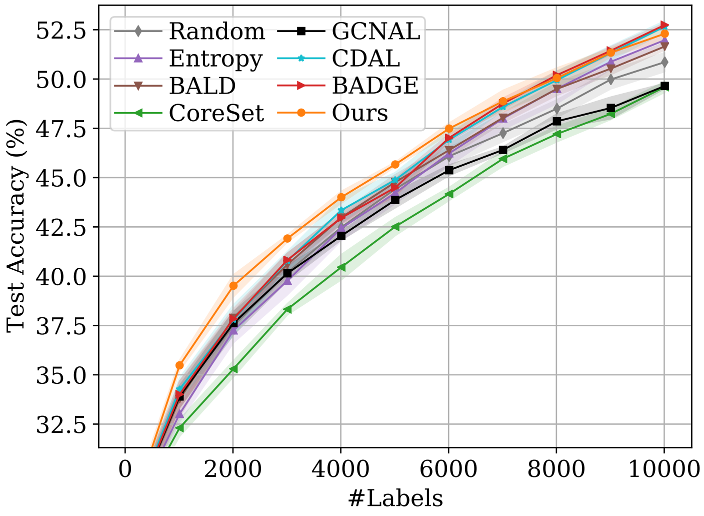


FIGURE 3.16: Small Budget, ViT-Small, CIFAR100

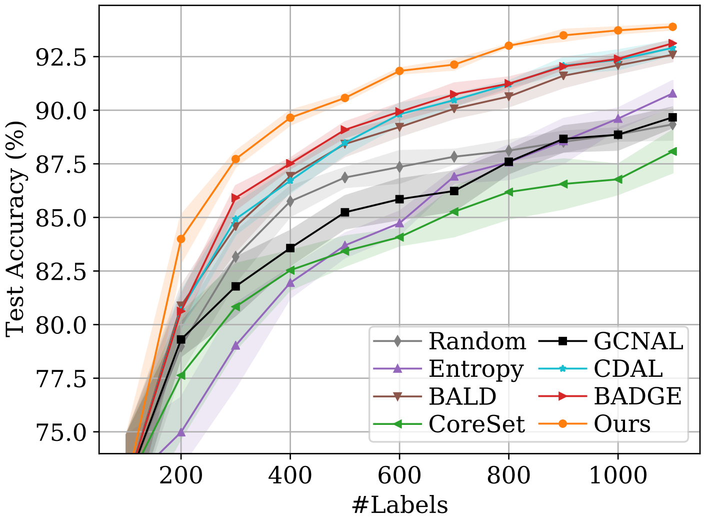


FIGURE 3.17: Small Budget, MLP, MNIST

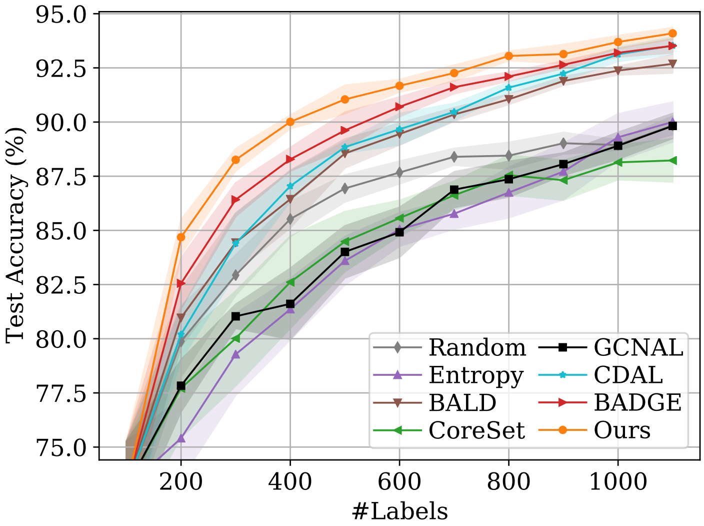


FIGURE 3.18: Small Budget, MLP, MNIST, Continue

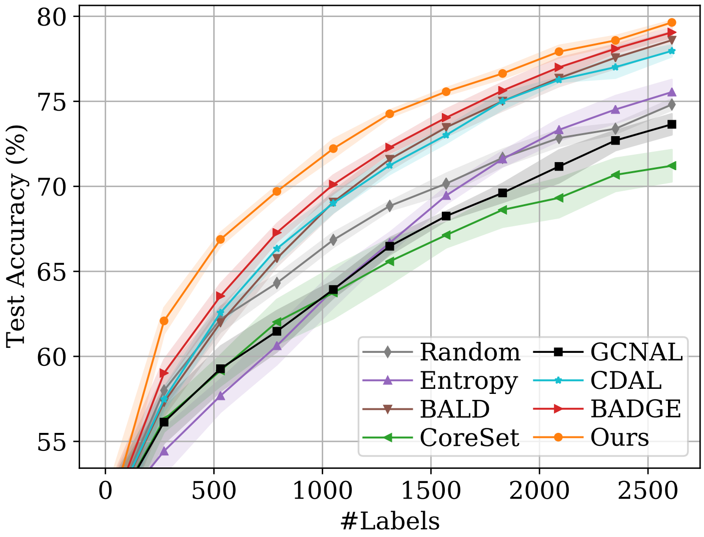


FIGURE 3.19: Small Budget, MLP, EMNIST

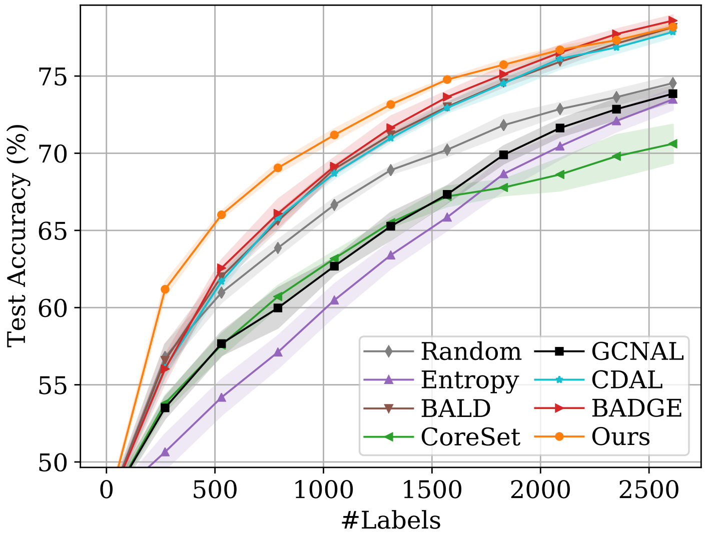


FIGURE 3.20: Small Budget, MLP, EMNIST, Continue

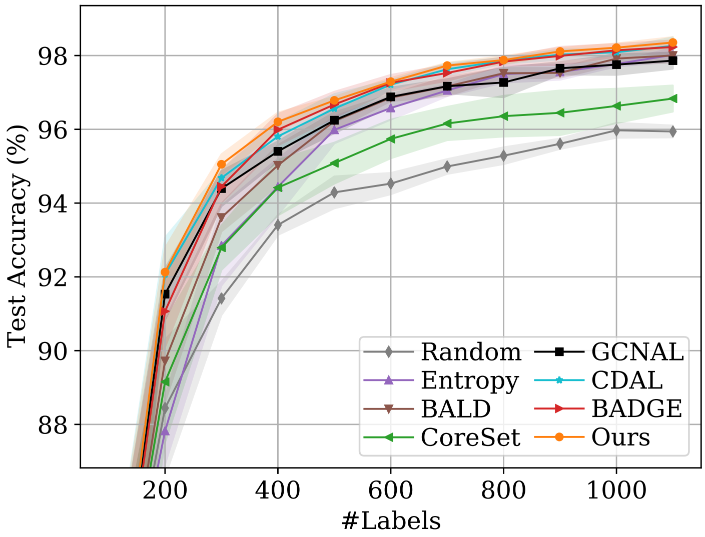


FIGURE 3.21: Small Budget, LeNet-5, MNIST

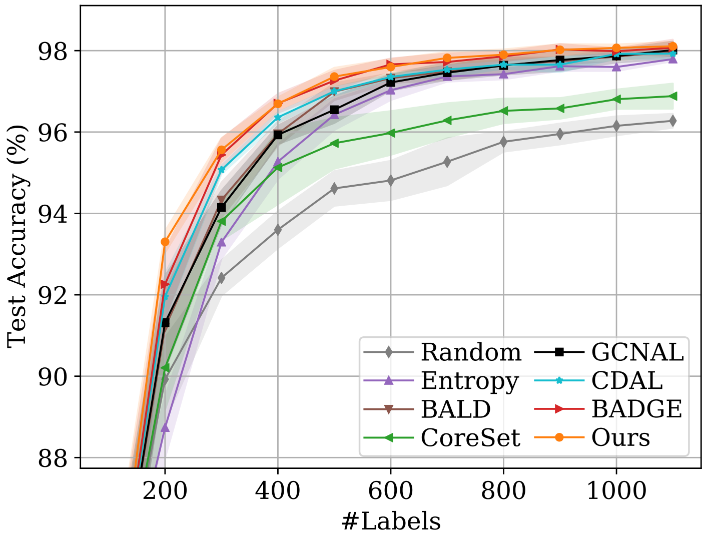


FIGURE 3.22: Small Budget, LeNet-5, MNIST, Continue

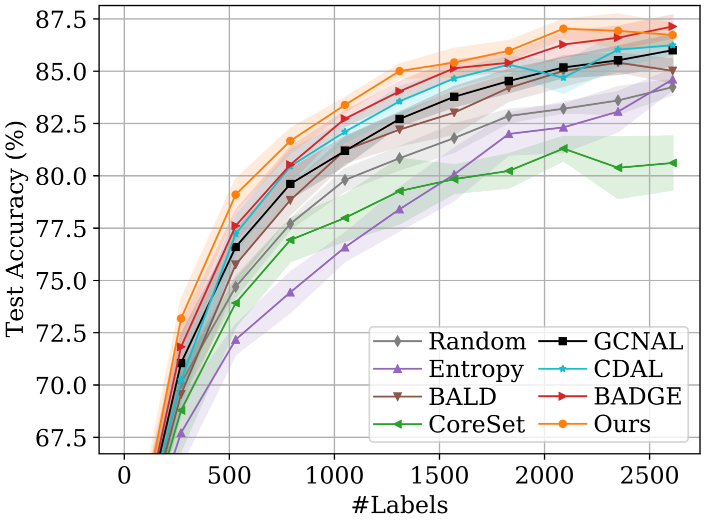


FIGURE 3.23: Small Budget, LeNet-5, EMNIST

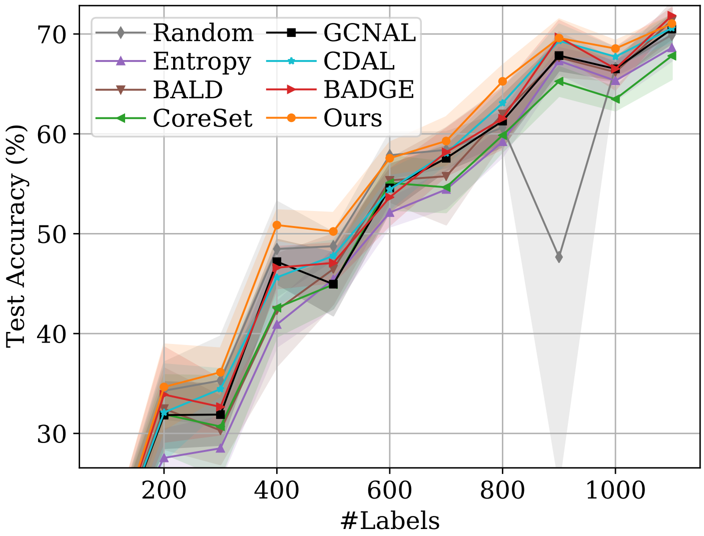


FIGURE 3.24: Small Budget-ResNet-18, SVHN

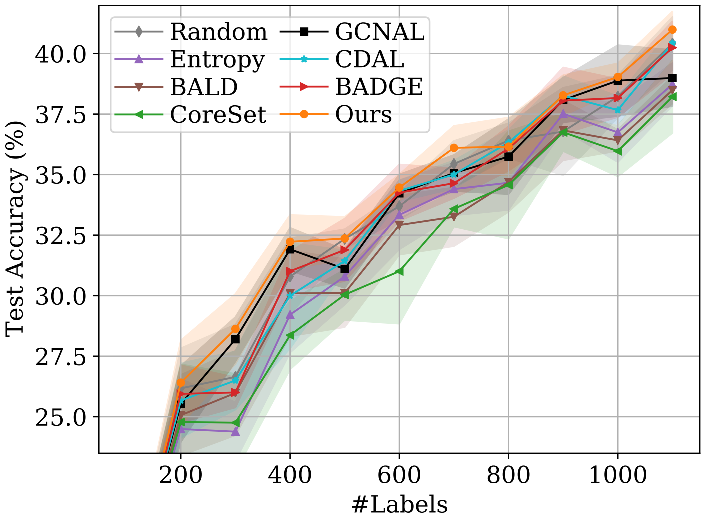


FIGURE 3.25: Small Budget, ResNet-18, CIFAR10

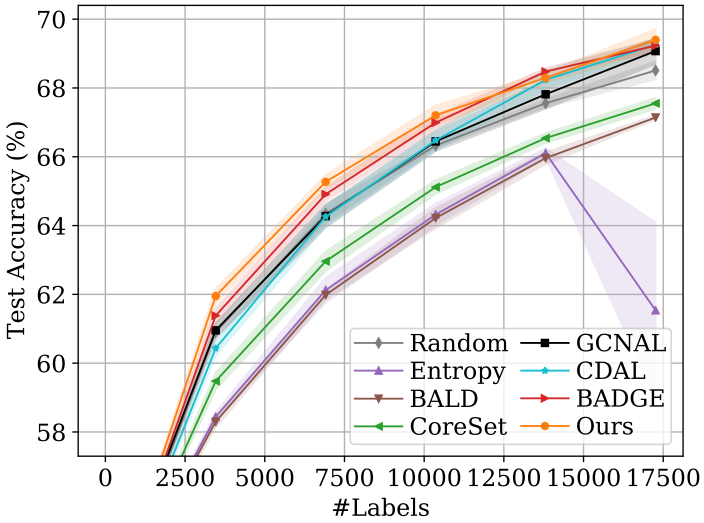


FIGURE 3.26: Small Budget, ResNet-18, DomainNet-Real

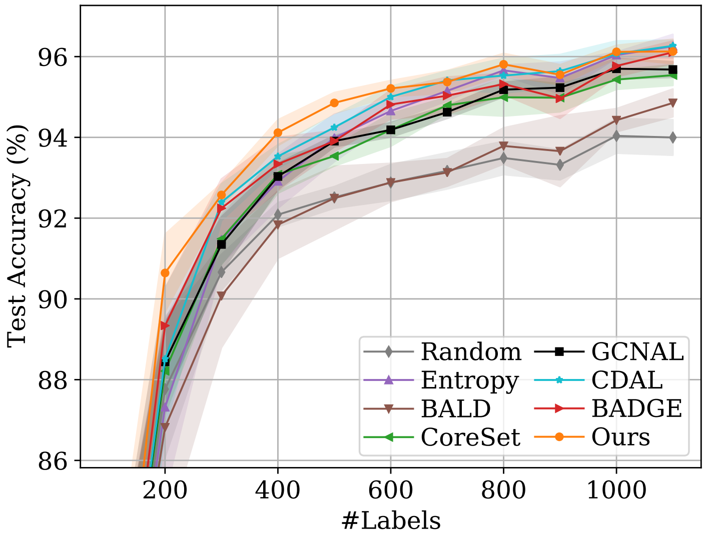


FIGURE 3.27: Small Budget, ResNet-18, DomainNet-Real-10

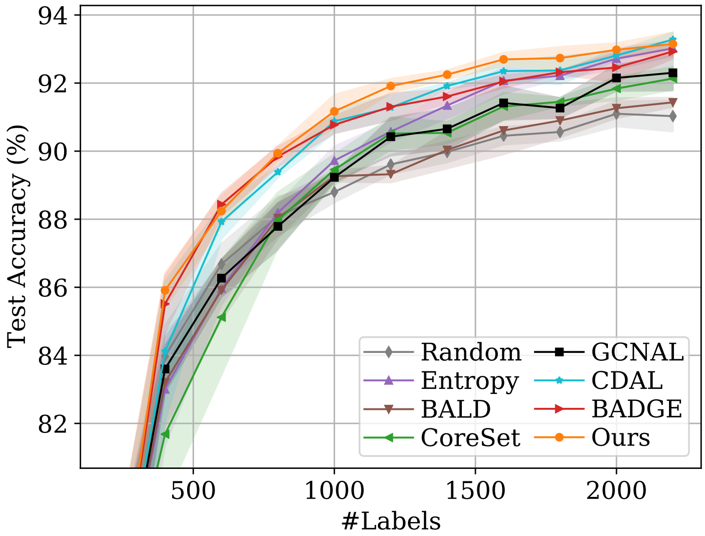


FIGURE 3.28: Small Budget, ResNet-18, DomainNet-Real-20

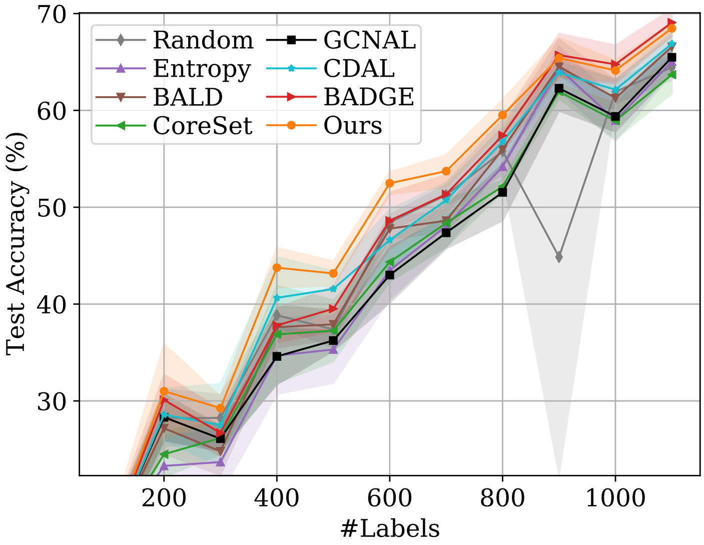


FIGURE 3.29: Small Budget, DenseNet-121, SVHN

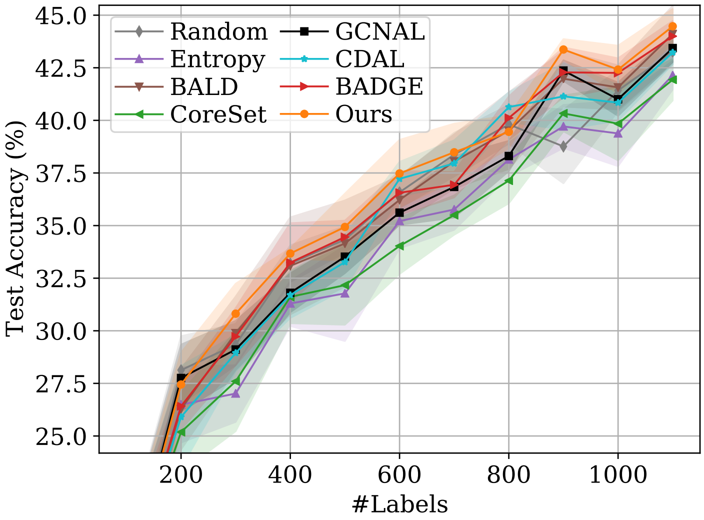


FIGURE 3.30: Small Budget, DenseNet-121, CIFAR10

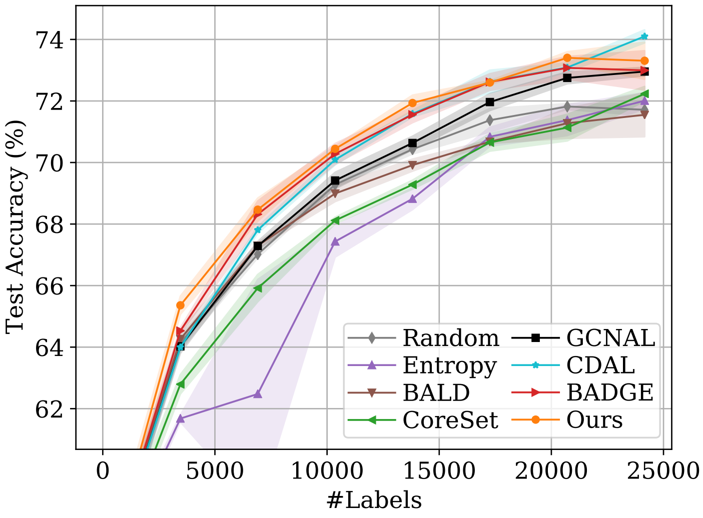


FIGURE 3.31: Small Budget, DenseNet-121, DomainNet-Real

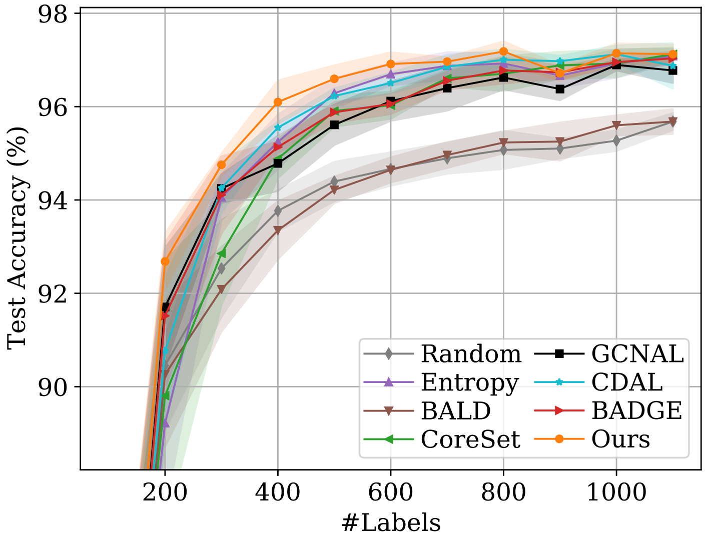


FIGURE 3.32: Small Budget, DenseNet-121, DomainNet-Real-10

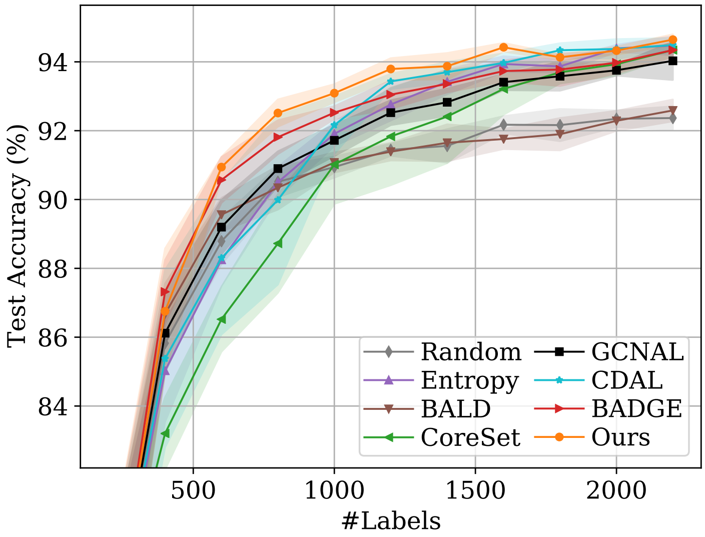


FIGURE 3.33: Small Budget, DenseNet-121, DomainNet-Real-20

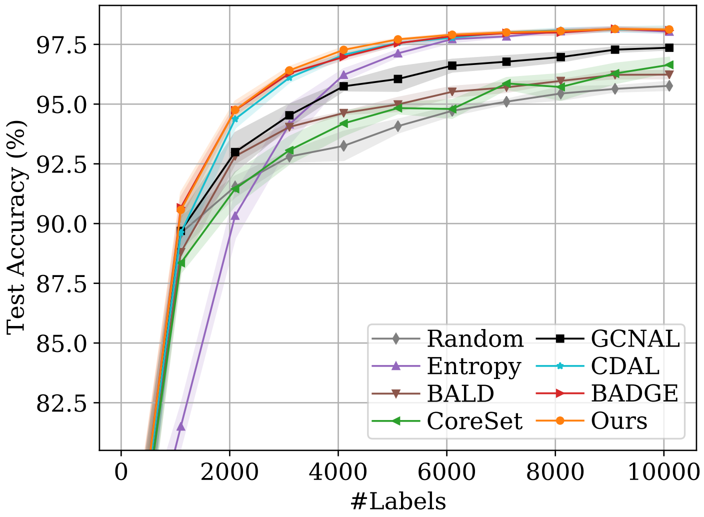


FIGURE 3.34: Large Budget, MLP, MNIST

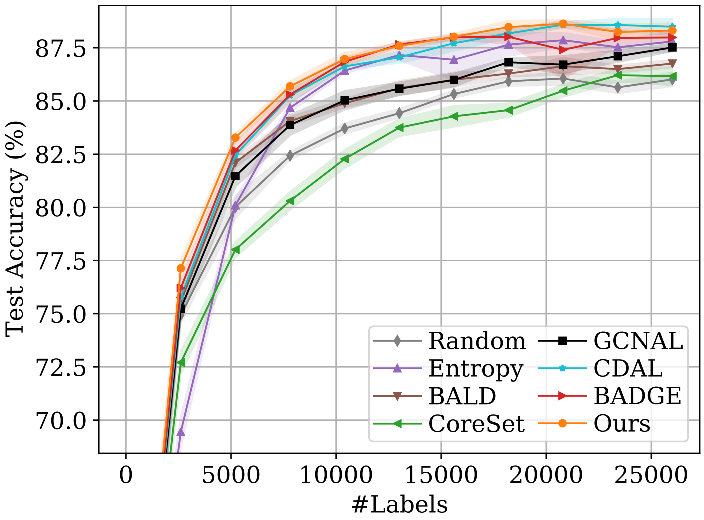


FIGURE 3.35: Large Budget,MLP, EMNIST

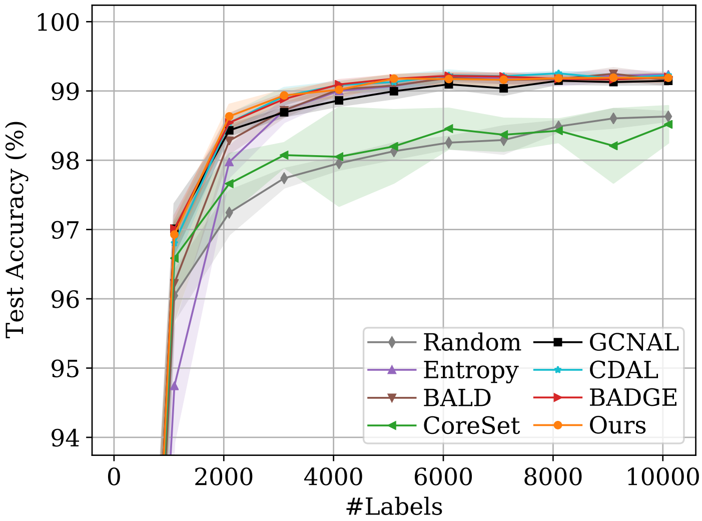


FIGURE 3.36: Large Budget, LeNet-5, MNIST

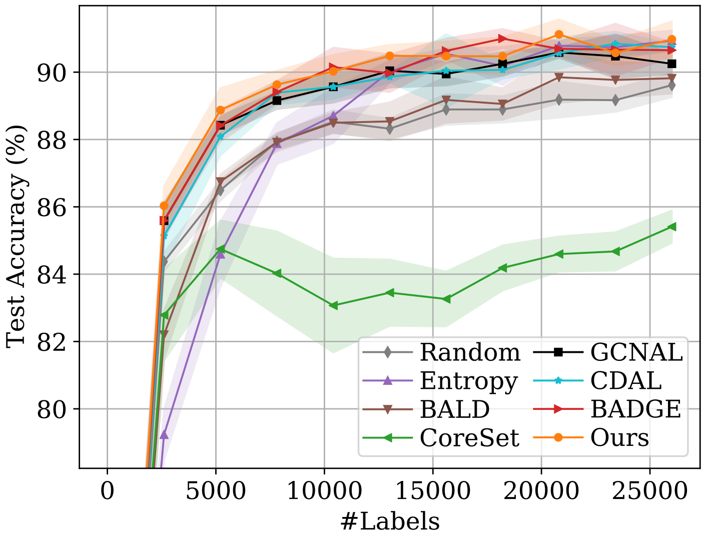


FIGURE 3.37: Large Budget, LeNet-5, EMNIST

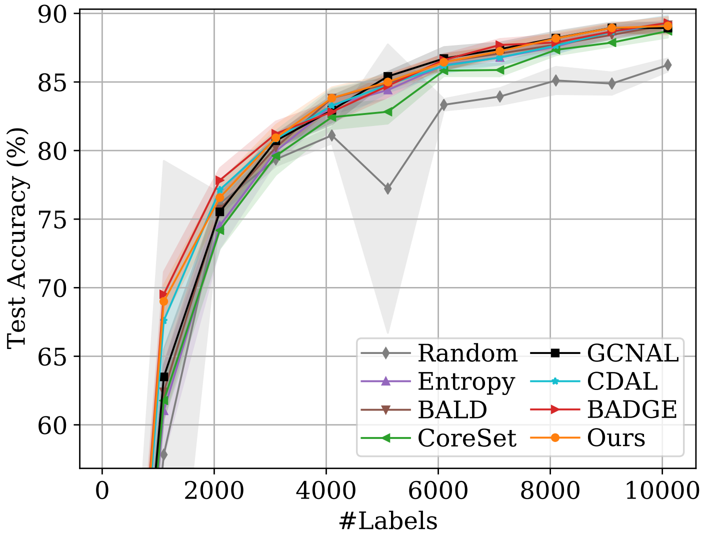


FIGURE 3.38: Large Budget, ResNet-18, SVHN

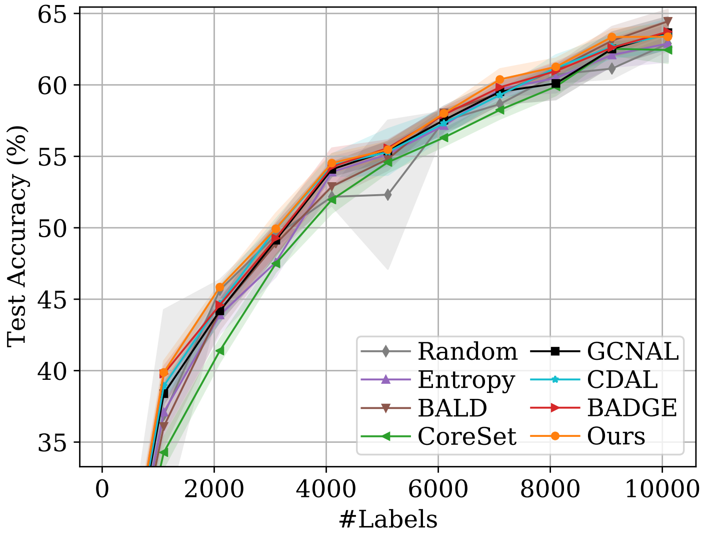


FIGURE 3.39: Large Budget, ResNet-18, CIFAR10

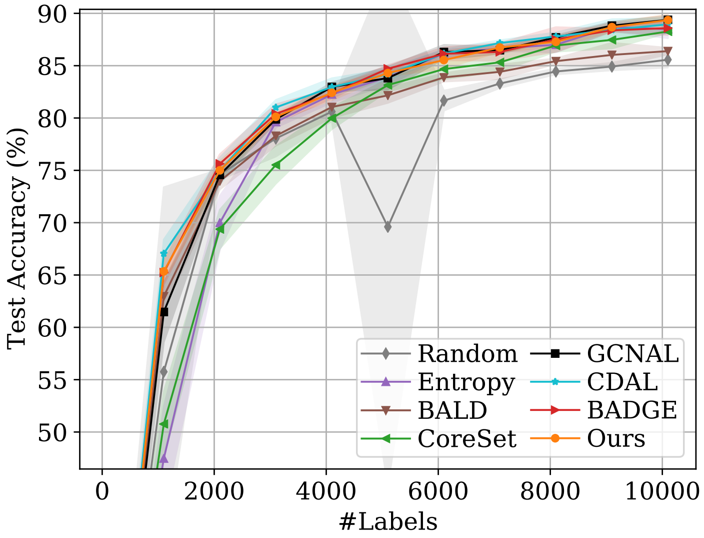


FIGURE 3.40: Large Budget, DenseNet-121, SVHN

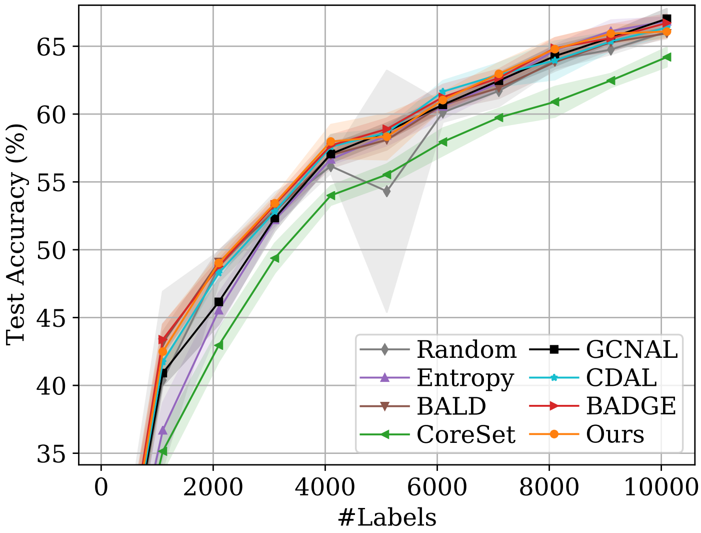


FIGURE 3.41: Large Budget, DenseNet-121, CIFAR10

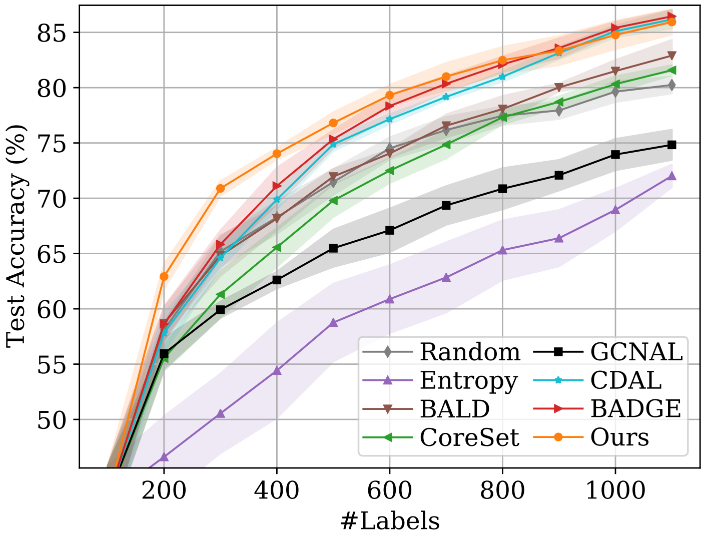


FIGURE 3.42: Small Budget, MLP, OpenML-6

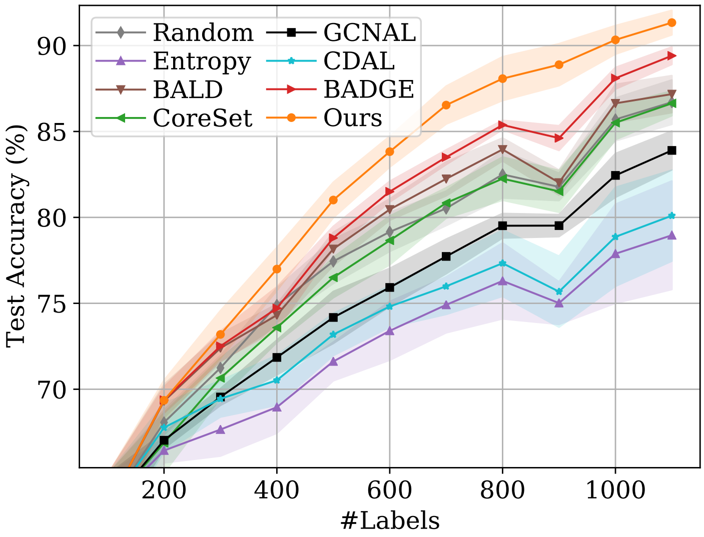


FIGURE 3.43: Small Budget, MLP, OpenML-155

Chapter 4

Counterfactual Vision and Language Learning

This chapter covers the problem of the model's tendency towards extracting statistical regularities instead of learning generalisable features in some vision and language applications with limited training instances. We discuss our proposed approach based on the counterfactuals, which leads to better generalisation of the model.

4.1 Overview

The ongoing success of visual question answering methods has been somewhat surprising given that, at its most general, the problem requires understanding the entire variety of both visual and language stimuli. It is particularly remarkable that this success has been achieved on the basis of comparatively small datasets, given the scale of the problem. One explanation is that this has been accomplished partly by exploiting bias in the datasets rather than developing deeper multi-modal reasoning. This fundamentally limits the generalisation of the method, and thus its practical applicability. We propose a method that addresses this problem by introducing counterfactuals in the training. In doing so we leverage structural causal models for counterfactual evaluation to formulate alternatives, for instance, questions that could be asked of the same image set. We show that simulating plausible alternative training data through this process results in better generalisation.

4.2 Introduction

Recent advances in computer vision and natural language understanding have paved the way for a variety of tasks that combine visual and textual modalities (Kiros et al. 2015; Das et al. 2017a; Agrawal et al. 2017; Abbasnejad et al. 2019; Ramakrishnan, Agrawal, and Lee 2018). Visual Question Answering (VQA) is one such task in which the goal is to answer a question framed in natural language that relates to an image. VQA thus requires a high-level understanding of the visual scene and the question, and an ability to relate (or ground) the two. Much of the interest around VQA, and the associated vision-and-language problems, stems from the fact that success might represent a step toward artificial intelligence. A variety of real-world applications have arisen also, including aiding the visually impaired, searching through large quantities of visual data via natural language interfaces, and flexible tasking of robots.

Current end-to-end VQA models achieve high accuracies on most of the available benchmarks and surpass human performance in a selection of cases (compositional reasoning (Johnson et al. 2017), for example). It has been shown, however, that these methods exploit statistical regularities and biases in the data to achieve this performance (Johnson et al. 2017; Ramakrishnan, Agrawal, and Lee 2018; Hudson and Manning 2019; Agrawal et al. 2018). In addition, although these approaches are expected to merge information from multiple modalities, in practice they often exploit unimodal biases and ignore the other modalities entirely. In addition, particular signals in the input trigger specific answers; for instance, when the image contains a banana, the answer is most likely to be yellow, irrespective of the remainder of the image, or the question. This dependence on spurious correlations in the training data leaves VQA methods vulnerable to a failure to generalise. In addition, this phenomenon highlights the lack of high-level understanding of the input and its connection to other modalities.

To remedy the weaknesses identified above and improve generalisation, we propose to utilize *counterfactuals (Pearl 2009; Charles, Chickering, and Simard 2013)*

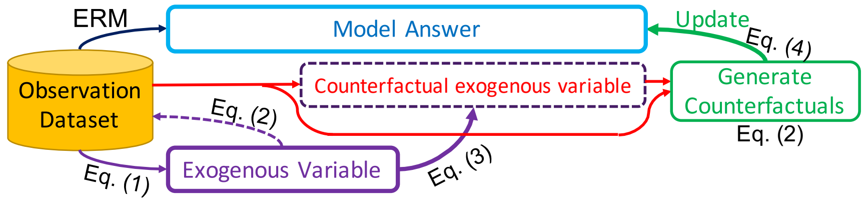


FIGURE 4.1: The training process with counterfactuals. We infer the posterior on the exogenous variables. Subsequently generate counterfactual samples using that variable and evaluate its output.

in the learning process. In traditional causal inference counterfactuals are *unobserved* scenarios, and are often used to estimate the effect of an intervention that is not directly represented in the data. In machine learning they can equally represent a potential training data element for which we do not have a label, or a data-label pair for which we do not have a reward. This is particularly relevant in those supervised learning settings where more than one true label might apply to each training data element, yet only one true answer is typically observed. This is the case in many vision-and-language problems, as the fact that the training set documents a particular answer to a VQA question does not mean that every alternate answer is wrong. This is referred to as *bandit feedback* (Johansson, Shalit, and Sontag 2016), and such problems are labelled *nonstochastic multiarmed bandit problems* (Auer et al. 2002). In the context of VQA, counterfactual analysis leads us to ask *"what would have happened if we observed a different image or asked a different question, given the past observations"*.

We consider the causal model underlying the training data, and introducing an extra (exogenous) variable that governs the question and image generation (from which the observed answers are produced). Then, we learn a distribution for that variable, providing a model of how the observational data was generated. Subsequently, we ask "*what would be the minimum alteration to the question or image that could change the answer*". To that end, we choose the exogenous variable such that the question or image generated using that variable yields an incorrect answer, thus effectively injecting an *intervention* into our causal model. Since the intervention can degrade the model's performance, we "reason" about these counterfactual instances by formulating an alternative to conventional empirical risk minimization, allowing the model to learn both from observational and counterfactual instances. This implicitly forces the VQA model to use both input modalities instead of relying on statistical regularities specific to either of them. Further, training a model to both learn to answer and "reason" about the intervention in questions and images, encourages generalisation. In Fig. 4.1, our approach is summarized.

By effectively "asking the algorithm" what would have happened, we aim to highlight the most interesting cases of disagreement between the counterfactuals and the training observations, while also demonstrating implicitly why the learned model is preferred.

We describe extensive experiments on VQA-CP (Agrawal et al. 2018), VQA 2.0 (Agrawal et al. 2017), Embodied QA (Das et al. 2018a) (where agent requires navigation to answer questions) and Room-to-Room (R2R) Navigation (where the agent should follow a natural language instruction to navigate) and demonstrate the ability of our approach to improve generalisation. Our contributions in this chapter are:

- We provide a counterfactual framework under which the interventions in the inputs, either the question or image, are anticipated. We show that a simple model of learning the distribution of an exogenous intervention variable of the observational data, and subsequently counterfactual samples generated from that variable improvs generalisation. We encourage the model to reason about "what the answer could be about a counterfactual image or question".

- We provide a theoretical analysis for the proposed approach to shed light on its underlying working mechanism. In addition, we show a lower bound on the likelihood of the counterfactuals based on the observations.

- Our extensive experiments show that our simple yet powerful approach is capable of improving the generalisation ability of diverse multimodal and unimodal vision and language tasks. In VQA-CP we observe more than 2%

improvement over the baseline when using the full set and 7% when using a fraction of the dataset. In both Embodied QA and R2R navigation, our approach improves the state-of-the-arts by more than 2%.

4.3 Related Work

Counterfactuals (Charles, Chickering, and Simard 2013; Pearl 2009) have gained recent interest in various areas in machine learning, in particular in applying insights from causal inference to augment the training as in bandit settings (Johansson, Shalit, and Sontag 2016; Abbasnejad, Domke, and Sanner 2015), reinforcement learning (Buesing et al. 2019), recommendation (Swaminathan and Joachims 2015) and explanation (Goyal et al. 2019b). *Adversarial* learning (Goodfellow, Shlens, and Szegedy 2015) is a prime instance of use of counterfactuals in learning and was shown to improve performance (e.g. (Zhang et al. 2018b)). However, most of the state-of-the-art in this area focus on the analysis of the outcome of an intervention of sorts, i.e. change in the input or model. Our approach however, focuses on both proper generation of the counterfactuals from intervention and ensuring the outcome is adjusted in an alternative risk minimization.

Data Augmentation lies at the heart of successful machine learning where substantial domain knowledge is leveraged to design suitable data transformations (e.g. rescaling, rotation, etc) leading to improved generalisation. While learning these invariances, using for instance generative models, can potentially alleviate the problem, their use is nontrivial.

Recently, MixUp (Zhang et al. 2018a) was proposed as a simple means for data augmentation and regularization which does not require significant domain knowledge. Similar to label smoothing, the supervision of every example is not overly dominated by the ground-truth label. Moreover, the augmented data is transformed from training instances to establish a linear relationship between data augmentation and the supervision signal. However, it requires sampling a mixing parameter that is not trivial to choose. Our approach on the other hand, learns to interpolate depending on the difficulty of producing its output for the model and the landscape in the feature space, hence harnessing the advantages of MixUp for sample generation.

Biases in VQA datasets and models are major pitfalls in current models where superficial correlations between inputs from one modality and the answers are exploited by models (Manjunatha, Saini, and Davis 2019; Goyal et al. 2019a; Ramakrishnan, Agrawal, and Lee 2018). Unfortunately, biased models that exploit statistical shortcuts from one modality usually reach impressive accuracy on most of the current benchmarks. VQA-CP (Agrawal et al. 2018) is a recent diagnostic datasets containing different answer distributions for each question-type leading to different distribution of train and test splits. Consequentially, models biased towards one of the modality often fail at this benchmark. Human provided additional balancing data, for instance in the case of VQA v2 (Goyal et al. 2019a) has not resolved the issue. More elaborate models to avoid biases such as Grounded VQA (Agrawal et al. 2018) introduces additional submodules that are not trivial to be used with novel architectures. Similarly, (Ramakrishnan, Agrawal, and Lee 2018) proposed a model-agnostic learning strategy to overcome language priors in VQA models by directly penalizing the input question-only bias. In (Damien Teney 2020), the authors cluster training questions using to their prefix to prevent the model from relying on them as features.

Our method is model-agnostic, easy to implement and does not need an elaborate parameter tuning or prior knowledge. In addition, our approach naturally leverages inherent dependencies to improve generalisation and discourage simple exploitation of the biases by the model. Our counterfactual training approach discourages learning the biases by relying on the capacity to generate samples that can change the predictions.

4.3.1 Visual Question Answering

Visual Question Answering (VQA) is the task of answering previously unseen questions framed in natural language about a previously unseen image. For training, we are interested in learning a model from a training set made up of image $\boldsymbol{v}$, question $\boldsymbol{q}$ and answer a triplets $\mathcal{D} = \{\langle \boldsymbol{q}_i, \boldsymbol{v}_i, a_i \rangle\}_{i=1}^{n}$. During test time, given an image and question, the trained model predicts the correct answer. The classical approach for VQA is to use an embedding of the questions $\mathrm{e}^q = f_q(\boldsymbol{q})$, an embedding of the image $\mathrm{e}^v = f_v(\boldsymbol{v})$ and a fusion function of the two $\mathbf{z} = h(\mathrm{e}^q, \mathrm{e}^v)$ into what is known as the joint space. We denote by $\boldsymbol{\theta}$ all of the parameters of the deep models used to learn these representations and generate answers. Using better embeddings yields better joint space representations and consequently more accurate answers. For brevity below we omit the parameters in the models, i.e. we use $p(a|\boldsymbol{q}, \boldsymbol{v})$ as a shorthand for $p(a|\boldsymbol{q}, \boldsymbol{v}, \boldsymbol{\theta})$.

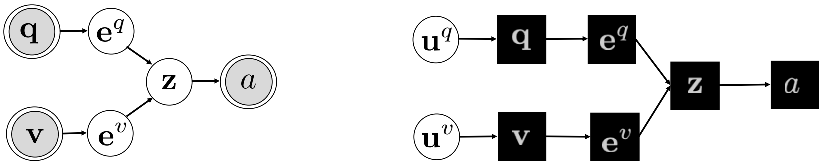
4.3.2 Counterfactuals

In the following we provide a background on counterfactuals that will form the basis for the rest of this chapter. Interested readers are referred to (Pearl 2009) for further details

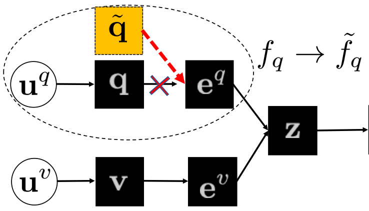
Definition 1 (Structural Causal Model (SCM))**.** A structural causal model $\mathcal{M}$ consists of a set of independent (exogenous) random variables $\mathbf{u} = \{\mathbf{u}_1, \ldots, \mathbf{u}_n\}$ with distribution $P(\mathbf{u})$, a set of functions $\mathbf{F} = \{f_1, \ldots, f_n\}$, and a set of variables $\mathbf{X} = \{X_1, \ldots, X_n\}$ such that $X_i = f_i(\mathbf{PA}_i, \mathbf{u}_i), \forall i$, where $\mathbf{PA}_i \subseteq \mathbf{X} \setminus X_i$ is the subset of $\mathbf{X}$ which are parents of X_i. As a result, the prior distribution $P(\mathbf{u})$ and functions determine the distribution $P^{\mathcal{M}}$.

An SCM defines the data generating process and the distribution of the observations. Using this model, we can investigate the consequences of intervention.

Definition 2 (Interventional Distribution)**.** For an SCM $\mathcal{M}$, a soft intervention $I = \mathrm{do}\big(X_i := \tilde{f}_i(\tilde{\mathbf{PA}}_i, \mathbf{u}_i)\big)$ corresponds to replacing the structural mechanism $f_i(\mathbf{PA}_i, \mathbf{u}_i)$ with $\tilde{f}_i(\tilde{\mathbf{PA}}_i, \mathbf{u}_i)$. We simply write $\mathrm{do}(X_i = x)$ to denote the hard intervention that substitutes the structural mechanism with a constant value. The resulting intervened SCM is denoted $\mathcal{M}^I$, and the resulting interventional distribution is denoted $P^{\mathcal{M};I}$.



(A) VQA model (B) Our SCM



(C) Our SCM with Intervention

FIGURE 4.2: The difference between a typical VQA graphical model (in Fig. 5.2a), our corresponding causal model (in Fig. 5.2b) and an example of intervention in the question representation of this model (in Fig. 4.2c). In our model two exogenous variables $\mathbf{u}^q$ and $\mathbf{u}^v$ are incorporated to learn and reason about the intervention caused by these variables.

We can also define the *counterfactual distribution* which tells us what might have happened had we acted differently.

Definition 3 (Counterfactual Distribution)**.** Given an SCM $\mathcal{M}$ and an observed assignment $\mathbf{X} = \mathbf{x}$ over any set of observed variables, the counterfactual distribution $P^{\mathcal{M}|\mathbf{X}=\mathbf{x};I}$ corresponds to the distribution entailed by the SCM $\mathcal{M}^I$ using the posterior distribution $P(\mathbf{u}|\mathbf{X} = \mathbf{x})$.

For an SCM $\mathcal{M}$, the counterfactual distribution can be estimated by first inferring the posterior over exogenous variables and then passing that distribution through the modified structural model $\mathcal{M}^I$ to obtain a counterfactual distribution over other variables*.

4.4 Counterfactual Vision and Language (CVL)

Our intuition is that the functions that extract the features in a VQA system, either from the image or the question, are prone to focusing on spurious correlations in the data, which diverts them from modeling the deeper relations that generalise better. Hence, we encourage the learning algorithm to consider counterfactuals–a set of imaginary alternative samples. Training a model to both learn to answer, and “reason”

---

*Called abduction, action, and prediction in (Pearl 2009) about the intervention in the questions and images allows better generalisation. To that end, we construct the SCM as shown in Fig. 5.2 where the functions for learning the embeddings are conditioned on the exogenous variables.

As is the convention for intervention in counterfactual reasoning, we are interested in replacing the embedding functions by their corresponding counterfactuals, that is, f_v is replaced by $\tilde{f}_v(\boldsymbol{v}, \mathbf{u}^v)$ or f_q by $\tilde{f}_q(\boldsymbol{q}, \mathbf{u}^q)$ where $\mathbf{u}^v$ and $\mathbf{u}^q$ are exogenous variables for image (vision module) and question (language module), respectively. Note that $\tilde{f}_v(\cdot, \cdot)$ and $\tilde{f}_q(\cdot, \cdot)$ are the functions of the exogenous variables for a given image and question pair. Effectively our approach reasons about the interventions in the embedding extractions. We use $\mathbf{u} = [\mathbf{u}^v, \mathbf{u}^q]$ to denote both of the exogenous variables. We denote by $\tilde{\boldsymbol{q}}$ and $\tilde{\boldsymbol{v}}$ the variables obtained after the intervention and $\tilde{\mathrm{e}}^q$ and $\tilde{\mathrm{e}}^v$ as their corresponding embeddings. This intuitively allows our model to answer image-based questions it has never observed. We are generally interested in the following objectives: (1) the joint space of the question-image embedding must lead to a low-error rate on the factual data; (2) the conditional distribution of the factual and counterfactual data considering the exogenous distribution must be similar; (3) the distribution of the exogenous variables must be obtained from the observations; and (4) the embedding has to yield small error on the unobserved counterfactual distribution (obtained from the intervention in the structural model).

The first objective is the same as any other vision and language task. The second is a necessary constraint to ensure using a model from the observations we can predict answers for counterfactuals. The third objective ensures the possible intervention distribution from the exogenous variable is learned as part of the model. Lastly, our approach should be able to reason about the answer to the counterfactual instances (see Fig. 4.3 for an example). As such, we devise the following steps through which our model is trained and the distribution of the exogenous variable is found:

1. Infer the predictive model for the observed data using one step of the conventional risk minimization.

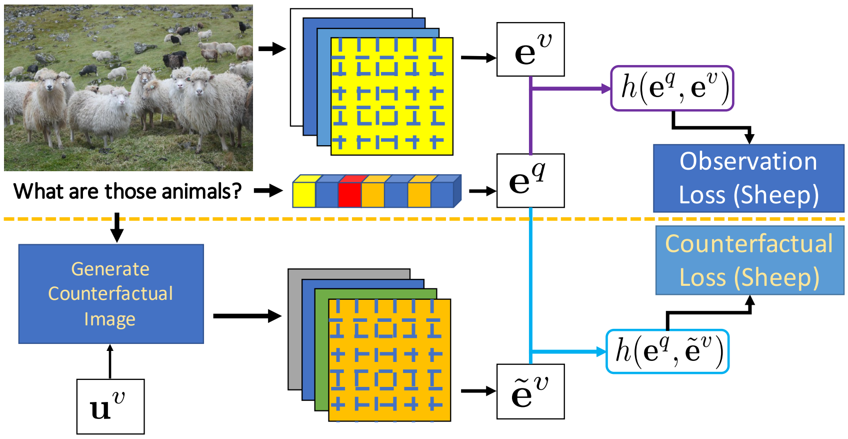

What are those animals?
$\mathbf{e}^v$
$\mathbf{e}^q$
$h(\mathbf{e}^q, \mathbf{e}^v)$
Observation Loss (Sheep)
Counterfactual Loss (Sheep)
Generate Counterfactual Image
$\mathbf{u}^v$
$\tilde{\mathbf{e}}^v$
$h(\mathbf{e}^q, \tilde{\mathbf{e}}^v)$

FIGURE 4.3: Counterfactual examples that can be generated for a given image. The fusion function h is used with the observational data as well as the counterfactual data to predict the answer. For the counterfactual loss, we need to consider the relationship between the predicted counterfactual answer and its observational counterpart.

2. Perform intervention I on $\mathcal{M}$. This yields $\mathcal{M}^I$, which entails the counterfactual distribution $p^{\mathrm{do}(I)|\tilde{\boldsymbol{q}},\tilde{\boldsymbol{v}}}$.

3. Reason about the effect of that intervention on the answer and the loss that incurs.

Intuitively, first we learn what distribution of the exogenous variable is obtained from the observations, then model how the answer is affected by the intervention on this variable.

4.5 Counterfactual Distribution

The counterfactual distribution is the posterior of the exogenous variables obtained from the observations. Hence, using the training data we are interested in[†]

$$p(\mathbf{u}|\mathcal{D}) \propto p(\mathbf{u}) \prod_{i=1}^{n} p(a_i|\boldsymbol{q}_i, \boldsymbol{v}_i) p(\boldsymbol{v}_i|\mathbf{u}^v) p(\boldsymbol{q}_i|\mathbf{u}^q). \tag{4.1}$$

We use independent priors, i.e. $p(\mathbf{u}) = p(\mathbf{u}^q)p(\mathbf{u}^v)$ with Beta distributions for $\mathbf{u}^v$ and $\mathbf{u}^q$ (i.e. $\mathbf{u}^v \sim \mathrm{Beta}(\alpha_0, \beta_0)$). Although we could estimate $p(\boldsymbol{v}_i|\mathbf{u}^v)$ and $p(\boldsymbol{q}_i|\mathbf{u}^q)$ using various methods (including autoencoders (Kingma and Welling 2013; Abbasnejad,

---

[†] We note that without loss of generality and for brevity we drop the dependence on the embedding features $p(a_i|\boldsymbol{q}_i, \boldsymbol{v}_i) = p(a_i|\mathbf{e}^v, \mathbf{e}^q) \times \delta(\mathbf{e}^q - f_q(\boldsymbol{q}_i)) \times \delta(\mathbf{e}^v - f_v(\boldsymbol{v}_i))$ where δ is the Dirac delta.

Dick, and Hengel 2017) and GANs (Goodfellow et al. 2014; Abbasnejad et al. 2019; Abbasnejad, Shi, and Hengel 2018)), we use a simple approach to model the question or image's conditional likelihood. To obtain the posterior, considering the generating process of $\boldsymbol{q}_i$ and $\boldsymbol{v}_i$ for a given sample of the variable $\mathbf{u}^q, \mathbf{u}^v$ and an arbitrary constant $0 \leq \epsilon < 1$, we have

$$\begin{aligned} \boldsymbol{q} \sim p(\boldsymbol{q}|\mathbf{u}^q) &= \begin{cases} \boldsymbol{q} & \mathbf{u}^q \geq 1-\epsilon \\ \mathbf{u}^q\boldsymbol{q} \oplus (1-\mathbf{u}^q)\boldsymbol{q}', & \text{otherwise} \end{cases}, \quad \text{and} \\ \boldsymbol{v} \sim p(\boldsymbol{v}_i|\mathbf{u}^v) &= \begin{cases} \boldsymbol{v} & \mathbf{u}^v \geq 1-\epsilon \\ \mathbf{u}^v\boldsymbol{v} \oplus (1-\mathbf{u}^v)\boldsymbol{v}', & \text{otherwise} \end{cases} \end{aligned} \tag{4.2}$$

where $\boldsymbol{q}'$ and $\boldsymbol{v}'$ are uniformly sampled at random from the dataset and $\oplus$ denotes an interpolation. It is easy to see that for $\epsilon \rightarrow 0$ we have more interpolated samples and for $\epsilon \rightarrow 1$, we obtain samples that are independent of the prior. An advantage of this approach of sampling the observations is that we effectively reduce the conditional independence assumption of the training data allowing for the relation between observations to be established.

Since we use all conjugate priors, the posterior is also a Beta distribution with parameters α, β where $\alpha = \alpha_0 + \sum \mathbb{I}[a_i = \arg\max p(a_i|\boldsymbol{q}_i, \boldsymbol{v}_i)]$ and $\beta = \beta_0 + \sum \mathbb{I}[a_i \neq \arg\max p(a_i|\boldsymbol{q}_i, \boldsymbol{v}_i)]$. Intuitively, samples from the regions of the prior that produce the correct answers are "successful" and encourage the posterior to concentrate. Notice that the samples from the posterior are drawn from the regions where the likelihood of the correct answer is higher (since the expectation of the posterior is $\alpha/(\alpha+\beta)$).

4.5.1 Generating Counterfactuals

Once the posterior on the exogenous variables $p(\mathbf{u}|\mathcal{D})$ is obtained, we perform the intervention. That is, we generate the counterfactuals and replace the $\boldsymbol{v}$ (or $\boldsymbol{q}$) with its alternative $\tilde{\boldsymbol{v}}$ (or $\tilde{\boldsymbol{q}}$) and anticipate the answer. This corresponds to replacing the function $f_v(\cdot,\cdot)$ (or $f_q(\cdot,\cdot)$) with an alternative $\tilde{f}_v(\cdot,\cdot)$ (or $\tilde{f}_q(\cdot,\cdot)$) which leads to a different answer prediction.

In obtaining the counterfactual samples we are interested in the minimum interventions that will change the answer for a given question-image pair $(\boldsymbol{q},\boldsymbol{v})$ to $(\tilde{\boldsymbol{q}},\tilde{\boldsymbol{v}})$ when using the generating process in Eq. 5.9. This corresponds to a sample from the posterior of the exogenous variable with high likelihood (minimum intervention) that will alter the answer for $(\boldsymbol{q},\boldsymbol{v})$ to an incorrect one. As such, we formalize the problem as:

$$\begin{aligned}
\max_{\mathbf{u}} \quad & \log(p^{\mathrm{do}(I)|\boldsymbol{q},\boldsymbol{v}}(\tilde{\boldsymbol{q}},\tilde{\boldsymbol{v}}|\mathbf{u})) \\
\text{s.t.} \quad & \tilde{a} = \mathrm{argmax}_{a'} \quad p^{\mathrm{do}(I)|\boldsymbol{q},\boldsymbol{v}}(a'|\tilde{\boldsymbol{q}},\tilde{\boldsymbol{v}}), \forall \tilde{a} \neq a \\
& 0 \leq \mathbf{u} < 1
\end{aligned}$$

Considering the generative process in Eq. 5.9, the minimum intervention (the minimum edit of the factual (Qin et al. 2019; Goyal et al. 2019b)) is achieved when $\mathbf{u}$ is largest. Since the constraint is not computationally feasible, we relax the objective and choose the variable that has the minimum likelihood of having the same answer as the observations. Thus, we choose $\mathbf{u}$ from the relaxed alternative (we project $\mathbf{u}$ to be bounded in $[0,1)$)

$$\max_{\mathbf{u}} \quad \|\mathbf{u}\|^2 - \lambda \log\left(p^{\mathrm{do}(I)|\boldsymbol{q},\boldsymbol{v}}(a|\tilde{\boldsymbol{q}},\tilde{\boldsymbol{v}})\right) \tag{4.3}$$

where λ is a hyper-parameter. We note that simply sampling from the posterior $p(\mathbf{u}|\mathcal{D})$ and generating $\boldsymbol{v}$ (or $\boldsymbol{q}$) to infer the answer, is not the counterfactual (alternating between sampling the variable $\mathbf{u}$ and learning parameter $\boldsymbol{\theta}$ resembles conventional Gibbs sampling). Hence, this step is critical to obtain instances that are *not* merely from the learned distribution, yet very likely. Consequently, enabling our approach to generalise better beyond observations.

4.5.2 Counterfactual Loss

We alternate between intervening in the inputs, and minimizing the risk on the corresponding counterfactual along with the observations. As is common practice in empirical risk minimization (ERM), the objective in using observational training instances is minimizing $\mathbb{E}_{\boldsymbol{q},\boldsymbol{v}}\mathbb{E}_{p(a|\boldsymbol{q},\boldsymbol{v})}[\ell(f_{\boldsymbol{\theta}}(\boldsymbol{q},\boldsymbol{v}),a)]$ where $\ell(f_{\boldsymbol{\theta}}(\boldsymbol{q},\boldsymbol{v}))$ is the loss of the function predicting the answer. Note that in practice $f_{\boldsymbol{\theta}}$ and $p(a|\boldsymbol{q},\boldsymbol{v},\boldsymbol{\theta})$ may be the same function or share architecture (e.g. $p(a|\boldsymbol{q},\boldsymbol{v},\boldsymbol{\theta}) = \text{softmax}(f_{\boldsymbol{\theta}}(\boldsymbol{q},\boldsymbol{v}))$). In the case of using counterfactuals, we can rewrite the risk by changing the distribution (Charles, Chickering, and Simard 2013):

$$\begin{aligned} R(\boldsymbol{\theta}) &= \mathbb{E}_{\boldsymbol{q},\boldsymbol{v}}\mathbb{E}_{p(a|\boldsymbol{q},\boldsymbol{v})}[\ell(f_{\boldsymbol{\theta}}(\boldsymbol{q},\boldsymbol{v}),a)] \\ &= \mathbb{E}_{\boldsymbol{q},\boldsymbol{v}}\mathbb{E}_{p^{\text{do}(I)|\boldsymbol{q},\boldsymbol{v}}(a|\tilde{\boldsymbol{q}},\tilde{\boldsymbol{v}})}\left[\ell(f_{\boldsymbol{\theta}}(\boldsymbol{q},\boldsymbol{v}),a)\frac{p(a|\boldsymbol{q},\boldsymbol{v},\boldsymbol{\theta})}{p^{\text{do}(I)|\boldsymbol{q},\boldsymbol{v}}(a|\tilde{\boldsymbol{q}},\tilde{\boldsymbol{v}},\boldsymbol{\theta})}\right] \end{aligned}$$

Note that $p^{\text{do}(I)|\boldsymbol{q},\boldsymbol{v}}(a|\tilde{\boldsymbol{q}},\tilde{\boldsymbol{v}},\boldsymbol{\theta})$ has part of SCM altered. Intuitively, the counterfactuals that have smaller scores are more penalized and conversely the over-confident ones are discouraged. This subsequently adjusts the decision boundary to be discriminative for both observations and counterfactuals. Furthermore, since this risk can have a very high variance we can clip this value similar to (Charles, Chickering, and Simard 2013),

$$\begin{aligned} R^M(\boldsymbol{\theta}) = \mathbb{E}_{\boldsymbol{q},\boldsymbol{v}}\mathbb{E}_{\tilde{p}^u(a|\boldsymbol{q},\boldsymbol{v})}\Big[&\ell(f_{\boldsymbol{\theta}}(\boldsymbol{q},\boldsymbol{v}),a) \\ &\times \min\Big\{M,\frac{p(a|\boldsymbol{q},\boldsymbol{v},\boldsymbol{\theta})}{p^{\text{do}(I)|\boldsymbol{q},\boldsymbol{v}}(a|\tilde{\boldsymbol{q}},\tilde{\boldsymbol{v}},\boldsymbol{\theta})}\Big\}\Big] \end{aligned}$$

This is because we may have very low probability in predicting an output of an intervened observation. Thus, the empirical counterfactual risk is,

$$\begin{aligned} \hat{R}^M(\boldsymbol{\theta}) &= \frac{1}{n}\sum_{i=1}^{n}\ell(f_{\boldsymbol{\theta}}(\boldsymbol{q}_i,\boldsymbol{v}_i),a_i)\times\omega_i(\boldsymbol{\theta}) \\ \text{where}\quad \omega_i(\boldsymbol{\theta}) &= \min\Big\{M,\frac{p(a_i|\boldsymbol{q}_i,\boldsymbol{v}_i,\boldsymbol{\theta})}{p^{\text{do}(I)|\boldsymbol{q},\boldsymbol{v}}(a|\tilde{\boldsymbol{q}},\tilde{\boldsymbol{v}},\boldsymbol{\theta})}\Big\}. \end{aligned} \tag{4.4}$$

Here, $\omega_i(\boldsymbol{\theta})$ is the clipped ratio of evaluation of the factual sample i and its corresponding counterfactual. We intentionally use a shorthand to underscore The fact that the parameters are optimized with respect to $\boldsymbol{\theta}$ in p. The objective of the counterfactual risk minimization for vision and language tasks is therefore

$$\hat{R}^{M*} \quad = \quad \arg\min_{\boldsymbol{\theta}} \hat{R}^{M}(\boldsymbol{\theta})$$

In practice, we alternate between the conventional ERM (i.e. when $\omega(\boldsymbol{\theta}) = 1$) and the counterfactual risk.

4.5.3 Further Analysis

When we generate samples in Eq. 5.9, $\boldsymbol{q}'$ is likely to have a different answer to $\boldsymbol{q}$ (with probability $(1 - {}^{n_a}\!/_{n})$ for n_a denoting the number of instances with answer a). As such, interpolating between the questions and images will lead to samples for which the answer is uncertain. In the case of the generated counterfactuals, however, such interpolations are in fact close to the decision boundary. Hence, when weighted by the confidence of the classifier in Eq. 4.4, the connection between samples in the fusion space (i.e. the common semantic space) is adjusted to account for the sensitivity of the representations to changes in the input.

Furthermore, one main question is how do we know that the interventions won't lead to divergence, or learning useless models. We can derive the bound on the risk using the following theorem:

Theorem 4. *Denote* $u^i(\boldsymbol{\theta}) \equiv \ell(f_{\boldsymbol{\theta}}(\boldsymbol{q}_i, \boldsymbol{v}_i), a_i)\omega_i(\boldsymbol{\theta})$, $\overline{u} \equiv \sum_{i=1}^{n} u^i(\boldsymbol{\theta})/n$, $\hat{\mathbb{V}}(u) \equiv \sum_{i=1}^{n} \left(u^i(\boldsymbol{\theta}) - \overline{u}\right)^2 / (n-1)$ *and* $\mathcal{Q}_\gamma \equiv \log\left(10 \cdot \epsilon/\gamma\right)$ *for* $0 < \gamma < 1$ *and* ϵ *the* ϵ*-cover for the function class that predicts the answer. With probability at least* $1 - \gamma$ *for* $n \geq 16$ *we have*

$$R(\boldsymbol{\theta}) \leq \hat{R}^{M}(\boldsymbol{\theta}) + \sqrt{18\hat{\mathbb{V}}(u)\mathcal{Q}_\gamma/n} + 15M\mathcal{Q}_\gamma/(n-1)$$

Proof. Follows the proof in Theorem 6 of Maurer and Pontil 2009. □

This result implies that when we have the counterfactual risk minimized, we achieve the minimum variance.

We note that for counterfactuals, we have:

$$\begin{aligned}
p^{\mathrm{do}(I)}(\tilde{\boldsymbol{q}}, \tilde{\boldsymbol{v}}) &= \int p^{\mathrm{do}(I)}(\tilde{\boldsymbol{q}}, \tilde{\boldsymbol{v}}|\mathbf{u}) p^{\mathrm{do}(I)}(\mathbf{u}) d\mathbf{u} \\
&= \int p^{\mathrm{do}(I)}(\tilde{\boldsymbol{q}}, \tilde{\boldsymbol{v}}|\mathbf{u}) p(\mathbf{u}) d\mathbf{u} \\
&= \int p^{\mathrm{do}(I)}(\tilde{\boldsymbol{q}}, \tilde{\boldsymbol{v}}|\mathbf{u}) \left(\int p\left(\boldsymbol{q}, \boldsymbol{v}, \mathbf{u}\right) dp\left(\boldsymbol{q}, \boldsymbol{v}\right) \right) d\mathbf{u} \\
&= \iint p^{\mathrm{do}(I)}(\tilde{\boldsymbol{q}}, \tilde{\boldsymbol{v}}|\mathbf{u}) p\left(\mathbf{u}|\boldsymbol{q}, \boldsymbol{v}\right) dp\left(\boldsymbol{q}, \boldsymbol{v}\right) d\mathbf{u} \\
&= \mathbb{E}_{(\boldsymbol{q}, \boldsymbol{v}) \sim p} \left[\int p^{\mathrm{do}(I)}(\tilde{\boldsymbol{q}}, \tilde{\boldsymbol{v}}|\mathbf{u}) p\left(\mathbf{u}|\boldsymbol{q}, \boldsymbol{v}\right) d\mathbf{u} \right] \\
&= \mathbb{E}_{(\boldsymbol{q}, \boldsymbol{v}) \sim p} \left[p^{\mathrm{do}(I)|\boldsymbol{q}, \boldsymbol{v}} \left(\tilde{\boldsymbol{q}}, \tilde{\boldsymbol{v}}\right) \right].
\end{aligned}$$

Therefore, we can compute the density of the counterfactuals based on the observations, i.e.

$$p^{\mathrm{do}(I)}(\tilde{\boldsymbol{q}}, \tilde{\boldsymbol{v}}) = \mathbb{E}_{(\boldsymbol{q}, \boldsymbol{v}) \sim p(\boldsymbol{q}, \boldsymbol{v})} \left[p^{\mathrm{do}(I)|\boldsymbol{q}, \boldsymbol{v}} \left(\tilde{\boldsymbol{q}}, \tilde{\boldsymbol{v}}\right) \right] \tag{4.5}$$

This result shows that the density of intervened variables $(\tilde{\boldsymbol{q}}, \tilde{\boldsymbol{v}})$ is the marginal of the observations. Hence, the factual, counterfactual and exogenous variables are connected with the following lemma:

Lemma 5. *We have the following lower bound on the log-density of the counterfactuals:*

$$\begin{aligned}
\log(p^{do(I)}(a, \tilde{\boldsymbol{q}}, \tilde{\boldsymbol{v}})) \ \geq \ & \mathbb{E}_{(\boldsymbol{q}, \boldsymbol{v}) \sim p(\boldsymbol{q}, \boldsymbol{v})} \Big[\log(p^{\mathrm{do}(I)|\boldsymbol{q}, \boldsymbol{v}}(a|\tilde{\boldsymbol{q}}, \tilde{\boldsymbol{v}})) \Big] \\
& + \mathbb{E}_{\mathbf{u} \sim p(\mathbf{u})} \Big[\log(p^{\mathrm{do}(I)}(\tilde{\boldsymbol{q}}, \tilde{\boldsymbol{v}}|\mathbf{u})) \Big].
\end{aligned}$$

Proof. We have,

$$\begin{aligned}
p^{\mathrm{do}(I)}(a, \tilde{\boldsymbol{q}}, \tilde{\boldsymbol{v}}) &= p^{\mathrm{do}(I)}(a|\tilde{\boldsymbol{q}}, \tilde{\boldsymbol{v}}) p^{\mathrm{do}(I)}(\tilde{\boldsymbol{q}}, \tilde{\boldsymbol{v}}) \\
&= \mathbb{E}_{(\boldsymbol{q}, \boldsymbol{v}) \sim p} \left[p^{\mathrm{do}(I)|\boldsymbol{q}, \boldsymbol{v}}(a|\tilde{\boldsymbol{q}}, \tilde{\boldsymbol{v}}) p^{\mathrm{do}(I)|\boldsymbol{q}, \boldsymbol{v}} \left(\tilde{\boldsymbol{q}}, \tilde{\boldsymbol{v}}\right) \right].
\end{aligned}$$

Then using Jensen's inequality we have,

$$\begin{aligned}
&\log(\mathbb{E}_{(\boldsymbol{q},\boldsymbol{v})\sim p}\left[p^{\mathrm{do}(I)|\boldsymbol{q},\boldsymbol{v}}(a|\tilde{\boldsymbol{q}},\tilde{\boldsymbol{v}})p^{\mathrm{do}(I)|\boldsymbol{q},\boldsymbol{v}}\,(\tilde{\boldsymbol{q}},\tilde{\boldsymbol{v}})\right]) \\
&\quad\geq \mathbb{E}_{(\boldsymbol{q},\boldsymbol{v})\sim p}\left[\log(p^{\mathrm{do}(I)|\boldsymbol{q},\boldsymbol{v}}(a|\tilde{\boldsymbol{q}},\tilde{\boldsymbol{v}})p^{\mathrm{do}(I)|\boldsymbol{q},\boldsymbol{v}}\,(\tilde{\boldsymbol{q}},\tilde{\boldsymbol{v}}))\right],
\end{aligned}$$

We have:

$$\begin{aligned}
&\mathbb{E}_{(\boldsymbol{q},\boldsymbol{v})\sim p}\left[\log(p^{\mathrm{do}(I)|\boldsymbol{q},\boldsymbol{v}}(a|\tilde{\boldsymbol{q}},\tilde{\boldsymbol{v}})p^{\mathrm{do}(I)|\boldsymbol{q},\boldsymbol{v}}\,(\tilde{\boldsymbol{q}},\tilde{\boldsymbol{v}}))\right] \\
&= \mathbb{E}_{(\boldsymbol{q},\boldsymbol{v})\sim p}\Big[\log(p^{\mathrm{do}(I)|\boldsymbol{q},\boldsymbol{v}}(a|\tilde{\boldsymbol{q}},\tilde{\boldsymbol{v}})) + \log(p^{\mathrm{do}(I)|\boldsymbol{q},\boldsymbol{v}}\,(\tilde{\boldsymbol{q}},\tilde{\boldsymbol{v}}))\Big] \\
&= \mathbb{E}_{(\boldsymbol{q},\boldsymbol{v})\sim p}\Big[\log(p^{\mathrm{do}(I)|\boldsymbol{q},\boldsymbol{v}}(a|\tilde{\boldsymbol{q}},\tilde{\boldsymbol{v}})) + \log(\int p^{\mathrm{do}(I)}(\tilde{\boldsymbol{q}},\tilde{\boldsymbol{v}}|\mathbf{u})p\,(\mathbf{u}|\boldsymbol{q},\boldsymbol{v})\,d\mathbf{u})\Big] \\
&= \mathbb{E}_{(\boldsymbol{q},\boldsymbol{v})\sim p}\Big[\log(p^{\mathrm{do}(I)|\boldsymbol{q},\boldsymbol{v}}(a|\tilde{\boldsymbol{q}},\tilde{\boldsymbol{v}})) + \log(\int p^{\mathrm{do}(I)}(\tilde{\boldsymbol{q}},\tilde{\boldsymbol{v}}|\mathbf{u})p\,(\mathbf{u}|\boldsymbol{q},\boldsymbol{v})\,d\mathbf{u})\Big]
\end{aligned}$$

which is then lower-bounded as

$$\begin{aligned}
&\geq \mathbb{E}_{(\boldsymbol{q},\boldsymbol{v})\sim p}\Big[\log(p^{\mathrm{do}(I)|\boldsymbol{q},\boldsymbol{v}}(a|\tilde{\boldsymbol{q}},\tilde{\boldsymbol{v}})) + \int \log(p^{\mathrm{do}(I)}(\tilde{\boldsymbol{q}},\tilde{\boldsymbol{v}}|\mathbf{u}))p\,(\mathbf{u}|\boldsymbol{q},\boldsymbol{v})\,d\mathbf{u}\Big] \\
&\quad= \mathbb{E}_{(\boldsymbol{q},\boldsymbol{v})\sim p}\Big[\log(p^{\mathrm{do}(I)|\boldsymbol{q},\boldsymbol{v}}(a|\tilde{\boldsymbol{q}},\tilde{\boldsymbol{v}}))\Big] + \mathbb{E}_{(\boldsymbol{q},\boldsymbol{v})\sim p}\Big[\int \log(p^{\mathrm{do}(I)}(\tilde{\boldsymbol{q}},\tilde{\boldsymbol{v}}|\mathbf{u}))p\,(\mathbf{u}|\boldsymbol{q},\boldsymbol{v})\,d\mathbf{u}\Big] \\
&\quad= \mathbb{E}_{(\boldsymbol{q},\boldsymbol{v})\sim p}\Big[\log(p^{\mathrm{do}(I)|\boldsymbol{q},\boldsymbol{v}}(a|\tilde{\boldsymbol{q}},\tilde{\boldsymbol{v}}))\Big] + \int \log(p^{\mathrm{do}(I)}(\tilde{\boldsymbol{q}},\tilde{\boldsymbol{v}}|\mathbf{u}))p\,(\mathbf{u})\,d\mathbf{u}
\end{aligned}$$

□

In fact we can show that even if $\mathbf{u}$ is not drawn from the true generating prior, we can use an arbitrary distribution q and obtain an alternative lower bound to that of Lemma 5:

$$\begin{aligned}
\log(p^{\mathrm{do}(I)}(a,\tilde{\boldsymbol{q}},\tilde{\boldsymbol{v}})) \geq\, & \mathbb{E}_{(\boldsymbol{q},\boldsymbol{v})\sim p(\boldsymbol{q},\boldsymbol{v})}\Big[\log(p^{\mathrm{do}(I)|\boldsymbol{q},\boldsymbol{v}}(a|\tilde{\boldsymbol{q}},\tilde{\boldsymbol{v}}))\Big] \\
& + \mathbb{E}_q[\log(p^{\mathrm{do}(I)}(\tilde{\boldsymbol{q}},\tilde{\boldsymbol{v}}|\mathbf{u}))] \\
& + H(q) - H_q(p).
\end{aligned} \tag{4.6}$$

Effectively using Lemma 5, we know even if the distribution of the exogenous variable for generating the counterfactuals deviates from the true posterior obtained from observations, we can lower-bound the marginal of the counterfactuals which depends on the likelihood of predicting the correct answer, the difference of entropy of the true prior versus the one used and the likelihood of the counterfactual examples.

4.6 Experiments

To evaluate the performance of our approach, we construct experiments on various datasets. We note that our approach is agnostic to the base model used and as such is widely applicable to a wide range of applications. To optimize the objective in Eq. (4.3), we use a simple gradient ascent where we set the learning rate to a constant. We use prior for the exogenous variable as Beta$(0.1, 0.1)$ for the experiments unless otherwise stated. We alternate between the observational training and the counterfactuals.

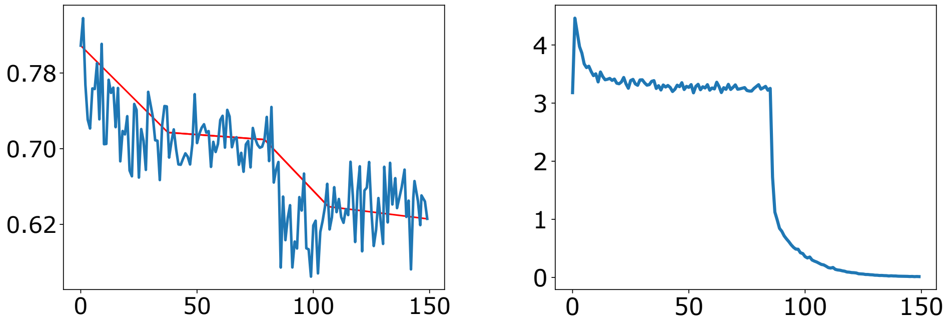
4.6.1 Unimodal Problems

The motivation of our approach is multimodal problems, but it is equally effective for problems involving only a single modality. In this case the description of the process stands, with the exception that either $\mathbf{u}^v$ or $\mathbf{u}^q$ is inferred and used for counterfactual generation.

Stanford Sentiment Treebank (SST) (Socher et al. 2013) is a natural language dataset of movie reviews (neutrals are removed in our experiments). This dataset contains 11855 instances with vocabulary size of 17836 and 5 classes. We follow the implementation of (Tai, Socher, and Manning 2015) where a tree structured LSTM is used with this dataset. We use two alternative baselines for embedding words to be used when sampling in Eq. (5.9): random embedding and trainable GloVe (Pennington, Socher, and Manning 2014) initialized word embeddings. We report mean scores over 5 runs and use 10 epochs for training. Here we examine how the change in the embedding representation effects the performance of the model.

	LSTM	T	LSTM+P	T+P	LSTM+C	T+C
Random	84.4	82.0	84.53	85.21	**85.61**	85.56
GloVe	84.9	86.4	85.77	87.1	87.24	**88.4**

TABLE 4.1: Accuracy (%) obtained by the testing methods using LSTM (with randomly initialized, trainable embeddings). Best results highlighted in Bold. T abbreviates TreeLSTM (Tai, Socher, and Manning 2015); +P and +C indicate posterior and Counterfactuals respectively.



(A) Values of $\mathbf{u}^v$ in Training

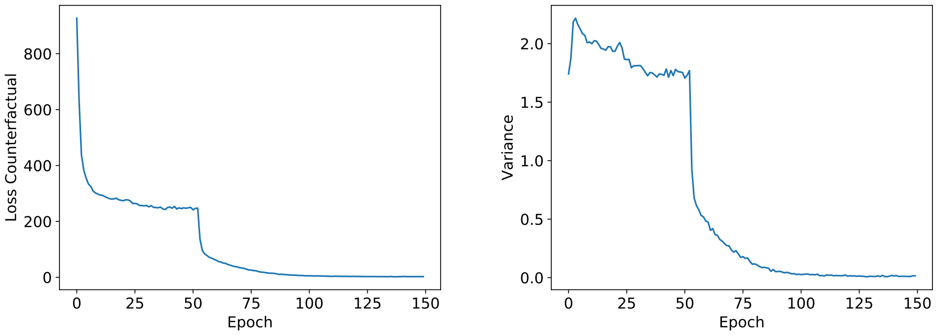
(B) Variance of Loss

FIGURE 4.4: Training metrics in CIFAR experiments.

Since we don't have the image input, we only infer $\mathbf{u}^q$ with prior Beta$(0.1, 0.1)$ and the counterfactual learning rate is set to 0.01. As shown in Table 4.1 using either the posterior (+P models) or the optimized exogenous variable (+C) from Eq. (4.3) improves algorithm accuracy. As expected, when pretrained models are tuned, the overall performance is better.

We further evaluate the generalisation performance of our approach when only the visual data is available on the **CIFAR-10 and CIFAR-100** image classification datasets. In particular, we compare the baseline architectures for: VGG-19 (Simonyan and Zisserman 2014), ResNet-18 (He et al. 2016), ResNet-101 (He et al. 2016), and DenseNet (Huang et al. 2017). All models are trained for 100 epochs on the training set with 128 examples per minibatch and learning rate 0.1, using SGD and evaluated on the test set. The learning rate is then reduced to 0.001 for an additional 150 epochs. We use the interpolations in the input images for Eq. (5.9). In the experiments we have not observed any noticeable difference. We set the prior of $\mathbf{u}^v$ to Beta$(0.1, 0.1)$ and run the counterfactual optimizer for 10 iterations.

We summarize our results in Table 4.2. In both CIFAR-10 and CIFAR-100 classification problems, the models trained using our approach consistently improve



(A) Loss ResNet18

FIGURE 4.5: CIFAR-10 results

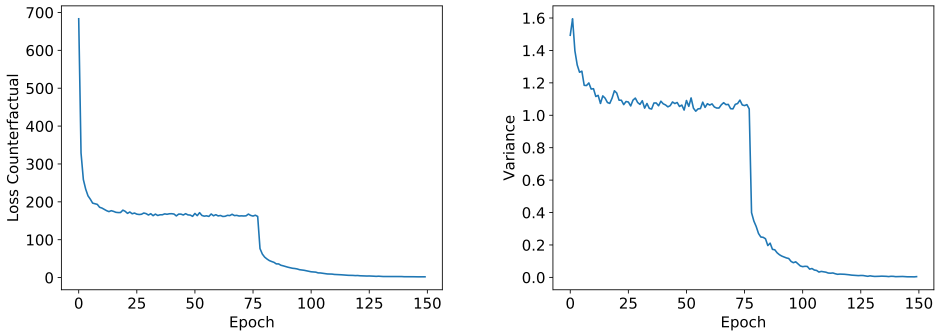
Dataset	Model	Baseline	Ours+P	Ours+C
CIFAR-10	VGG-19	95.04	95.92	**96.73**
	ResNet-18	93.02	94.2	**94.91**
	ResNet-101	93.75	94.1	**95.34**
	DenseNet-121	95.04	95.92	**96.73**
CIFAR-100	VGG-19	72.23	73.45	**74.8**
	ResNet-18	75.61	76.5	**77.75**
	ResNet-101	77.78	78.9	**80.0**
	DenseNet-121	77.01	79.67	**79.67**

TABLE 4.2: Test errors for the CIFAR experiments.

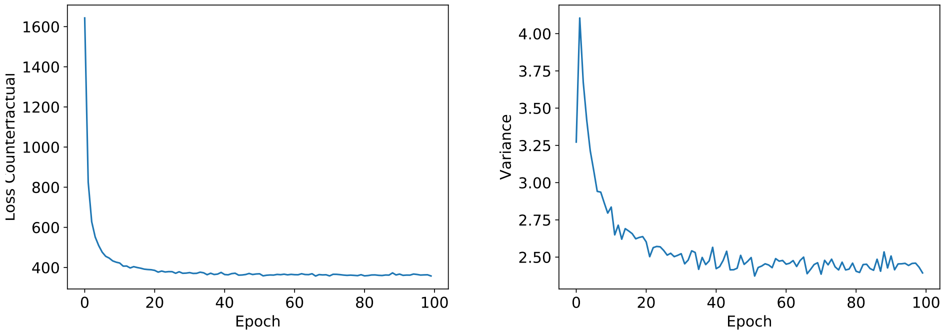
on the baselines by a margin. As seen in Fig. 4.4, the variance is also reduced during training which, as discussed in Theorem 4, is an indication of the convergence of counterfactual training. As observed, the values of $\mathbf{u}^v$ decreases over time to find the samples that are harder to predict. Our experiments thus indicate that our approach provides improvements to even unimodal problems.

4.6.2 Visual Question Answering

Visual Question Answering is used to evaluate our model with two datasets: VQA-CP (Agrawal et al. 2018) and VQA v2 (Goyal et al. 2019a). VQA-CP is specifically designed to measure the generalisation ability of VQA models. Since our model learns how the data is generated, we expect it to be particularly robust towards bias. We follow the same training and evaluation protocol as (Anderson et al. 2018b). For each model, we report the standard VQA accuracy metric (Agrawal et al. 2017). In



(A) Loss PreResNet18-CIFAR10 (B) Loss Variance PreResNet18-CIFAR10



(C) Loss PreResNet18-CIFAR100 (D) Loss Variance PreResNet18-CIFAR100

FIGURE 4.6: CIFAR results

this experiment, we interpolate the word/visual embeddings rather than actual inputs to generate counterfactuals.

We implemented our approach on top of the original UpDn system (Anderson et al. 2018b). The base system utilizes a Faster R-CNN head in conjunction with a ResNet-101 base network as the object detection module. For the VQA v2 experiment we utilize the ResNet-152 for detection. The detection head is pre-trained on the Visual Genome dataset. UpDn takes the final detection outputs and performs non-maximum suppression (NMS) for each object category using an IoU threshold of 0.7. Then, the convolutional features for the top 36 objects are extracted for each image as the visual features. For question embedding, we perform standard text pre-processing and tokenization. In particular, questions are first converted to lower case and then trimmed to a maximum of 14 words, and the words that appear less than 5 times are replaced with an "<unk>" token. We use GloVe embeddings and subsequently GRU for VQA-CP and LSTM for VQA v2A to sequentially process the word vectors and produce a sentential representation for the pre-processed question.

Model	Overall	Yes/No	Number	Other
Question-Only (Agrawal et al. 2018)	15.95	35.09	11.63	7.11
RAMEN (Shrestha, Kafle, and Kanan 2019)	39.21	-	-	-
BAN (Kim, Jun, and Zhang 2018)	39.31	-	-	-
MuRel (Cadene et al. 2019)	39.54	42.85	13.17	45.04
UpDn (Anderson et al. 2018b)	39.74	42.27	11.93	46.05
UpDn+Q-Adv+DoE (Ramakrishnan, Agrawal, and Lee 2018)	41.17	65.49	15.48	35.48
UpDn+C Images	41.01	44.61	12.38	46.11
UpDn+C Questions	40.62	42.33	14.17	48.32
UpDn+C (Q+I)	**42.12**	45.72	12.45	**48.34**

TABLE 4.3: State-of-the-art results on VQA-CP test. **UpDn+C** indicates our approach based on UpDn baseline. **(Q+I)** denotes both question and images are intervened.

Model	Overall
Question-Only (Agrawal et al. 2018)	25.98
BAN (Kim, Jun, and Zhang 2018)	**69.08**
MuRel (Cadene et al. 2019)	65.14
UpDn (Anderson et al. 2018b)	63.48
UpDn+Q-Adv+DoE (Ramakrishnan, Agrawal, and Lee 2018)	62.75
Pythia (Singh et al. 2019)	68.49
Pythia+C	68.77

TABLE 4.4: Performance of our approach on VQA v2 validation. **Pythia+C** is our counterfactual implementation of (Singh et al. 2019).

In Table 4.3, we compare our approach consisting of our baseline architecture trained with additional counterfactual training on VQA-CP against the state-of-the-art. To be fair, we only report approaches that use the visual features from (Anderson et al. 2018b). Our approach improves the baseline more than 2 percentage point beyond UpDn+Q-Adv+DoE which regularizes the model for better performance. In addition, our approach gains most from the "other" category that encompass the most valuable improvement indicating better reasoning about the answers. We should note that since our approach is architecture agnostic, we expect more against better baselines.

Ablation Study on Modality Intervention: In Table 4.3, we perform an ablation study of learning to intervene in multimodal problems by only either inferring $\mathbf{u}^q$ (i.e. intervention in the question) or $\mathbf{u}^v$ (intervention in the images). Even though intervening in both $\mathbf{u}^q, \mathbf{u}^v$ improves performance, counterfactual questions lead to better "number" results indicating strong bias in the baseline for questions with number answers.

Smaller Training Sets: As shown in Fig. 4.7, when the number of training instances

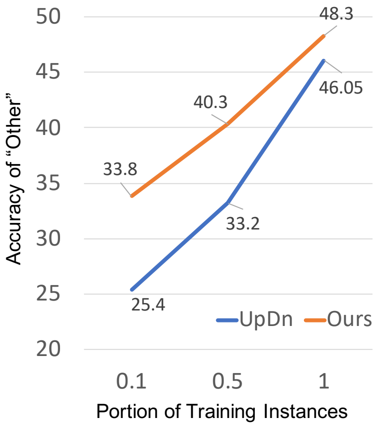
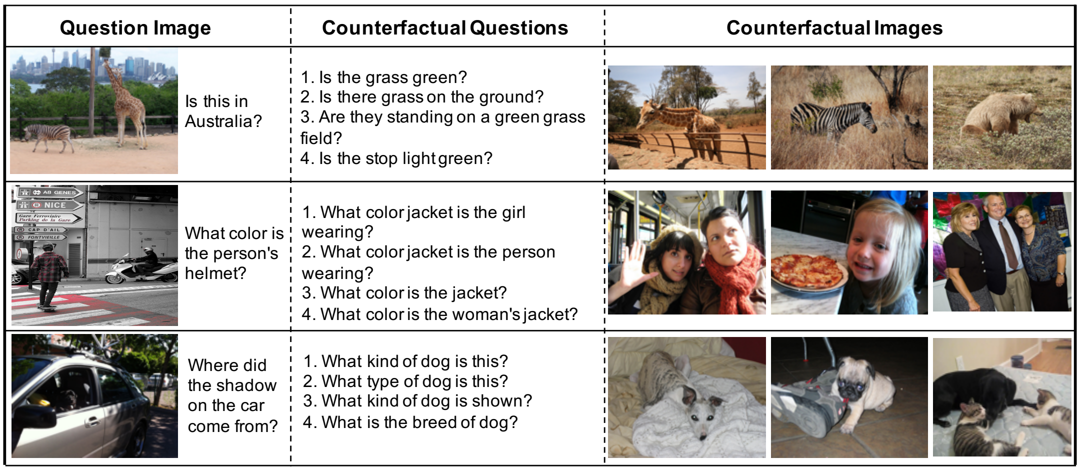


FIGURE 4.7: The performance of our approach vs. the baseline using fraction of the training data.



Question Image		Counterfactual Questions	Counterfactual Images
	Is this in Australia?	1. Is the grass green? 2. Is there grass on the ground? 3. Are they standing on a green grass field? 4. Is the stop light green?	
	What color is the person's helmet?	1. What color jacket is the girl wearing? 2. What color jacket is the person wearing? 3. What color is the jacket? 4. What color is the woman's jacket?	
	Where did the shadow on the car come from?	1. What kind of dog is this? 2. What type of dog is this? 3. What kind of dog is shown? 4. What is the breed of dog?	

FIGURE 4.8: Given the image-question pair in the first column, the closest instances of the questions (in second column) and images (in the third column) are found from the VQA v2 test dataset corresponding to the generated counterfactuals (using the exogenous variables).

is smaller our approach achieves significantly better performance compared to the baseline. This is due to our approach being able to exploit the alternative instances with counterfactuals.

Impact on VQA v2: We use the standard VQA v2 dataset (Goyal et al. 2019a) by following the implementation in (Singh et al. 2019; Singh et al. 2018). Since by exploiting statistical regularities in this dataset it is easier to achieve better performance, large gains are not expected. As shown in this section, counterfactual samples improve the accuracy in VQA-CP, while marginally improving in VQA v2 compared to its baseline. It is interesting to note that in adversarial training in UpDn+Q-Adv+DoE, the performance drops in VQA v2 indicating the same phenomenon.

In Fig. 4.8 we show samples of the counterfactuals for the given question-image pairs

	d_T Lower is better			d_Δ Higher is better		
Model	T_{-10}	T_{-30}	T_{-50}	T_{-10}	T_{-30}	T_{-50}
PACMAN (Das et al. 2018a)	1.39	4.98	9.33	-0.45	0.49	1.66
GRU (Das et al. 2018a)	0.74	3.99	8.74	0.20	1.48	2.26
GRU+C	**0.67**	**3.90**	**8.47**	**0.26**	**1.57**	**2.52**

TABLE 4.5: Evaluation metrics for EQA navigation. Spawning the agent 10, 30, or 50 steps away from the target location, d_0 shows the distance between these initial locations and the target location, while d_T reveals the distance of the final locations and the target ones by starting from these initial location and using the model for the maximum of 100 steps. Finally, $d_\Delta = d_T - d_0$ measures the overall progress of the agent towards the target. GRU+C is ours.

from the test set. These samples are generated by following Eq. (5.9) (i.e. randomly sampling another question-image pair and interpolating the embeddings using the samples from the posterior) and subsequently finding the closest instances (either question or image with smallest Euclidean distance in the embedding space) in the test set. As observed, some of the questions are reasonable alternatives to the ones asked and conversely, the given question can be asked of the counterfactual images showing that our approach successfully generates alternatives.

4.6.3 Vision and Language Navigation

Embodied Question Answering (EQA) (Das et al. 2018a) is proposed as a novel variant of VQA where an agent is spawned at a random location in a 3D environment and asked a question for which the answer requires exploration in the environment. We closely follow the instructions of (Das et al. 2018a) for the experimental setup. Similar to VQA, the agent is tasked with utilizing both vision (i.e. the input egocentric RGB image from the robot's camera) and language (i.e. the given instructions) to answer questions. However, a distinct feature of this task is, unlike VQA, the final answer is produced after the agent takes a finite number of intermediate actions (i.e. navigation by choosing the action right, left, straight, stop at each step for which we use a 2-layer GRU to predict). During training, each batch contains a random environment, a question in that environment and its corresponding answer along with the path to reach the corresponding location in the target room.

	Val Seen				Val unseen			
Model	**NL↓**	**NE↓**	**SR↑**	**SPL↑**	**NL↓**	**NE↓**	**SR↑**	**SPL↑**
Seq-to-Seq (Anderson et al. 2018c)	11.3	6.01	38.6	-	8.4	7.81	21.8	-
Speaker-Follower (Fried et al. 2018)	-	4.86	52.1	-	-	7.07	31.2	-
Co-Grounding (Ma et al. 2019a)	-	3.65	65.0	0.56	-	6.07	42.0	0.28
BC (Tan, Yu, and Bansal 2019)	9.9	5.34	50.2	0.48	9.5	6.10	42.6	0.40
BC+C*	9.9	**5.37**	**50.2**	**0.48**	9.0	**6.23**	**44.2**	**0.41**
BC+RL (Tan, Yu, and Bansal 2019)	10.3	4.65	55.8	0.53	9.7	5.73	44.9	0.41
BC+RL+C*	10.4	**4.43**	**57.1**	**0.55**	9.5	**4.42**	**48.1**	**0.45**

TABLE 4.6: Evaluation metrics for R2R Navigation. **Navigation Length (NL)**: The distance (in meters) that the agent has travelled during an episode. **Navigation Error (NE)**: The difference (in meters) between the agent's final position and the target location. **Success Rate (SR)**: The percentage at which the agent ends up less than 3 meters away from the target location. **Success weighted by Path Length (SPL)**: The success rate weighted by the inverse ratio of the traversed trajectory length to the length of the shortest path.. ↑ shows higher is better, while ↓ means lower is better.. Rows indicated by * our results from Behaviour Cloning and Counterfactuals (BC+C) and Behaviour Cloning, Reinforcement Learning and Counterfactuals (BC+RL+C).

In our approach, we intervene in both the image and question embeddings using a randomly sampled environment and question to generate counterfactual instances in Eq. (5.9). We set the prior for the exogenous variables $\mathbf{u}^q$ and $\mathbf{u}^v$ to Beta$(0.75, 0.75)$. We trained the model based on shortest path trajectories to target objects inside 640 houses (total 6,912 questions) for 30 epochs and then evaluated it on 57 unseen environments during the inference. In particular we consider three cases which correspond to being $10, 30$ and 50 steps away from the target room, with the distance corresponding to $0.94, 5.47$ and 10.99 respectively. In this experiment we measure the number of correct intermediate steps that the agent correctly takes to increase its proximity to the room with the answer. The results are shown in Table 4.5. As is shown, our approach of allowing the agent to contemplate counterfactual questions and images enables the robot to travel closer to the target room and improves generalisation. This further illustrates our approach's success in improving generalisation in various tasks and input-output alternatives. Note that in this task the output is a sequence of actions to be predicted (before the answer).

Room-to-Room (R2R) is one of the hottest tasks introduced recently in the area of Vision and Language Navigation (VLN) (Anderson et al. 2018c). In this task, the agent, which is spawned in a random location inside a photo-realistic house, should understand and follow a natural language instruction and reach a goal location by sequentially choosing the right directions. Similar to EQA, the inputs are both vision (here, panoramic views at each step) and language (a human-crafted instruction that can lead the agent to the target location).

We applied our approach on this task by intervening both visual and textual embeddings. To that end, we generate counterfactual samples by choosing random instances and following the Eq. (5.9) with exogenous variables set to Beta$(0.75, 0.75)$. Following the methods proposed in (Tan, Yu, and Bansal 2019), we applied our approach on both Behavioural Cloning (BC) and Behavioural Cloning+Reinforcement Learning (BC+RL) methods. The results exhibited in Table 5.1 show the superior performance of our model, empowered with counterfactual observation and reasoning, in both seen and unseen environments.

4.7 Conclusion

The tendency to focus on spurious correlations in the training data is one of the key factors limiting the practical application of modern machine learning methods. We have shown that this failure to generalise can, in part, be tackled by generating a set of counterfactual examples to augment the training data. This is motivated by the success that the counterfactual approach has had in causal reasoning. We have demonstrated the effectiveness and generality of the proposed approach on a wide variety of problems including multimodal vision-and-language tasks. An additional advantage of the method is that the sample generation strategy relieves the conditional independence assumption of the training data, which is too strong for most real datasets.

Chapter 5

Counterfactual Vision-and-Language Navigation: Unravelling the Unseen

When interacting with humans through the language, the ideal agent is expected to perform well regardless of the changes in the surrounding environment. Additionally, the agent is supposed to perform in environments that have not been seen during the training. In this chapter, we focus on this challenging aspect of training interactive models in the context of vision-and-language navigation (VLN). We cover our proposed approach based on counterfactuals that significantly improves the model's generalisation to unseen environments.

5.1 Overview

The task of vision-and-language navigation (VLN) requires an agent to follow text instructions to find its way through simulated household environments. A prominent challenge is to train an agent capable of generalising to new environments at test time, rather than one that simply memorises trajectories and visual details observed during training. We propose a new learning strategy that learns both from observations and generated *counterfactual* environments. We describe an effective algorithm to generate counterfactual observations on the fly for VLN, as linear combinations of existing environments. Simultaneously, we encourage the agent's actions to remain stable between original and counterfactual environments through our novel training objective – effectively removing spurious features that would otherwise bias the agent. Our experiments show that this technique provides significant improvements in generalisation on benchmarks for Room-to-Room navigation and Embodied Question Answering.

5.2 Introduction

Deep learning has generated significant advances in computer vision and natural language processing. The most striking successes are witnessed on perceptual tasks that essentially amount to pattern matching. A strength of deep learning is its ability to pick up statistical patterns in large labeled datasets. As a flip side, this capacity leads to models that indiscriminately rely on dataset biases and spurious correlations as much as task-relevant features. This limits the generalisation capabilities of learned models and restrict their applicability on complex tasks (e.g. (Geirhos et al. 2019; Jo and Bengio 2017) with images and (Johnson et al. 2017; Ramakrishnan, Agrawal, and Lee 2018; Hudson and Manning 2019; Agrawal et al. 2018) in multimodal tasks). Most successful applications of deep learning rely on settings where the *seen* training data and the *unseen* test data are statistically similar. Yet we argue that better generalisation could be achieved with new training strategies. This is particularly relevant to multimodal, high-level tasks where training examples can only cover a tiny part of the input space.

In this chapter, we propose to *consider the unseen* to learn representations that lead to better generalisation. The method is applied to the task of vision-and-language navigation (VLN, (Thomason et al. 2020; Anderson et al. 2018c; Das et al. 2018a)) which requires relating complex inputs with observations of unseen environments. In VLN, an agent receives instructions in natural language and it must decide on a sequence of actions (e.g. *turn left*, *move forward*, ...) to reach a target location while observing 2D images of its environment. The task is extremely ambitious: the agent must learn to ground language with visual observations, to understand sequences of instructions and high-level actions (e.g. *wait by the door*), to generate navigation plans, etc. The standard approach is to train an agent with a combination of reinforcement learning (Wang et al. 2019a; Tan, Yu, and Bansal 2019) and imitation learning with human-generated examples of instructions and trajectories. These agents can memorise successful sequences of actions and grounding associations but they often fail to apply their capabilities to unseen environments at test time (Tan, Yu, and Bansal 2019). Our intuition is that a mechanism to reason about alternative observations and trajectories during training could help learning robust navigation strategies. We would like to consider, for example, *what would happen if a desk were observed instead of a chair ?*

Various methods have been proposed to improve generalisation in VLN, such as feature and environment dropout (Tan, Yu, and Bansal 2019), fine-tuning based on the exploration of unseen environments (Wang et al. 2019a; Fried et al. 2018) or using beam search (Fried et al. 2018; Ma et al. 2019a). The method we propose is inspired by the framework of counterfactual reasoning (Pearl 2009). Counterfactuals serve to reason about unobserved scenarios and to estimate the effect of an intervention not represented in the data. In the context of VLN, we essentially want to consider during training *what if we observed a different environment*. Throughout this chapter, we call *counterfactuals* training environment examples that we could have observed. We consider the causal model underlying the training environments and introduce an *exogenous* variable that governs their visual features yet is unobserved. We utilise this variable in generating counterfactuals. Intuitively, this exogenous variable captures variations in visual features in the environments that are rather insignificant for the decision making of the agent and can be ignored. At each training iteration, we generate counterfactuals that represent the minimum edit of an existing training data that causes the model to change its action. Thereafter, we formulate a novel objective that encourages the agent to learn from both observed training data and their counterfactuals by explicitly removing the effects of intervention in the agent's policy (see Fig. 5.1). By introducing additional variations in the observations during training, we

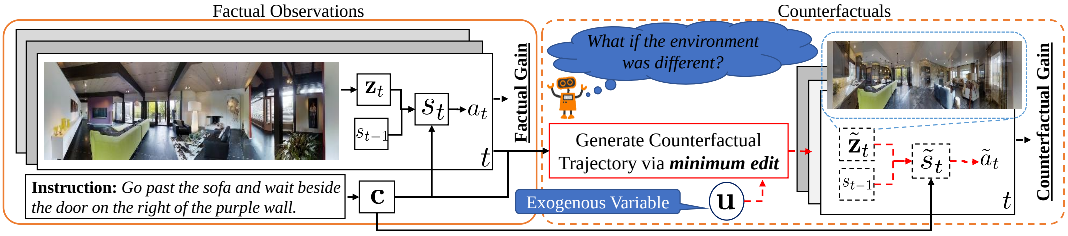


FIGURE 5.1: We seek to improve a VLN agent's capability to generalise to unseen environments at test time. Agents are typically trained by reinforcement and imitation learning, using ground-truth pairs of instructions/trajectories ("factual observations", left). We propose to generate alternative, *counterfactual* training observations with combinations of existing environments. We determine the minimum intervention on the factual data that causes the current model to produce different outputs. We then formulate our novel training objective to best exploit these additional examples and improve its generalisation capabilities (right). The generation process is formalised with a causal model of the data, in which we introduce the interpolation coefficients as an exogenous variable $\mathbf{u}$, effectively modelling an intervention on the environment.

encourages the model to rely less on idiosyncrasies of a given environment, and rather learn a policy that better generalises to unseen environments at test time.

The contributions of this chapter are summarized as follows:

- We propose a novel training strategy for VLN that generates counterfactuals on the fly to account for unseen scenarios. Using both training data and their counterfactuals, we improve agent's capabilities to generalise to new environments at test time.

- We formalise the new procedure with a causal generative view of the data, in which we introduce an exogenous variable representing interpolation coefficients between original training examples. We derive an efficient algorithm to generate counterfactual instances that represent minimum interventions over original examples that cause the model to change its output.

- We implement the technique on top of a VLN agent for both reinforcement and imitation learning. Experiments on benchmarks for Room-to-Room (R2R) navigation (Anderson et al. 2018c) and Embodied Question Answering (Das et al. 2018a) show significant improvements. We reduce the success rate gap between seen and unseen environments in R2R from about 8% to less than 2.5%.

5.3 Related Work

Vision and Language Navigation (VLN) has gained popularity in various forms (instruction following (Anderson et al. 2018c; Chen et al. 2019), object or room probing (Wu et al. 2018; Wortsman et al. 2019), embodied question answering (Das et al. 2018a; Wijmans et al. 2019), vision and language dialogue (Thomason et al. 2020; Nguyen and Daumé III 2019)). Generalisation to unseen environments remains an unsolved challenge, despite techniques like enhanced features and beam search, panorama view (Fried et al. 2018), attention mechanisms (Ma et al. 2019a), and other heuristics (Wang et al. 2019a; Wang et al. 2018; Ma et al. 2019b). Environment Dropout (Tan, Yu, and Bansal 2019) randomly drops visual features to simulate variations in environments. Our approach does not require access to held-out trajectories, which may not be available in other tasks (rather than R2R). Our method can be used in a variety of tasks, as demonstrated with EQA in the experiments.

Principles of **counterfactual reasoning** (Pearl 2009; Charles, Chickering, and Simard 2013) have been applied beyond standard causal inference to augment training in bandit settings (Johansson, Shalit, and Sontag 2016), and in recommendation (Swaminathan and Joachims 2015) and explanation systems (Goyal et al. 2019b). Kaushik et al. (Kaushik, Hovy, and Lipton 2020) proposed a human-in-loop process to augment datasets with counterfactual instances. In reinforcement learning (Buesing et al. 2019; Oberst and Sontag 2019), counterfactuals are used in off-policy settings to improve sample efficiency. Our technique is also related to adversarial training (Goodfellow, Shlens, and Szegedy 2015; Hoffman, Roberts, and Yaida 2019; Xie et al. 2020; Woods, Chen, and Teuscher 2019) in that we generate variations of training examples that cause the current model to switch its predictions. The major difference is that our approach provides alteration to the input, or rather its representations, by a variable that is conditioned on the real training data rather than a simple perturbation.

Using counterfactuals for VLN was explored in Fu et al. 2019 in which adversarial paths that are hard for the policy to navigate are generated. Our approach differs from their adversarial augmentation method in that intervene in visual features rather than focusing on difficult trajectories. Our method, while being simpler, outperforms theirs with almost 10% in success rate.

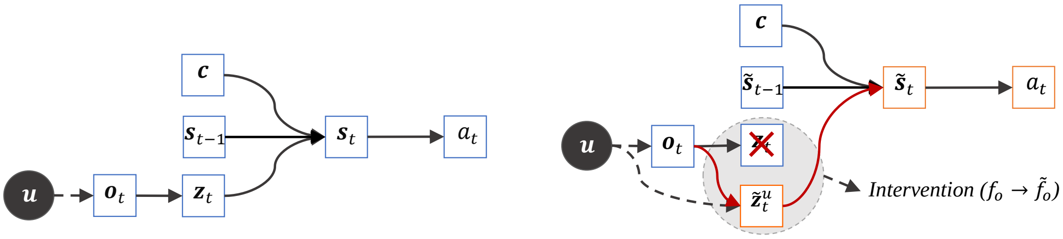
The closest work to this one is Abbasnejad et al. 2020. The authors generate counterfactual data using interpolations for vision-and-language tasks, including visual question answering. The differences with this work are that (1) we only intervene on visual features, (2) we backpropagate the loss in counterfactual environments instead of using it as a change ratio for factual loss calculation, and (3) we explicitly focus on removing the effects of intervention. Our work also extensively focuses on VLN.

In comparison to standard **data augmentation**, our counterfactual instances do not rely on handcrafted or domain-specific rules, and they are generated on the fly. MixUp (Zhang et al. 2018a; Verma et al. 2019a) performs data augmentation with interpolations and label smoothing. Mixup is not directly applicable to VLN since (1) VLN is sequential in nature, (2) an interpolation of state-action from one trajectory to another may lead to catastrophic difference in the objective. Our approach intervenes in the visual features to simulate the agent's behaviour in a counterfactual environment, where the agent still has to follow the same instruction and sequence of actions

5.4 Background on Counterfactuals

We provide a brief background on counterfactuals. Further details can be found in Pearl 2009.

Definition 6 (Structural Causal Model (SCM))**.** A structural causal model $\mathcal{M}$ over variables $\mathbf{X} = \{X_1, \dots, X_n\}$ consists of a set of independent (exogenous) random variables $\mathbf{U} = \{\mathbf{u}_1, \dots, \mathbf{u}_n\}$ with prior distributions $P(\mathbf{u}_i)$ and a set of functions $f_1, \dots, f_n$ such that $X_i = f_i(\mathbf{PA}_i, \mathbf{u}_i)$, where $\mathbf{PA}_i \subset \mathbf{X}$ are parents of X_i. Therefore, the distribution of the SCM, which is denoted $P^{\mathcal{M}}$, is determined by the functions and the prior distributions of exogenous variables.



(A) VLN SCM (B) VLN SCM with interventions

FIGURE 5.2: Structural Causal Model (SCM) of the vision-and-language navigation (VLN). We incorporate an exogenous variable in the SCM that is learned and utilised for reasoning about interventions in the observation.

Inferring the exogenous random variables based on the observations, we can intervene in the observations and inspect the consequences.

Definition 7 (Interventions in SCM)**.** A soft intervention $I = \text{do}\big(X_i := \tilde{f}_i(\tilde{\mathbf{PA}}_i, \mathbf{u}_i)\big)$ is defined as replacing some functions $f_i(\mathbf{PA}_i, \mathbf{u}_i)$ with $\tilde{f}_i(\tilde{\mathbf{PA}}_i, \mathbf{u}_i)$. Additionally, a hard intervention $I = \text{do}(X_i = x)$ is defined as substituting the structural mechanism of X_i with a constant value x. The intervened SCM is indicated as $\mathcal{M}^I$, and, consequently, its distribution is denoted $P^{\mathcal{M};I}$.

The counterfactual inference with which we can answer the "what if" questions will be obtained in the following process:

1. Infer the posterior distribution of exogenous variable $P(\mathbf{U}_i|\mathbf{X} = \mathbf{x})$, where **x** is a set of observations.

2. Replace the prior distribution $P(\mathbf{u}_i)$ with the posterior distribution $P(\mathbf{u}_i|\mathbf{X} = \mathbf{x})$ in the SCM. We denote the resulted SCM as $\mathcal{M}_{\mathbf{x}}$ and its distribution as $P^{\mathcal{M}_{\mathbf{x}}}$

3. Perform an intervention I on $\mathcal{M}_{\mathbf{x}}$ to reach $P^{\mathcal{M}_{\mathbf{x}};I}$.

4. Return the output of $P^{\mathcal{M}_{\mathbf{x}};I}$ as the counterfactual inference.

5.4.1 Counterfactual Vision and Language Navigation

We concentrate on the interventions on the visual observations to improve the generalisation of the model to the unseen environments. Our intuition is that the visual feature extractor functions in VLN usually focus on spurious features in the scene. To that end, by constructing the SCM for VLN and introducing interventions in the training environments, we train models that better generalise to unseen environments.

Fig. 5.2 shows the SCM for VLN at a time-step. The SCM consists of an exogenous variable $\mathbf{u}$ (for observations) and a set of functions that transmit the observation $\mathbf{o}_t$, language instruction $\mathbf{c}$ and previous state $\boldsymbol{s}_{t-1}$ to the next state $\boldsymbol{s}_t$ and, subsequently, to the next action a_t. We intervene in observations by replacing their embedding function $f_{\mathbf{o}}$ with $\tilde{f}_{\mathbf{o}}$. Specifically, after learning exogenous variable $\mathbf{u}$, we replace the latent representation of the observations $\mathbf{z}_t$, to $\tilde{\mathbf{z}}_t^{\mathbf{u}}$.

In Eq. (5) in the chapter, we effectively remove the effect of the intervention leading to an agent that is less biased towards spurious features. For computing the expectation, we could take samples from the posterior $p(\mathbf{u} \mid \tau, \mathbf{c})$ and average using multiple counterfactual trajectories (an MCMC approach). Instead, in the interest of efficiency in Sec 3.3, we take only one instance from the mode of the posterior that alters the navigation policy's output.

5.5 Methodology

5.5.1 Problem Definition

Our task is to train an agent capable of grounding a command, in the form of natural language, to the current visual view and taking suitable actions that lead to the target location. Formally, the agent is given natural language instructions or commands as a sequence of words $\mathbf{c} = [w_1, w_2, .., w_L]$ to be executed in the environment $\mathcal{E}$. We consider all the instructions to be in a set $\mathcal{C}$. The process can be viewed as a Partially Observable Markov Decision Process (POMDP) where a trajectory is a sequence of length T of observation $\mathbf{o}_t$, state $\boldsymbol{s}_t$ and action a_t for each time step t i.e. $\tau = \{\mathbf{o}_1, \boldsymbol{s}_1, a_1, \dots, \mathbf{o}_T, \boldsymbol{s}_T, a_T\}$. The probability of each trajectory given the instruction is[*]

$$\pi_{\boldsymbol{\theta}}(\tau \,|\, \mathbf{c}) \;=\; \prod_{t=1}^{T} p(a_t \,|\, \boldsymbol{s}_t)\; p(\boldsymbol{s}_t \,|\, \boldsymbol{s}_{t-1}, \mathbf{z}_t, \mathbf{c})\; p(\mathbf{z}_t \,|\, \mathbf{o}_t)\ . \tag{5.1}$$

Here, $\pi_{\boldsymbol{\theta}}$ is the agent's *policy* (Unless explicitly mentioned otherwise, $\boldsymbol{\theta}$ represents all parameters which is omitted from the right-hand side probabilities for brevity). In the visual navigation scenario we consider, $\mathbf{o}_t$ as the visual observation of the scene in which the agent is, $\boldsymbol{s}_t$ as a representation of the trajectory history[†] and a_t as the chosen action at time t (e.g. `turn left` or `stop` for when the trajectory is finished). By convention, $\boldsymbol{s}_0$ is a sample from the state prior (e.g. uniform). We denote a latent representation of the visual scene by $\mathbf{z}$ and assume it is obtained using a function $\mathbf{z} = f_{\mathbf{o}}(\mathbf{o})$, e.g. a pretrained CNN for the visual inputs, thus $p(\mathbf{z}_t \,|\, \mathbf{o}_t) = \delta(\mathbf{z} - f_{\mathbf{o}}(\mathbf{o}))$ where δ is the Dirac delta.

Training with imitation learning and reinforcement learning. The common practice in visual navigation is to use a training set $\mathcal{D} = \{(\tau_i, \mathbf{c}_i)\}_{i=1}^{n}$ containing human-provided trajectories and instructions. This training set is used in supervised learning to bootstrap the agent' behaviour through cloning human's actions. In addition, reinforcement learning is used so that the agent learns from the environment's feedback. The training procedure optimises the following objective (Tan, Yu, and Bansal 2019):

$$\max_{\boldsymbol{\theta}} \;\underbrace{\mathbb{E}_{(\tau,\mathbf{c})\sim\mathcal{D}}\big[\log \pi_{\boldsymbol{\theta}}(\tau \,|\, \mathbf{c})\big]}_{\mathcal{G}_{\mathrm{IL}}(\boldsymbol{\theta})} + \lambda\, \underbrace{\mathbb{E}_{\mathbf{c}\sim\mathcal{C}}\big[\,\mathbb{E}_{\tau\sim\pi_{\boldsymbol{\theta}}(\tau \,|\, \mathbf{c})}[R(\tau)]\,\big]}_{\mathcal{G}_{\mathrm{RL}}(\boldsymbol{\theta})}\ . \tag{5.2}$$

The first term $\mathcal{G}_{\mathrm{IL}}(\boldsymbol{\theta})$ is a simple log-likelihood of human-provided examples using Eq. (5.1) (imitation learning). The second term $\mathcal{G}_{\mathrm{RL}}(\boldsymbol{\theta})$ corresponds to the execution of the policy in the environment and receiving a reward $R(\tau)$. The hyperparameter λ serves to balance the importance of imitation learning versus reinforcement learning.

---

[*] We model $\pi_{\boldsymbol{\theta}}$ as a recurrent model. For the language command, we use a separate recurrent model.

[†] We consider the hidden state of the agent's policy as $\boldsymbol{s}_t$.

The reward captures the agent's success in navigating the environment. In a Room-to-Room navigation task, the reward is a combination of a large positive number for reaching the target location at the end of each episode, and a small positive/negative number for reducing/increasing the distance to that location at each step. To update the parameters of the policy during RL, we employ an on-policy algorithm such as actor-critic (Mnih et al. 2016b).

5.5.2 Counterfactual Formulation in VLN

The state variable $\boldsymbol{s}$ ideally is the representation of the history of observations and actions. The final decision of the agent is taken conditioned on this variable and as such is of great importance. However, as is common with other multi-modal problems (e.g. VQA (Agrawal et al. 2018; Ramakrishnan, Agrawal, and Lee 2018)) this variable captures particular biases and regularities in the input and may even ignore important patterns which significantly limits the generalisation ability of the agent. To remedy the situation, we consider an exogenous variable that intervenes the observations. By introducing and reasoning about this variable, the agent is encouraged to consider alternative observations and representations. In addition, the agent obtains the capacity to reason about "what if" the observations were different.

To that end, we consider the counterfactual distribution of the trajectory where each observation is replaced by its intervened alternative $\tilde{\mathbf{z}}_t^{\mathbf{u}}$:

$$\tilde{\pi}_{\boldsymbol{\theta}}(\tilde{\tau} \,|\, \mathbf{c},\, \mathbf{u}) \;=\; \prod_{t=1}^{T} p(a_t \,|\, \tilde{\mathbf{s}}_t)\; p(\tilde{\mathbf{s}}_t \,|\, \tilde{\mathbf{s}}_{t-1},\, \tilde{\mathbf{z}}_t^{\mathbf{u}},\, \mathbf{c}). \tag{5.3}$$

In this distribution, the conditional dependence on the scene observations $\mathbf{o}_t$ is suppressed because of the intervention. We denote with $\tilde{\tau}$ the trajectories obtained by replacing a given embedding of the visual scene $\mathbf{z}_t$ with its counterfactual $\tilde{\mathbf{z}}_t^{\mathbf{u}}$ based on the influence of $\mathbf{u}$. Imagine that the agent observes a chair that represents an obstacle to be avoided. A counterfactual situation would ask, for example "what if the agent observed a table?". The exogenous variable is conditioned on the factual trajectories observed in the training set. The expectation with respect to the exogenous variable serves to consider a whole range of possible alternatives. The expected reward for counterfactual trajectories $\tilde{\mathcal{G}}_{\text{RL}}(\boldsymbol{\theta})$ (to be compared with $\mathcal{G}_{\text{RL}}(\boldsymbol{\theta})$ of Eq. (5.2)), is obtained from the states intervened based on the exogenous variable $\mathbf{u}$:

$$\begin{aligned}\tilde{\mathcal{G}}_{\text{RL}}(\boldsymbol{\theta}) &:= \mathbb{E}_{(\tau,\mathbf{c})\sim\mathcal{D}}\Big[\,\mathbb{E}_{\mathbf{u}\sim p(\mathbf{u}\,|\,\tau,\,\mathbf{c})}\big[\,\mathbb{E}_{\tilde{\tau}\sim\tilde{\pi}_{\boldsymbol{\theta}}(\tilde{\tau}\,|\,\mathbf{c},\,\mathbf{u})}[R(\tilde{\tau})]\,\big]\,\Big] \\ \tilde{\mathcal{G}}_{\text{IL}}(\boldsymbol{\theta}) &:= \mathbb{E}_{(\tau,\mathbf{c})\sim\mathcal{D}}\Big[\,\mathbb{E}_{\mathbf{u}\sim p(\mathbf{u}\,|\,\tau,\,\mathbf{c})}\big[\,\log\tilde{\pi}_{\boldsymbol{\theta}}(\tilde{\tau}\,|\,\mathbf{c},\,\mathbf{u})\,\big]\,\Big]\end{aligned} \tag{5.4}$$

We detail $p(\mathbf{u}\,|\,\tau,\mathbf{c})$ and how to generate counterfactuals using $\tilde{\pi}_{\boldsymbol{\theta}}(\tilde{\tau}\,|\,\mathbf{c},\,\mathbf{u})$ in Section 5.5.3.

The differences between $\mathcal{G}_{\text{RL}}(\boldsymbol{\theta})$ and $\tilde{\mathcal{G}}_{\text{RL}}(\boldsymbol{\theta})$ as well as between $\mathcal{G}_{\text{IL}}(\boldsymbol{\theta})$ and $\tilde{\mathcal{G}}_{\text{IL}}(\boldsymbol{\theta})$ correspond to the Conditional Average Treatment Effect (CATE) (Johansson, Shalit, and Sontag 2016). These differences reflect how the intervention influences the reward and log-likelihood. They are defined as

$$\Delta_d = \mathcal{G}_{\text{IL}}(\boldsymbol{\theta}) - \tilde{\mathcal{G}}_{\text{IL}}(\boldsymbol{\theta}) \qquad \text{and} \qquad \Delta_\tau = \mathcal{G}_{\text{RL}}(\boldsymbol{\theta}) - \tilde{\mathcal{G}}_{\text{RL}}(\boldsymbol{\theta})\,. \tag{5.5}$$

We want to optimise our agent such that, after learning from the training set, performs similarly when faced with unobserved alternative scenarios. In other words, we want Δ_τ and Δ_d to be small. This effectively reduces the influence of interventions and as such discourages bias to spurious features. We add, to the objective of Eq. (5.2), constraints on the magnitude of Δ_d and Δ_τ:

$$\max_{\boldsymbol{\theta}} \quad \mathcal{G}_{\text{IL}}(\boldsymbol{\theta}) + \lambda\,\mathcal{G}_{\text{RL}}(\boldsymbol{\theta}) \quad \text{s.t.} \quad \Delta_\tau \le \epsilon_\tau \quad \text{and} \quad \Delta_d \le \epsilon_d\,, \tag{5.6}$$

with ϵ_d and ϵ_τ small constants. Introducing the Lagrange multipliers α and β, we have

$$\max_{\boldsymbol{\theta}} \quad (1-\alpha)\,\mathcal{G}_{\text{IL}}(\boldsymbol{\theta}) \;+\; \alpha\,\tilde{\mathcal{G}}_{\text{IL}}(\boldsymbol{\theta}) \;+\; (\lambda-\beta)\,\mathcal{G}_{\text{RL}}(\boldsymbol{\theta}) \;+\; \beta\,\tilde{\mathcal{G}}_{\text{RL}}(\boldsymbol{\theta})\,. \tag{5.7}$$

We assume $\beta = \alpha\lambda$ and $(1-\alpha) > 0$ for simplicity, which gives the final objective:

$$\max_{\boldsymbol{\theta}} \underbrace{\Big(\mathcal{G}_{\mathrm{IL}}(\boldsymbol{\theta}) + \lambda\,\mathcal{G}_{\mathrm{RL}}(\boldsymbol{\theta})\Big)}_{\text{Original navigation}} + \tfrac{\alpha}{(1-\alpha)} \underbrace{\Big(\tilde{\mathcal{G}}_{\mathrm{IL}}(\boldsymbol{\theta}) + \lambda\,\tilde{\mathcal{G}}_{\mathrm{RL}}(\boldsymbol{\theta})\Big)}_{\text{Counterfactual navigation}}. \tag{5.8}$$

Technically, when increasing $\alpha/(1-\alpha)$, we choose to give more weight to what could have been seen (variations in the environment) rather than maximising the gain. Therefore, when the trajectories are longer we need smaller $\alpha/(1-\alpha)$ which intuitively allows the model to focus on correct actions at each state rather than variations that could have been observed. Note, learning longer trajectories are generally harder and a small mistake has more significant impact. This novel objective is used with the counterfactuals, of which we next discuss the generation.

5.5.3 Counterfactual Distribution Learning and Generation

Computing Eq. (5.4) hinders on: (1) the distribution of the counterfactual trajectories given the intervention by exogenous variable $\tilde{\pi}_{\boldsymbol{\theta}}(\tau|\mathbf{u},\mathbf{c})$, (2) the conditional of the exogenous $p(\mathbf{u}|\tau,\mathbf{c})$ given the observed trajectory-instruction pair from data, and (3) combining (1) and (2) to have the probability of the counterfactual trajectory as $\tilde{\pi}_{\boldsymbol{\theta}}(\tau\,|\,\mathbf{c}) = \mathbb{E}_{p(\mathbf{u}\,|\,\tau,\,\mathbf{c})}[\tilde{\pi}_{\boldsymbol{\theta}}(\tau\,|\,\mathbf{c},\,\mathbf{u})]$. Here, $\mathbf{u}$ is marginalised out to remove the impact of the intervention or spurious features.

1. **Sampling from $\tilde{\pi}_{\boldsymbol{\theta}}(\tau|\mathrm{c},\mathbf{u})$:** To sample a counterfactual trajectory, we first sample a pair of real trajectories from the observations such that at least one has the language instruction, i.e. $\{(\tau,\mathbf{c}),(\tau',\mathbf{c}')\} \sim \mathcal{D}$. Subsequently, we choose the counterfactual visual features to be a linear interpolation. Given a sample $\mathbf{u} \in [0,1]^d$ (d being the dimensionality of $\mathbf{z}$) with slight abuse of notation, we have:

$$\begin{aligned} \tilde{\tau} = &\{\tilde{\mathbf{z}}_0^{\mathbf{u}}, \tilde{\mathbf{s}}_0, a_0, \ldots, \tilde{\mathbf{z}}_T^{\mathbf{u}}, \tilde{\mathbf{s}}_T, a_T\} \sim \tilde{\pi}_{\boldsymbol{\theta}}(\tau|\mathbf{u},\mathbf{c}), \quad \tilde{\mathbf{z}}_t^{\mathbf{u}} = \mathbf{u} \odot \mathbf{z}_t + (\mathbf{1}-\mathbf{u}) \odot \mathbf{z}'_t, \\ &\text{with} \quad \mathbf{z}_t = f_{\mathbf{o}}(\mathbf{o}_t)\,, \quad \mathbf{z}'_t = f_{\mathbf{o}}(\mathbf{o}'_t), \quad \mathbf{o}_t \in \tau\,, \quad \mathbf{o}'_t \in \tau'. \end{aligned} \tag{5.9}$$

We use $\odot$ to represent an element-wise product. When the length of the second trajectory τ' is shorter, we choose to repeat its final visual features for interpolation. Alternative approaches such as generative adversarial networks (Goodfellow et al. 2014) could be employed, albeit our simple option presents a clear advantage in computational efficiency.

2. **Exogenous variable's distribution $p(\mathbf{u} \,|\, \tau, \mathbf{c})$:** Given the prior $p(\mathbf{u})$, we have $p(\mathbf{u} \,|\, \tau, \mathbf{c}) \propto p(\mathbf{u})\tilde{\pi}_{\boldsymbol{\theta}}(\tau \,|\, \mathbf{c}, \mathbf{u})$ as the posterior. It is easy to see that with our definition in Eq. (5.9), when $\mathbf{u} = 1$ we uncover $\pi_{\boldsymbol{\theta}}(\tau \,|\, \mathbf{c})$ in Eq. (5.1). In other words, $\mathbf{u} = 1$ provides the max-likelihood since that gives rise to an observed trajectory. We consider a Beta distribution for the prior.

3. **Finding minimum interventions that change the agent's decision:** Having (1) and (2) we can sample a counterfactual trajectory $\tilde{\pi}_{\boldsymbol{\theta}}(\tau \,|\, \mathbf{c})$ (with $\mathbf{u}$ marginalised out). One can resort to MCMC or a variational lower bound to sample the most likely counterfactual. However, in the interest of efficiency and simplicity, we choose the exogenous variable with the highest likelihood that produces the most likely counterfactual. In other words, we seek the minimum intervention (i.e. minimum edit) that changes the agent's decision (remember, we want our counterfactuals to be very different from observations). Since changing the agent's decision may lead to a different route in the environment, we additionally constrain the counterfactual trajectory to have the same instructions. Given a training example $(\mathbf{c}, \tau)$, the following optimisation identifies such an intervention parametrised by $\mathbf{u}$ (note $\tilde{\tau}$ is the counterfactual of τ):

$$\begin{aligned} &\max_{\mathbf{u} \in [0,1]^d} \quad p(\mathbf{u} \,|\, \tau, \mathbf{c}) \;+\; \log p(\mathbf{c} \,|\, \tilde{\tau}, \boldsymbol{\phi}) \\ &\quad \text{s.t.} \quad a'_t \neq a_t \;\forall\, t \qquad \text{with} \quad a'_t \;=\; \arg\max_{a_t} p(a_t \,|\, \tilde{\mathbf{s}}_t)\, p(\tilde{\mathbf{s}}_t \,|\, \tilde{\mathbf{s}}_{t-1}, \tilde{\mathbf{z}}^{\mathbf{u}}_t, \mathbf{c})\;. \end{aligned} \tag{5.10}$$

The second term in Eq. (5.10) measures how likely an instruction is for a trajectory for which we utilise the speaker model of (Fried et al. 2018) with parameters $\boldsymbol{\phi}$. The optimisation of Eq. (5.10) is too expensive to perform for every training

Algorithm 2: Training of a VLN agent through IL and RL, with factual data (original training set) and counterfactual observations (generated instances).

```
Inputs: dataset $\mathcal{D}$, initial policy parameters $\boldsymbol{\theta}^0$, learning rate $\xi_u, \xi_\theta$
for $i = 1$ to max_iterations do
    Pick a sample from the dataset $(\tau, \mathbf{c}) \sim \mathcal{D}$
    Generate exogenous variable from the prior: $\mathbf{u}^0 \sim p(\mathbf{u})$
    Pick another sample from the dataset $(\tau', \mathbf{c}') \sim \mathcal{D}$
    // use Eq. (5.11) to get the counterfactual trajectory
    for $j$ to $N$ do
        $\tilde{\tau} = \{\tilde{\mathbf{z}}^{\mathbf{u}}_0, \tilde{\mathbf{s}}_0, a_0, \dots, \tilde{\mathbf{z}}^{\mathbf{u}}_T, \tilde{\mathbf{s}}_T, a_T\}, \;\; \tilde{\mathbf{z}}^{\mathbf{u}}_t = \mathbf{u} \odot \mathbf{z}_t + (\mathbf{1} - \mathbf{u}) \odot \mathbf{z}'_t$
            // Eq. (5.9)
        $\mathbf{u}^{j+1} =$
        $\mathbf{u}^j + \xi_u \nabla_{\mathbf{u}} \Big( \|\mathbf{u}\| + \log p(\mathbf{c}|\tilde{\tau}, \boldsymbol{\phi}) - \gamma \sum_{t=1}^{T} \big( \log p(a_t|\tilde{\mathbf{s}}_t) + \log p(\tilde{\mathbf{s}}_t|\tilde{\mathbf{s}}_{t-1}, \tilde{\mathbf{z}}^{\mathbf{u}}_t, \mathbf{c}) \big) \Big)$
    $g_{\text{IL}} = \log \pi_{\boldsymbol{\theta}}(\tau \,|\, \mathbf{c}) + \frac{\alpha}{1-\alpha} \log \tilde{\pi}_{\boldsymbol{\theta}}(\tilde{\tau} \,|\, \mathbf{c})$   // imitation learning gain
    Given the instruction $\mathbf{c}$, rollout trajectories $\tau_{\text{rl}}$ and $\tilde{\tau}_{\text{rl}}$ from the current
      navigation policy without and with interventions respectively
    $g_{\text{RL}} = \mathbb{E}_{\tau_{\text{rl}} \sim \pi_{\boldsymbol{\theta}}(\tau_{\text{rl}} \,|\, \mathbf{c})}[R(\tau_{\text{rl}})] + \frac{\alpha}{1-\alpha} \mathbb{E}_{\tilde{\tau}_{\text{rl}} \sim \tilde{\pi}_{\boldsymbol{\theta}}(\tilde{\tau}_{\text{rl}} \,|\, \mathbf{c})}[R(\tilde{\tau}_{\text{rl}})]$   // RL gain
    $\boldsymbol{\theta}^i = \boldsymbol{\theta}^{i-1} + \xi_\theta \nabla_{\boldsymbol{\theta}} \big( g_{\text{IL}} + \lambda g_{\text{RL}} \big)$   // update based on Eq. (5.8)
```

trajectory. We note that the first term is maximised when $\mathbf{u}$ is close to one, as such a relaxed version by turning the constraint into an extra term in the objective is devised:

$$\max_{\mathbf{u} \,\in\, [0,1]^d} \quad \|\mathbf{u}\| \;+\; \log p(\mathbf{c} \,|\, \tilde{\tau}, \boldsymbol{\phi}) \;-\; \gamma \sum_{t=1}^{T} \Big(\log p(a_t \,|\, \tilde{\mathbf{s}}_t) \;+\; \log p(\tilde{\mathbf{s}}_t \,|\, \tilde{\mathbf{s}}_{t-1}, \tilde{\mathbf{z}}^{\mathbf{u}}_t, \mathbf{c}) \Big)\,, \tag{5.11}$$

where γ is a hyper-parameter. The first two terms in this equation ensure the intervention is minimal and the counterfactual trajectory is most likely to follow the same instructions. The constraint, on the other hand, finds the counterfactual trajectory by fooling the current policy.

A summary of the whole training algorithm is provided in Algorithm 2.

5.6 Experiments

To show the effectiveness of our counterfactual contemplation approach we applied it to both Room-to-Room (R2R) navigation and Embodied Question Answering (EQA).

In all of our experiments, we only intervene in the visual features as discussed in Sec. 5.5.3. We set the prior $p(\mathbf{u})$ to Beta$(0.75, 0.75)$, and use 5 interactions to optimise Eq. (5.11) with the learning rate set to 0.1. Using grid search, we concluded $\gamma = 0.1$ provides best results.

Following Algorithm 2, the learning process of exogenous variables $\mathbf{u}$ for two samples picked from the dataset ($\{(\tau, \mathbf{c}), (\tau', \mathbf{c}')\} \sim \mathcal{D}$) is as follows:

1. Repeat the last observation of τ' to be the same length as τ.
2. Sample $\mathbf{u}$ from the prior distribution Beta$(0.75, 0.75)$.
3. Generate counterfactual visual features using $\mathbf{u}$ and based on Eq. 5.9.
4. Feed the counterfactual trajectory into Speaker and Navigator.
5. Update $\mathbf{u}$ based on Eq. 5.11 with learning rate of 0.1.
6. Repeat steps 3 to 5 for N iterations ($N = 5$ in the experiments).

5.6.1 Room-to-Room Navigation

Dataset: Room-to-Room (R2R) (Anderson et al. 2018c) is a dataset of natural language instructions for indoor navigation collected using Amazon Mechanical Turk (AMT) and employing a simulator based on Matterport3D environments (Chang et al. 2017). The training is based on $14,025$ pairs of instruction-visual path in 61 environments. The validation is done in two settings: (1) *seen* where the environment is from the training set but the instructions are not and (2) *unseen* where both the instructions and the visual observations are never seen by the agent.

Implementation details: We closely follow the experiment setup of (Tan, Yu, and Bansal 2019) where the visual observations consists of the features extracted using the pretrained ResNet-152 (He et al. 2016) from the egocentric panoramic view of the agent. Following the approach proposed in (Fried et al. 2018), our *speaker* is a sequence-to-sequence model which evaluates the likelihood of an instruction for a trajectory.

Our navigation policy is a attention encoder-decoder network that encodes the navigation history conditioned on the instruction and decodes the next direction that the agent should follow. To have a fair comparison and show the effectiveness of our approach, we closely follow the implementation proposed by Fried et al. 2018 and evolved in Tan, Yu, and Bansal 2019. Our encoder is a recurrent neural network:

$$\boldsymbol{h}_i^e = f^e(f^w(w_i), \boldsymbol{h}_{i-1}^e), \tag{5.12}$$

where f^w represents an embedding layer, f^e is a bidirectional LSTM and $\boldsymbol{h}_i^e$ is the latent representation vector for word i in the instruction ($\boldsymbol{h}_i^e \in \mathbb{R}^{512}$), which is obtained from the concatenation of forward and backward layers of the LSTM.

We calculate the attention over a collection of V values ($\mathbf{v}_i$) with respect to a key vector ($\mathbf{k}$) as:

$$\begin{aligned} \alpha, \textbf{att} &= \text{Attention}(\mathbf{k}, \{\mathbf{v}_i\}_{i=1}^V), \qquad (5.13) \\ \text{with} \quad \alpha_i &= \text{Softmax}(\mathbf{v}_i^\intercal \mathbf{W} \mathbf{k}_i), \quad \textbf{att} = \sum_{i=1}^{V} \alpha_i \mathbf{v}_i, \end{aligned}$$

where W are the parameters to be learned, α_i is the weight of i-th value item and **att** is the attentive feature vector.

Our decoder is an attentive RNN:

$$\_, \hat{\mathbf{z}}_t = \text{Attention}^v(\boldsymbol{h}_{t-1}^d, \{\mathbf{z}_i^t\}_{i=1}^{36}), \tag{5.14}$$

$$\boldsymbol{h}_t^d = f^d\big([f^a(a_{t-1}); \hat{\mathbf{z}}_t], \boldsymbol{h}_{t-1}^d\big), \tag{5.15}$$

$$\_, \hat{\boldsymbol{h}}_t^d = \text{Attention}^l(\boldsymbol{h}_t^d, \{\boldsymbol{h}_i^e\}_{i=1}^L), \tag{5.16}$$

$$\{p_j\}_{j=1}^N, \_ = \text{Attention}^d(\hat{\boldsymbol{h}}_t^d, \{\mathbf{z}_j^t\}_{j=1}^N), \tag{5.17}$$

where $\mathbf{z}_i^t$ is the concatenation of 2048-dimensional visual feature vector (extracted from a pretrained ResNetHe et al. 2016) and a 128-dimensional angle embedding vector, f^a is an embedding layer to embed the previous action into a 64-dimensional vector, f^d is another LSTM, and $\hat{\boldsymbol{h}}_t^d \in \mathbb{R}^{512}$ represents the language-grounded state of the navigation. The action is chosen greedily or by sampling (in IL or RL setting respectively) among the N possible movable directions based on their corresponding weight p_j. It worth mentioning that we apply a Dropout of 0.5 between all layers of the network.

We optimise our models using RMSprop with a learning rate of 1×10^{-4} and batch size of 64 for $80,000$ iterations in all of our experiments, except when indicated.

We set $\alpha \approx 0.83$ (i.e. $\frac{\alpha}{(1-\alpha)} = 5$) by grid search in behavioural cloning setting (without counterfactual learning) for all the experiments. Value of α balances the factual and counterfactual and as shown in Fig. 5.3 increasing it (more weights for counterfactuals) improves the performance in the unseen environments to a point. Increasing it further reduces the generalisation since the agent forgets the factual observations.

Reinforcement Learning: When using reinforcement learning, the reward function is measured based on both agent's progress toward the target location and its final success/failure. To that end, at each step we calculate the distance to the target location (d_t) and, based on that, we measure the progress reward ($d_t - d_{t-1}$). Additionally, at the end of each episode (either by reaching the maximum number of steps or after choosing the `stop` action), if d_t is lower than 3 meters, we provide the agent with a big reward of size $+2$. Otherwise, we punish the agent with a negative signal of -2. Note that we set the discount factor to 0.9 in all experiments.

Baselines: To evaluate our approach, we conduct extensive experiments in different learning settings similar to that of (Tan, Yu, and Bansal 2019; Anderson et al. 2018c) for fair comparison: *imitation learning* (IL; $\lambda = 0$), with additional *reinforcement learning* (IL+RL), and with additional *data augmentation* (IL+RL+Aug). We employ behaviour cloning and advantage actor-critic (A2C) algorithm (Mnih et al. 2016b) when IL and RL are needed respectively. The reward is calculated based on the agent's progress toward the target and its final success/failure similar to the baselines (details in the suppl.). In addition, in the augmented setting, similar to (Tan,

	Validation-Seen				**Validation-Unseen**			
Model	**NL↓**	**NE↓**	**SR↑**	**SPL↑**	**NL↓**	**NE↓**	**SR↑**	**SPL↑**
Seq-to-Seq (Anderson et al. 2018c)	11.3	6.01	38.6	-	8.4	7.81	21.8	-
Speaker-Follower (Fried et al. 2018)	-	4.86	52.1	-	-	7.07	31.2	-
Co-Grounding (Ma et al. 2019a)	-	3.65	65.0	0.56	-	6.07	42.0	0.28
IL* (Tan, Yu, and Bansal 2019)	9.9	5.34	50.2	0.48	9.5	6.10	42.6	0.40
IL+Prior	9.9	**5.17**	**50.5**	**0.48**	9.2	5.89	45.5	0.43
IL+Counterfactuals	**9.8**	5.37	48.9	0.47	**9.1**	**5.75**	**46.4**	**0.44**
IL+RL* (Tan, Yu, and Bansal 2019)	10.3	**4.65**	**55.8**	**0.53**	9.7	5.73	44.9	0.41
IL+RL+Prior	11.2	4.78	54.0	0.51	14.9	5.52	48.5	0.44
IL+RL+Counterfactuals	10.7	4.75	53.6	0.51	11.8	**5.42**	**49.4**	**0.46**
IL+RL+Aug* (Tan, Yu, and Bansal 2019)	10.3	4.01	62.5	0.60	**9.7**	5.48	50.3	0.47
IL+RL+Aug+Prior	11.0	3.65	64.4	0.61	13.5	5.13	52.4	0.48
IL+RL+Aug+Counterfactuals	**10.8**	**3.65**	**68.2**	**0.64**	12.4	**4.95**	**53.5**	**0.49**

TABLE 5.1: Evaluation metrics for R2R Navigation. Navigation Length (NL) and Navigation Error (NE) values are represented in meters, while Success Rate (SR) values are percentage. ↑ indicates higher is better, while ↓ shows lower is better. Results indicated by * are as reported in the official implementation: `https://github.com/airsplay/R2R-EnvDrop`.

Yu, and Bansal 2019), we fine-tune our trained model from IL+RL for the maximum of $200,000$ iterations with additional samples obtained from instructions sampled from the speaker.

Evaluation metrics: Similar to (Anderson et al. 2018c; Tan, Yu, and Bansal 2019; Wang et al. 2018; Fried et al. 2018), we employ both the Navigation Error (NE), the difference as measured in meters between the agent's final position and the target location, and the Success Rate (SR), the the portion of traversed trajectories at which the NE is less than 3 meters, to evaluate the performance of a navigating agent. However, Success weighted by Path Length (SPL) (Anderson et al. 2018a) better represents the efficiency by taking into account the inverse ratio of the agent's Trajectory Length (TL)–the distance the agent travelled– to the ground-truth. We demonstrate all of these metrics for both seen and unseen environments.

Results: We report the evaluation performance of the proposed approach in Table 5.1. We indicate the use of counterfactual objective in Eq. (5.8) by **+Counterfactuals**. We consider reporting the performance of our approach by simply conditioning the interventions on the prior distribution of the exogenous variable indicated by **+Pior**. Using the prior, compared to the one optimised in Eq. (5.11),

	Validation-Unseen				Test-Unseen			
Model	**NL↓**	**NE↓**	**SR↑**	**SPL↑**	**NL↓**	**NE↓**	**SR↑**	**SPL↑**
Random (Anderson et al. 2018c)	9.8	9.23	16.3	-	9.9	9.79	13.2	0.12
Seq-to-Seq (Anderson et al. 2018c)	8.4	7.81	-	0.22	8.1	7.85	20.4	0.18
Speaker-Follower (Fried et al. 2018)	-	6.62	35.5	-	14.8	-	35.0	0.28
Self-Monitoring (Ma et al. 2019a)	-	5.41	47.0	0.34	18.0	5.67	48.0	0.35
Reinforced Cross-Modal (Wang et al. 2019a)	11.5	6.09	50.1	0.43	12.0	6.12	43.0	0.38
Tactical-Rewind (Ke et al. 2019)	21.2	4.97	**56.0**	0.43	22.1	5.14	54.0	0.41
Counterfactual VLN (Fu et al. 2019)	-	5.40	47.7	0.43	-	5.80	45.1	0.41
Environment Dropout (Tan, Yu, and Bansal 2019)	10.7	5.22	52.2	0.48	11.7	5.23	51.5	0.47
Ours	12.4	**4.95**	53.5	**0.49**	13.0	**4.90**	**54.9**	**0.50**
PRESS* (Li et al. 2019)	10.4	5.28	49.0	0.45	10.8	5.49	49.0	0.45
PREVALENT* (Hao et al. 2020)	10.2	4.71	58.0	0.53	10.5	5.30	54.0	0.51

TABLE 5.2: The comparison of our results with others in unseen environments. Test-unseen results are reported on the task's leaderboard in single-run setting. The methods indicated by * are taking advantage of self-supervised pre-training. Our method can be applied on top of these methods to result in a further improvement.

evaluates the value of estimating the posterior. As shown, by incorporating the counterfactuals the navigation performance of the imitating agent, in particular for the unseen environments, improves significantly. We particularly observe around 4% improvement in SR and SPL compared to the baseline.

More importantly, our method improves the generalisation by decreasing the SR gap between the seen and unseen environments from around 8 to 2.5%–a significant improvement indeed.

Once the reinforcement signal is added (i.e. $\lambda = 5$), our proposed policy's performance improves further by more than 3% for SR compared to its IL counterpart. Furthermore, our method enjoys about 5% improvement in SR and SPL in unseen environments, and, more importantly, an approximately 6.7% drop in the seen versus unseen performance gap. Further, using augmentations, our model enjoys another 4% boost in both SR and SPL.

Finally, we submitted our proposed model to the leaderboard for the evaluation on the test set–a hold-out dataset of 18 environments for a fair challenge[‡]. Table 5.2 demonstrates the superior performance of our model in comparison to other baselines.

[‡] Our evaluation on the test set is available at: `https://evalai.cloudcv.org/web/challenges/challenge-page/97/leaderboard/270`

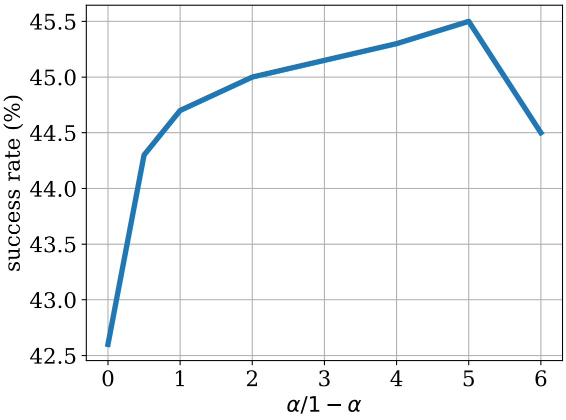


FIGURE 5.3: The effect of α on the results inside unseen environments. $\frac{\alpha}{(1-\alpha)} = 0$ means no counterfactual is used (conventional training).

Interestingly, our model outperforms the EnvDrop model (Tan, Yu, and Bansal 2019), the most similar model to ours, by a significant margin of 3.4 percent in SR and 3 points in SPL. Besides, our agent surpasses self-supervised pre-training of (Hao et al. 2020), in terms of success rate and navigation error–a model that we believe can further benefit from our approach.

5.6.2 Embodied Question Answering

Dataset: Embodied Question Answering (EQA) (Das et al. 2018a) is a challenging variant of Vision and Language Navigation where in contrast to R2R task, the agent is given a general question about an object in the environment, e.g. “what colour is the car?”. Spawning in a random location in an unseen environment at test time, the agent must first navigate to the proximity of the desired object and subsequently answer the given question. The dataset consists of $6,912$ tuples of route-question-answer in 645 distinct training environments and a collection of 898 tuples in 57 unseen environments for the test set. At each step, the agent is provided with an egocentric RGB image based on which the agent should choose the next action among a set of 4 discrete choices (`forward`, `turn-left`, `turn-right` and `stop`). We treat the question as the instructions of the R2R dataset.

Implementation details: To attend the visual features of the egocentric RGB image in House3D environments, we utilise the pre-trained CNN proposed in Das et al. 2018a. The network consist of 4 convolutional blocks in which a 5×5 convolution

Model	$d_T\downarrow$ T_{-10}	$d_T\downarrow$ T_{-30}	$d_T\downarrow$ T_{-50}	$d_\Delta\uparrow$ T_{-10}	$d_\Delta\uparrow$ T_{-30}	$d_\Delta\uparrow$ T_{-50}
PACMAN (Das et al. 2018a)	1.39	4.98	9.33	-0.45	0.49	1.66
Neural Modular Control (Das et al. 2018b)	0.85	4.32	9.29	0.09	1.15	1.70
GRU	0.74	3.99	8.73	0.20	1.48	2.26
GRU+Prior	0.73	3.95	8.50	0.21	1.52	2.49
GRU+Counterfactuals	**0.71**	**3.88**	**8.46**	**0.23**	**1.59**	**2.53**

TABLE 5.3: Evaluation metrics for EQA navigation.

layer is followed by BatchNorm, ReLU and 2 × 2 MaxPool layers. The network is trained in a multi-task learning setting where the outputs of the last convolutional block are fed into three separate decoder heads for RGB image reconstruction, pixel-wise semantic segmentation and semantic classification. In our experiments, we extract the outputs of the last convolutional block ($\mathbb{R}^{3200}$) and downsize its dimension to 128 using a fully-connected layer to reach latent observation representations $\mathbf{z}$.

We train all of the models for 30 epochs (more than $10,000$ iterations) in a behavioural cloning setting with a batch size of 20 and learning rate set to 1×10^{-3} using Adam optimiser. It should be noted that since there is no instructions to be followed (just the question here) we disregard the second term in Eq. (5.11) for this task.

Evaluation metrics: For the evaluation, we spawn the agent in 10, 30, or 50 steps away from the target location in terms of the shortest path (similar to (Das et al. 2018a)). The main metric for the evaluation is the distance (in meters) between the location where the agent stops and the ground-truth target denoted by d_T. Additionally, we consider $d_\Delta = d_T - d_0$ as another critical metric measuring the overall progress of the agent from its initial position d_0 towards the target. In contrast to d_T, higher values of d_Δ show better performance. The agent is constrained to a maximum of 100 steps at each episode.

Results: As shown in Table 5.3, almost 10% increase in generalisation to unseen environments is achieved by letting the agent contemplate the unseen. Finally, not only our approach improves the performance of the agent in reaching short-term goals (T_{-10}), but it also enhances its accuracy in finding distant objects (T_{-50}).

EQA is more complex than R2R (long trajectories and high-level language instructions) for which the scores are generally low and the agent learns trivial actions, e.g. going through the door. We found correspondingly using grid search, the best performance is when $\alpha \approx 0.29$ (i.e. $\frac{\alpha}{(1-\alpha)} = 0.4$)–a considerably smaller value to that of R2R. This supports our hypothesis for using longer trajectories in Eq. (5.8) in which, when the gain is low, the agent must primarily focus on maximising gain (even if that leads to trivial actions) rather than variations. Nevertheless, using counterfactuals even for such a difficult task improves performance of our agent to achieve state-of-the-art results.

5.7 Conclusions

Generalisation ability is paramount for developing a practical VLN in robots that can operate in the wild, yet many overfit the instructions to the visual stimuli in the training. More importantly, current approaches fail to incorporate any mechanism for reasoning about the likelihood of alternative trajectories – a crucial skill for the task. To remedy the issue, we turned to the counterfactuals as a principled approach for reasoning about unobserved scenarios for estimating the effect of an intervention that is not directly represented in the data. We formulated the new learning objective to incorporate both the real data as well as the counterfactuals obtained conditioned on the exogenous variable. This implicitly forces the navigation policy and the internal state representation to learn semantics and high-level relations rather than relying on statistical regularities specific to either visual observations or instructions. The effectiveness of our approach has been illustrated in two challenging VLN tasks. Crucially, our method is a general model that can be implemented not only in any VLN task but also in complex multi-modal problems where high-level reasoning is required and generalisation is paramount; thus, we consider exploring this avenue further in future.

Instruction: *Walk forward up the set of three stairs. Enter the room at the end of the hallway. Walk o the massage table, and stop.*

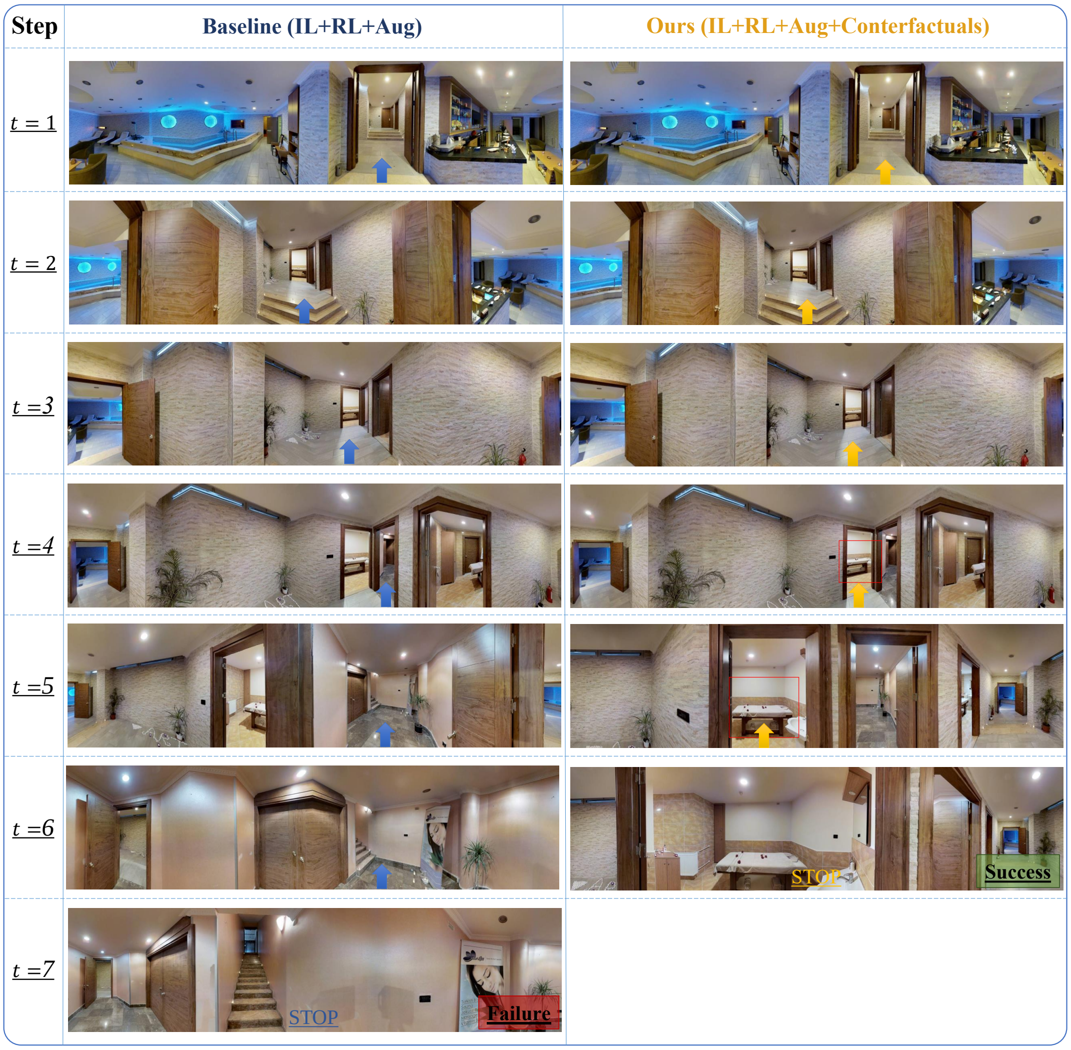


FIGURE 5.4: Trajectory Sample. The baseline agent (left trajectory) follows the language instruction until step 4, where instead of moving towards the *massage table*, it goes into the next *hallway*. We argue that since there are limited *massage tables* in the training set, the baseline method does not consider this one as a variant of table and continues searching to stop at the end of the next hallway (6 meters away from the target position). On the other hand, our agent (at the right side), succeeds in identifying the *massage table* and ends up at the target location without any navigation error.

Instruction: *Walk through double doors into the house. Continue around the dining table and through the entry way to the next room. Walk up to the couch and armchairs surrounding a coffee table.*

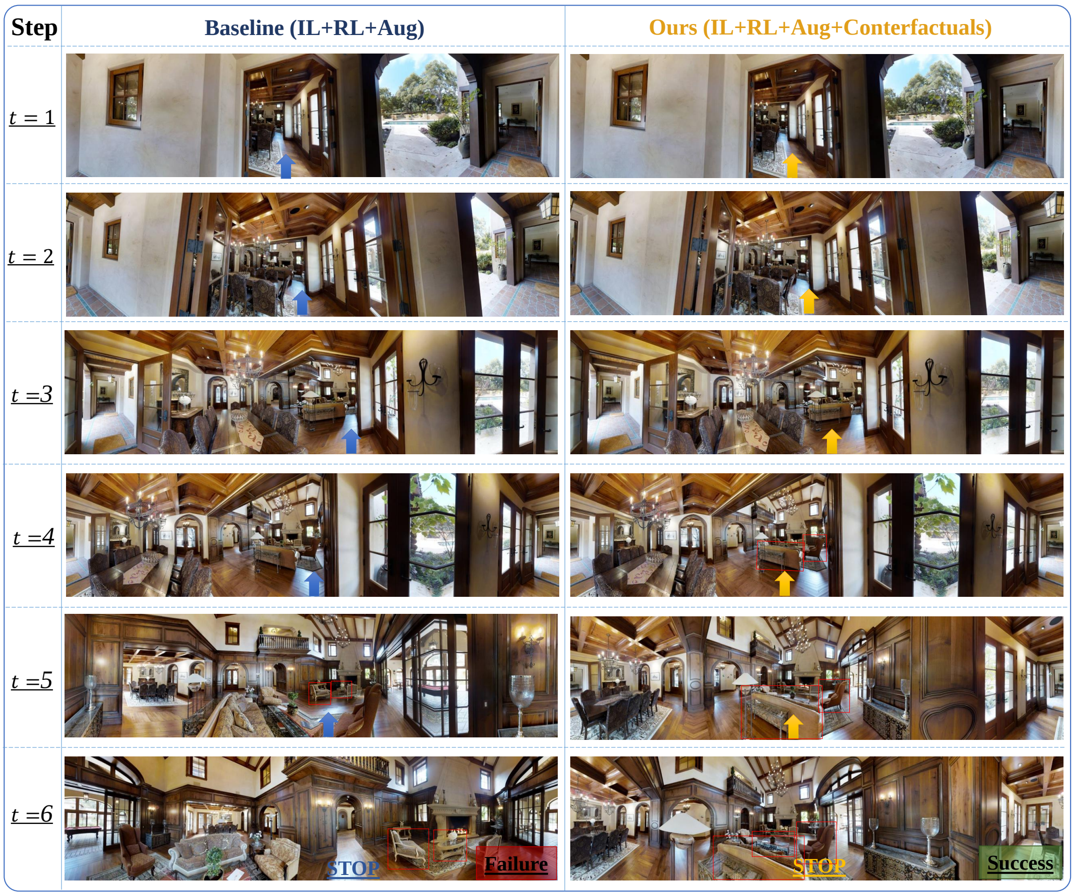


FIGURE 5.5: Trajectory Sample. Both agents follow the same path until step 3, where they need to identify and reach the *coffee table* that is surrounded by the *couch* and *armchairs*. In contrast to the baseline model that looks for typical tables in the environment and overlooks the couch from the back view, our model, recognises them and attends the target position successfully.

Instruction: *'Exit the room then go straight and turn left. Go straight until you pass an eye chart picture frame on the left wall then wait there.*

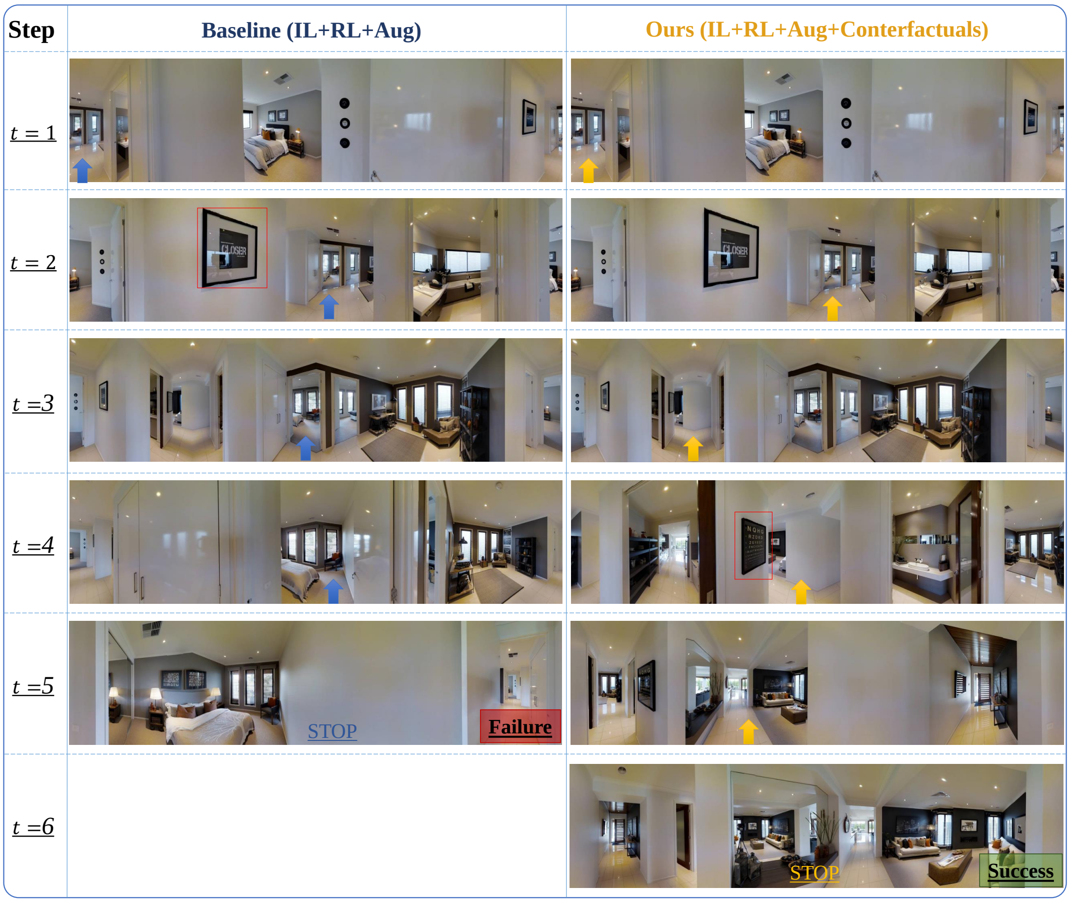


FIGURE 5.6: Trajectory Sample. The baseline model neglects a part of the instruction, and it seems that it has presumed the *picture frame* in step 2 as the one mentioned in the guidance improperly. Biased by the great number of trajectories in the training set, it decides to go into the door at the opposite side, which costs the agent to end up unsuccessfully (7 meters away from the target). On the contrary, our approach executes the instruction precisely, finds the *eye chart picture frame* correctly, and stops at the vicinity of the goal location (1 meter error).

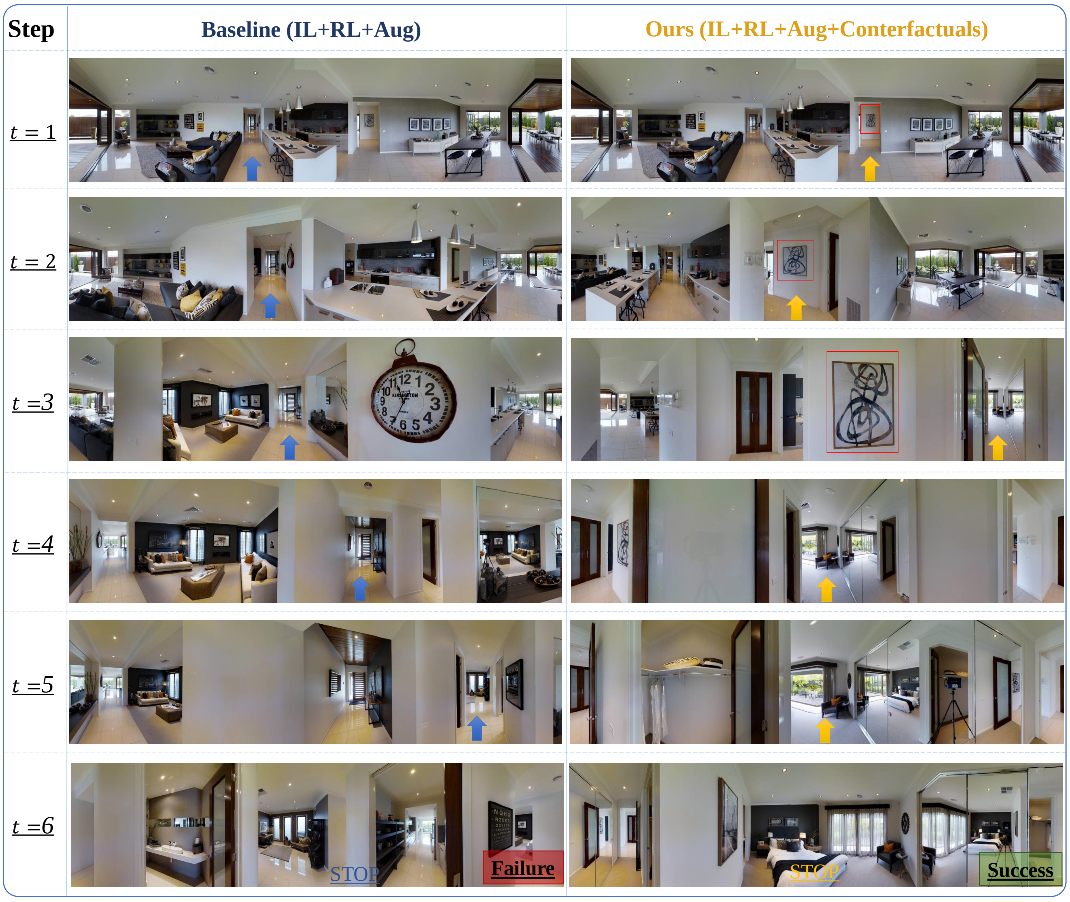

Instruction: *Walk straight past the bar through the doorway. Turn right at the picture and enter the bedroom. Stop and wait by the closet.*

FIGURE 5.7: Trajectory Sample. From the pictures it is evident that the baseline approach cannot find the right path, that is identifiable with the *picture* clue in the instruction, and, consequently, ends up about 13 meters away from the target. On the other hand, our approach succeeds in correlating the language instruction to the correct path and reaching the target location (less than 1 meter error).

Chapter 6

Conclusion

Training an intelligent agent able to interact with humans in interactive environments is suffering from some limitations, from which we addressed three major ones in this thesis. Firstly, we proposed a consistent and fluent visual negotiation model that is able to compete with humans over selling/buying an item. For that, we empowered the agent to go through the online shops in search of similar items with the purpose of understanding the real value of the item, which plays an essential role in human behaviour understanding. Moreover, in a hierarchical recurrent model, we enabled the agent to comprehend all the influential information in a negotiation from various modalities to later build competent and reasonable pricing and language generation policies. One potential improvement that we consider in our further research is the integration of transformer-based pre-trained models that have shown astonishing results recently (Devlin et al. 2018; Radford et al. 2019; Hong et al. 2021).

Secondly, to supply an interactive agent with the ability to efficiently seek more information from humans, we focused on active learning (AL), which aims at decreasing the labelling cost during a human-in-the-loop process. For that, we propose a novel AL method based on interpolations in the feature space that finds the most informative subset of unlabelled instances carrying distinctive features, which, after labelling and using for training, the model's performance would increase significantly. In our AL approach, we offer an efficient and scalable mixing algorithm between the features of an unlabelled sample and salient features of each class (extracted from the labelled set) to find unlabelled instances with novel features in their neighbourhood.

Extensive experiments on a wide range of image, video and non-visual data classification tasks revealed the superiority of our approach in comparison with state-of-the-art AL methods. We set the investigation of the applicability of our AL approach on interactive vision and language problems as a future direction of study.

Finally, to prevent the model's tendency towards learning biased correlations in the data and memorising features of seen environments, which are major limitations for the generalisation of interactive vision and language models, we proposed counterfactuals as a form of data augmentation. As for the biased correlation problem, we suggest to extend the training set to include counterfactual instances generated from the minimum interventions in the features of actual samples that affect their related output. We showed the effectiveness of our approach for the generalisation enhancement of various unimodal and multimodal applications in the area of vision and language. Regarding the later issue, we further expanded the utilisation of counterfactuals in interactive environments in a way that the agent can create counterfactual environments on the fly and learn from the interactions in those environments. This way, in the context of two challenging vision and language navigation tasks, we significantly enhanced the model's generalisation to unseen environments. One further extension of this work would be to apply it on interactive tasks where the agent is constantly in connection with humans and can ask for further guides during the execution of the policy (*i.e.* Vision and Dialogue Navigation).

Bibliography

Abbasnejad, Ehsan, Anthony R. Dick, and Anton van den Hengel (2017). "Infinite Variational Autoencoder for Semi-Supervised Learning". In: *CVPR*. IEEE Computer Society, pp. 781–790. ISBN: 978-1-5386-0457-1.

Abbasnejad, Ehsan, Justin Domke, and Scott Sanner (2015). "Loss-calibrated monte carlo action selection". In: *Twenty-Ninth AAAI Conference on Artificial Intelligence*.

Abbasnejad, Ehsan, Javen Shi, and Anton van den Hengel (2018). "Deep lipschitz networks and dudley GANs". In.

Abbasnejad, Ehsan, Damien Teney, Amin Parvaneh, Javen Shi, and Anton van den Hengel (2020). "Counterfactual Vision and Language Learning". In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Abbasnejad, Ehsan, Qi Wu, Qinfeng Shi, and Anton van den Hengel (2019). "What's to know? Uncertainty as a Guide to Asking Goal-oriented Questions". In: *Proceedings of International Conference in Computer Vision and Pattern Recognition*, pp. 4155–4164.

Agarwal, Sharat, Himanshu Arora, Saket Anand, and Chetan Arora (2020). "Contextual Diversity for Active Learning". In: *European Conference on Computer Vision*. Springer, pp. 137–153.

Agrawal, Aishwarya, Dhruv Batra, Devi Parikh, and Aniruddha Kembhavi (2018). "Don't Just Assume; Look and Answer: Overcoming Priors for Visual Question Answering". In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Agrawal, Aishwarya, Jiasen Lu, Stanislaw Antol, Margaret Mitchell, C. Lawrence Zitnick, Devi Parikh, and Dhruv Batra (2017). “VQA: visual question answering”. In: *Int. J. Comput. Vis.* 123.1, pp. 4–31. ISSN: 0920-5691. DOI: `10.1007/s11263-016-0966-6`. URL: `https://doi.org/10.1007/s11263-016-0966-6`.

Ammicht, E., E. Fosler-Lussier, and A. Potamianos (2007). “Information Seeking Spoken Dialogue Systems— Part I: Semantics and Pragmatics”. In: *IEEE Transactions on Multimedia* 9.3, pp. 532–549.

Anderson, Peter, Angel Chang, Devendra Singh Chaplot, Alexey Dosovitskiy, Saurabh Gupta, Vladlen Koltun, Jana Kosecka, Jitendra Malik, Roozbeh Mottaghi, Manolis Savva, and Amir R. Zamir (2018a). “On Evaluation of Embodied Navigation Agents”. In: *arXiv:1807.06757*.

Anderson, Peter, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang (2018b). “Bottom-Up and Top-Down Attention for Image Captioning and Visual Question Answering”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Anderson, Peter, Qi Wu, Damien Teney, Jake Bruce, Mark Johnson, Niko Sünderhauf, Ian Reid, Stephen Gould, and Anton van den Hengel (2018c). “Vision-and-Language Navigation: Interpreting Visually-Grounded Navigation Instructions in Real Environments”. In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Andrychowicz, Marcin, Filip Wolski, Alex Ray, Jonas Schneider, Rachel Fong, Peter Welinder, Bob McGrew, Josh Tobin, OpenAI Pieter Abbeel, and Wojciech Zaremba (2017). “Hindsight Experience Replay”. In: *Advances in Neural Information Processing Systems*. Ed. by I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett. Vol. 30. Curran Associates, Inc. URL: `https://proceedings.neurips.cc/paper/2017/file/453fadbd8a1a3af50a9df4df899537b5-Paper.pdf`.

Arora, Saurabh and Prashant Doshi (2021). “A survey of inverse reinforcement learning: Challenges, methods and progress”. In: *Artificial Intelligence* 297, p. 103500. ISSN: 0004-3702. DOI: `https://doi.org/10.1016/j.artint.2021.103500`. URL: `https://www.sciencedirect.com/science/article/pii/S0004370221000515`.

Ash, Jordan T., Chicheng Zhang, Akshay Krishnamurthy, John Langford, and Alekh Agarwal (2020). “Deep Batch Active Learning by Diverse, Uncertain Gradient Lower Bounds”. In: *International Conference on Learning Representations*. URL: `https://openreview.net/forum?id=ryghZJBKPS`.

Auer, Peter, Nicolo Cesa-Bianchi, Yoav Freund, and Robert E Schapire (2002). “The nonstochastic multiarmed bandit problem”. In: *SIAM journal on computing* 32.1, pp. 48–77.

Beluch, William H., Tim Genewein, Andreas Nurnberger, and Jan M. Kohler (2018). “The Power of Ensembles for Active Learning in Image Classification”. In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9368–9377. DOI: `10.1109/CVPR.2018.00976`.

Bordes, Antoine, Y-Lan Boureau, and Jason Weston (2017). “Learning end-to-end goal-oriented dialog”. In: *International Conference on Learning Representations*. URL: `https://openreview.net/forum?id=S1Bb3D5gg`.

Boyd, Stephen and Lieven Vandenberghe (2004). *Convex Optimization*. Cambridge University Press. ISBN: 0521833787. URL: `http://www.amazon.com/exec/obidos/redirect?tag=citeulike-20\&path=ASIN/0521833787`.

Buesing, Lars, Theophane Weber, Yori Zwols, Nicolas Heess, Sebastien Racaniere, Arthur Guez, and Jean-Baptiste Lespiau (2019). “Woulda, Coulda, Shoulda: Counterfactually-Guided Policy Search”. In: *International Conference on Learning Representations*. URL: `https://openreview.net/forum?id=BJG0voC9YQ`.

Cadene, Remi, Hedi Ben-Younes, Nicolas Thome, and Matthieu Cord (2019). "MUREL: Multimodal Relational Reasoning for Visual Question Answering". In: *IEEE Conference on Computer Vision and Pattern Recognition CVPR*. URL: http://remicadene.com/pdfs/paper_cvpr2019.pdf.

Caramalau, Razvan, Binod Bhattarai, and Tae-Kyun Kim (2021). "Sequential Graph Convolutional Network for Active Learning". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9583–9592.

Caron, Mathilde, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin (2021). "Emerging Properties in Self-Supervised Vision Transformers". In: *Proceedings of the International Conference on Computer Vision (ICCV)*.

Carreira, Joao and Andrew Zisserman (2017). "Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Chang, Angel, Angela Dai, Thomas Funkhouser, Maciej Halber, Matthias Niessner, Manolis Savva, Shuran Song, Andy Zeng, and Yinda Zhang (2017). "Matterport3D: Learning from RGB-D Data in Indoor Environments". In: *International Conference on 3D Vision (3DV)*.

Charles, Denis, Max Chickering, and Patrice Simard (2013). "Counterfactual Reasoning and Learning Systems: The Example of Computational Advertising". In: *Journal of Machine Learning Research* 14, pp. 3207–3260.

Chaudhari, Sneha, Varun Mithal, Gungor Polatkan, and Rohan Ramanath (2021). "An Attentive Survey of Attention Models". In: *ACM Trans. Intell. Syst. Technol.* 12.5. ISSN: 2157-6904. DOI: 10.1145/3465055. URL: https://doi.org/10.1145/3465055.

Chen, Hongshen, Xiaorui Liu, Dawei Yin, and Jiliang Tang (2017). "A Survey on Dialogue Systems: Recent Advances and New Frontiers". In: *ACM SIGKDD Explorations Newsletter* 19.2, pp. 25–35. DOI: 10.1145/3166054.3166058.

Chen, Howard, Alane Suhr, Dipendra Misra, Noah Snavely, and Yoav Artzi (2019). "TOUCHDOWN: Natural Language Navigation and Spatial Reasoning in Visual Street Environments". In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Choi, Jongwon, Kwang Moo Yi, Jihoon Kim, Jinho Choo, Byoungjip Kim, Jinyeop Chang, Youngjune Gwon, and Hyung Jin Chang (2021). "VaB-AL: Incorporating Class Imbalance and Difficulty With Variational Bayes for Active Learning". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6749–6758.

Cohen, Gregory, Saeed Afshar, Jonathan Tapson, and André van Schaik (2017). "EMNIST: Extending MNIST to handwritten letters". In: *2017 International Joint Conference on Neural Networks (IJCNN)*, pp. 2921–2926. DOI: `10.1109/IJCNN.2017.7966217`.

Cuayáhuitl, Heriberto, Simon Keizer, and Oliver Lemon (2015). "Strategic Dialogue Management via Deep Reinforcement Learning". In: *NIPS'15 Workshop on Deep Reinforcement Learning*.

Damien Teney Ehsan Abbasnejad, Anton van den Hengel (2020). "Unshuffling Data for Improved Generalization". In: *arXiv preprint arXiv:2002.11894*.

Das, Abhishek, Samyak Datta, Georgia Gkioxari, Stefan Lee, Devi Parikh, and Dhruv Batra (2018a). "Embodied Question Answering". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Das, Abhishek, Georgia Gkioxari, Stefan Lee, Devi Parikh, and Dhruv Batra (2018b). "Neural Modular Control for Embodied Question Answering". In: *Proceedings of the Conference on Robot Learning (CoRL)*.

Das, Abhishek, Satwik Kottur, Khushi Gupta, Avi Singh, Deshraj Yadav, Jose M. F. Moura, Devi Parikh, and Dhruv Batra (2017a). "Visual Dialog". In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Das, Abhishek, Satwik Kottur, José M. F. Moura, Stefan Lee, and Dhruv Batra (2017b). "Learning Cooperative Visual Dialog Agents with Deep Reinforcement Learning".

In: *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 2970–2979.

Deng, J., W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei (2009). “ImageNet: A Large-Scale Hierarchical Image Database”. In: *CVPR09*.

Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova (2018). “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding”. In: *arXiv:1810.04805v1*.

Dhingra, Bhuwan, Lihong Li, Xiujun Li, Jianfeng Gao, Yun-Nung Chen, Faisal Ahmed, and Li Deng (2017). “Towards End-to-End Reinforcement Learning of Dialogue Agents for Information Access”. In: *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Vancouver, Canada, pp. 484–495. DOI: `10.18653/v1/P17-1045`. URL: `http://aclweb.org/anthology/P17-1045`.

Dosovitskiy, Alexey, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby (2021). “An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale”. In: *International Conference on Learning Representations*. URL: `https://openreview.net/forum?id=YicbFdNTTy`.

Ducoffe, Melanie and Frederic Precioso (2018). “Adversarial Active Learning for Deep Networks: a Margin Based Approach”. In: *arXiv:1802.09841*.

Dušek, Ondřej and Filip Jurcicek (2016). “A Context-aware Natural Language Generator for Dialogue Systems”. In: *Proceedings of the 17th Annual Meeting of the Special Interest Group on Discourse and Dialogue*. Los Angeles, pp. 185–190. DOI: `10.18653/v1/W16-3622`. URL: `http://aclweb.org/anthology/W16-3622`.

El Asri, Layla, Hannes Schulz, Shikhar Sharma, Jeremie Zumer, Justin Harris, Emery Fine, Rahul Mehrotra, and Kaheer Suleman (2017). “Frames: a corpus for adding memory to goal-oriented dialogue systems”. In: *Proceedings of the 18th Annual*

SIGdial Meeting on Discourse and Dialogue. Saarbrücken, Germany, pp. 207–219. DOI: 10.18653/v1/W17-5526. URL: http://aclweb.org/anthology/W17-5526.

Engelcke, Martin, Adam R. Kosiorek, Oiwi Parker Jones, and Ingmar Posner (2020). "GENESIS: Generative Scene Inference and Sampling with Object-Centric Latent Representations". In: *International Conference on Learning Representations*. URL: https://openreview.net/forum?id=BkxfaTVFwH.

Fan, Haoqi, Bo Xiong, Karttikeya Mangalam, Yanghao Li, Zhicheng Yan, Jitendra Malik, and Christoph Feichtenhofer (2021). "Multiscale Vision Transformers". In: *CoRR* abs/2104.11227. arXiv: 2104.11227. URL: https://arxiv.org/abs/2104.11227.

Fang, Bin, Shidong Jia, Di Guo, Muhua Xu, Shuhuan Wen, and Fuchun Sun (2019). "Survey of imitation learning for robotic manipulation". In: *International Journal of Intelligent Robotics and Applications* 3, pp. 362–369. ISSN: 2366-598X. DOI: 10.1007/s41315-019-00103-5.

Fried, Daniel, Ronghang Hu, Volkan Cirik, Anna Rohrbach, Jacob Andreas, Louis-Philippe Morency, Taylor Berg-Kirkpatrick, Kate Saenko, Dan Klein, and Trevor Darrell (2018). "Speaker-Follower Models for Vision-and-Language Navigation". In: *Advances in Neural Information Processing Systems 31*. Ed. by S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett. Curran Associates, Inc., pp. 3314–3325. URL: http://papers.nips.cc/paper/7592-speaker-follower-models-for-vision-and-language-navigation.pdf.

Fu, Tsu-Jui, Xin Wang, Matthew Peterson, Scott Grafton, Miguel Eckstein, and William Yang Wang (2019). "Counterfactual Vision-and-Language Navigation via Adversarial Path Sampling". In: *arXiv preprint arXiv:1911.07308*.

Gal, Yarin, Riashat Islam, and Zoubin Ghahramani (2017). "Deep Bayesian Active Learning with Image Data". In: *Proceedings of the 34th International Conference* *on Machine Learning*. Ed. by Doina Precup and Yee Whye Teh. Vol. 70. Proceedings of Machine Learning Research. PMLR, pp. 1183–1192. URL: http://proceedings.mlr.press/v70/gal17a.html.

Geirhos, Robert, Patricia Rubisch, Claudio Michaelis, Matthias Bethge, Felix A. Wichmann, and Wieland Brendel (2019). “ImageNet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness.” In: *International Conference on Learning Representations*. URL: https://openreview.net/forum?id=Bygh9j09KX.

Goodfellow, Ian, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio (2014). “Generative Adversarial Nets”. In: *Advances in Neural Information Processing Systems 27*. Ed. by Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger. Curran Associates, Inc., pp. 2672–2680. URL: http://papers.nips.cc/paper/5423-generative-adversarial-nets.pdf.

Goodfellow, Ian J., Jonathon Shlens, and Christian Szegedy (2015). “Explaining and Harnessing Adversarial Examples”. In: *ICLR 2015 : International Conference on Learning Representations 2015*. URL: https://academic.microsoft.com/paper/2963207607.

Goyal, Yash, Tejas Khot, Aishwarya Agrawal, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh (2019a). “Making the V in VQA Matter: Elevating the Role of Image Understanding in Visual Question Answering”. In: *Int. J. Comput. Vision* 127.4, pp. 398–414. ISSN: 0920-5691. DOI: 10.1007/s11263-018-1116-0. URL: https://doi.org/10.1007/s11263-018-1116-0.

Goyal, Yash, Ziyan Wu, Jan Ernst, Dhruv Batra, Devi Parikh, and Stefan Lee (2019b). “Counterfactual Visual Explanations”. In: *Proceedings of the 36th International Conference on Machine Learning*. Ed. by Kamalika Chaudhuri and Ruslan Salakhutdinov. Vol. 97. Proceedings of Machine Learning Research. Long Beach, California, USA: PMLR, pp. 2376–2384.

Gu, Jiatao, Zhengdong Lu, Hang Li, and Victor O.K. Li (2016). “Incorporating Copying Mechanism in Sequence-to-Sequence Learning”. In: *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Berlin, Germany: Association for Computational Linguistics, pp. 1631–1640. DOI: 10.18653/v1/P16-1154. URL: https://www.aclweb.org/anthology/P16-1154.

Guo, Yulan, Hanyun Wang, Qingyong Hu, Hao Liu, Li Liu, and Mohammed Bennamoun (2021). “Deep Learning for 3D Point Clouds: A Survey”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 43.12, pp. 4338–4364. DOI: 10.1109/TPAMI.2020.3005434.

Ha, David and Jürgen Schmidhuber (2018). “Recurrent World Models Facilitate Policy Evolution”. In: *Advances in Neural Information Processing Systems*. Ed. by S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett. Vol. 31. Curran Associates, Inc. URL: https://proceedings.neurips.cc/paper/2018/file/2de5d16682c3c35007e4e92982f1a2ba-Paper.pdf.

Hao, Weituo, Chunyuan Li, Xiujun Li, Lawrence Carin, and Jianfeng Gao (2020). “Towards Learning a Generic Agent for Vision-and-Language Navigation via Pre-training”. In: *Conference on Computer Vision and Pattern Recognition (CVPR)*.

He, He, Anusha Balakrishnan, Mihail Eric, and Percy Liang (2017). “Learning Symmetric Collaborative Dialogue Agents with Dynamic Knowledge Graph Embeddings”. In: *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Vancouver, Canada, pp. 1766–1776. DOI: 10.18653/v1/P17-1162. URL: http://aclweb.org/anthology/P17-1162.

He, He, Derek Chen, Anusha Balakrishnan, and Percy Liang (2018). “Decoupling Strategy and Generation in Negotiation Dialogues”. In: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Brussels, Belgium, pp. 2333–2343. URL: http://aclweb.org/anthology/D18-1256.

He, K., X. Zhang, S. Ren, and J. Sun (2016). “Deep Residual Learning for Image Recognition”. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778. DOI: 10.1109/CVPR.2016.90.

He, Kaiming, Xiangyu Zhang, Shaoqing Ren, and Jian Sun (2016). “Identity mappings in deep residual networks”. In: *European Conference on Computer Vision*.

Hoffman, Judy, Daniel A Roberts, and Sho Yaida (2019). “Robust learning with jacobian regularization”. In: *arXiv preprint arXiv:1908.02729*.

Hong, Yicong, Qi Wu, Yuankai Qi, Cristian Rodriguez-Opazo, and Stephen Gould (2021). “VLN BERT: A Recurrent Vision-and-Language BERT for Navigation”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1643–1653.

Hsu, Wei-Ning and Hsuan-Tien Lin (2015). “Active Learning by Learning”. In: *AAAI Conference on Artificial Intelligence*. URL: https://www.aaai.org/ocs/index.php/AAAI/AAAI15/paper/view/9636.

Hu, Ronghang, Huazhe Xu, Marcus Rohrbach, Jiashi Feng, Kate Saenko, and Trevor Darrell (2016). “Natural Language Object Retrieval”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Huang, G., Z. Liu, L. Van Der Maaten, and K. Q. Weinberger (2017). “Densely Connected Convolutional Networks”. In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2261–2269. DOI: 10.1109/CVPR.2017.243.

Huang, Lun, Wenmin Wang, Jie Chen, and Xiao-Yong Wei (2019). “Attention on Attention for Image Captioning”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*.

Huang, Sheng-jun, Rong Jin, and Zhi-Hua Zhou (2010). “Active Learning by Querying Informative and Representative Examples”. In: *Advances in Neural Information*

Processing Systems. Ed. by J. Lafferty, C. Williams, J. Shawe-Taylor, R. Zemel, and A. Culotta. Vol. 23. Curran Associates, Inc., pp. 892–900.

Hudson, Drew A and Christopher D Manning (2019). “GQA: A New Dataset for Real-World Visual Reasoning and Compositional Question Answering”. In: *Conference on Computer Vision and Pattern Recognition (CVPR)*.

Jo, Jason and Yoshua Bengio (2017). “Measuring the tendency of CNNs to learn surface statistical regularities”. In: *arXiv preprint arXiv:1711.11561*.

Johansson, Fredrik, Uri Shalit, and David Sontag (2016). “Learning representations for counterfactual inference”. In: *International conference on machine learning*, pp. 3020–3029.

Johnson, Justin, Bharath Hariharan, Laurens van der Maaten, Li Fei-Fei, C Lawrence Zitnick, and Ross Girshick (2017). “CLEVR: A Diagnostic Dataset for Compositional Language and Elementary Visual Reasoning”. In: *CVPR*.

Kaiser, Łukasz, Mohammad Babaeizadeh, Piotr Miłos, Błażej Osiński, Roy H Campbell, Konrad Czechowski, Dumitru Erhan, Chelsea Finn, Piotr Kozakowski, Sergey Levine, Afroz Mohiuddin, Ryan Sepassi, George Tucker, and Henryk Michalewski (2020). “Model Based Reinforcement Learning for Atari”. In: *International Conference on Learning Representations*. URL: `https://openreview.net/forum?id=S1xCPJHtDB`.

Kaushik, Divyansh, Eduard Hovy, and Zachary Lipton (2020). “Learning The Difference That Makes A Difference With Counterfactually-Augmented Data”. In: *International Conference on Learning Representations*. URL: `https://openreview.net/forum?id=Sklgs0NFvr`.

Kazemzadeh, Sahar, Vicente Ordonez, Mark Matten, and Tamara Berg (2014). “ReferItGame: Referring to Objects in Photographs of Natural Scenes”. In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Doha, Qatar: Association for Computational Linguistics, pp. 787–798. DOI: `10.3115/v1/D14-1086`. URL: `https://aclanthology.org/D14-1086`.

Ke, Liyiming, Xiujun Li, Yonatan Bisk, Ari Holtzman, Zhe Gan, Jingjing Liu, Jianfeng Gao, Yejin Choi, and Siddhartha Srinivasa (2019). "Tactical Rewind: Self-Correction via Backtracking in Vision-And-Language Navigation". In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Kim, Jin-Hwa, Jaehyun Jun, and Byoung-Tak Zhang (2018). "Bilinear Attention Networks". In: *Advances in Neural Information Processing Systems 31*, pp. 1571–1581.

Kim, Kwanyoung, Dongwon Park, Kwang In Kim, and Se Young Chun (2021). "Task-Aware Variational Adversarial Active Learning". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8166–8175.

Kingma, Diederik P and Max Welling (2013). *Auto-Encoding Variational Bayes*.

Kiran, B Ravi, Ibrahim Sobh, Victor Talpaert, Patrick Mannion, Ahmad A. Al Sallab, Senthil Yogamani, and Patrick Pérez (2021). "Deep Reinforcement Learning for Autonomous Driving: A Survey". In: *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–18. DOI: `10.1109/TITS.2021.3054625`.

Kiros, Ryan, Yukun Zhu, Ruslan R Salakhutdinov, Richard Zemel, Raquel Urtasun, Antonio Torralba, and Sanja Fidler (2015). "Skip-Thought Vectors". In: *Advances in Neural Information Processing Systems 28*. Ed. by C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett. Curran Associates, Inc., pp. 3294–3302.

Kottur, Satwik, José M. F. Moura, Devi Parikh, Dhruv Batra, and Marcus Rohrbach (2019). "CLEVR-Dialog: A Diagnostic Dataset for Multi-Round Reasoning in Visual Dialog". 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)*. Minneapolis, Minnesota: Association for Computational Linguistics, pp. 582–595. DOI: `10.18653/v1/N19-1058`. URL: `https://www.aclweb.org/anthology/N19-1058`.

Krizhevsky, Alex (2009). "Learning Multiple Layers of Features from Tiny Images". In.

Kuehne, H., H. Jhuang, E. Garrote, T. Poggio, and T. Serre (2011). "HMDB: a large video database for human motion recognition". In: *Proceedings of the International Conference on Computer Vision (ICCV)*.

Lecun, Y., L. Bottou, Y. Bengio, and P. Haffner (1998). "Gradient-based learning applied to document recognition". In: *Proceedings of the IEEE* 86.11, pp. 2278–2324. DOI: `10.1109/5.726791`.

Lee, Sang-Woo, Tong Gao, Sohee Yang, Jaejun Yoo, and Jung-Woo Ha (2019). "Large-Scale Answerer in Questioner's Mind for Visual Dialog Question Generation". In: *International Conference on Learning Representations*. URL: `https://openreview.net/forum?id=rkgT3jRct7`.

Lee, Sang-Woo, Yu-Jung Heo, and Byoung-Tak Zhang (2018). "Answerer in Questioner's Mind: Information Theoretic Approach to Goal-Oriented Visual Dialog". In: *Advances in Neural Information Processing Systems*. Ed. by S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett. Vol. 31. Curran Associates, Inc. URL: `https://proceedings.neurips.cc/paper/2018/file/0829424ffa0d3a2547b6c9622c77de03-Paper.pdf`.

Lesniak, Damian, Igor Sieradzki, and Igor T. Podolak (2019). "Distribution-Interpolation Trade off in Generative Models". In: *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*.

Lewis, Mike, Denis Yarats, Yann Dauphin, Devi Parikh, and Dhruv Batra (2017). "Deal or No Deal? End-to-End Learning of Negotiation Dialogues". In: *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. Copenhagen, Denmark, pp. 2443–2453. DOI: `10.18653/v1/D17-1259`. URL: `http://aclweb.org/anthology/D17-1259`.

Li, Jiwei, Will Monroe, Tianlin Shi, Sébastien Jean, Alan Ritter, and Dan Jurafsky (2017a). "Adversarial Learning for Neural Dialogue Generation". In: *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. Copenhagen, Denmark, pp. 2157–2169. DOI: `10.18653/v1/D17-1230`. URL: `http://aclweb.org/anthology/D17-1230`.

Li, Xiujun, Yun-Nung Chen, Lihong Li, Jianfeng Gao, and Asli Celikyilmaz (2017b). "End-to-End Task-Completion Neural Dialogue Systems". In: *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Taipei, Taiwan, pp. 733–743. URL: `http://aclweb.org/anthology/I17-1074`.

Li, Xiujun, Chunyuan Li, Qiaolin Xia, Yonatan Bisk, Asli Celikyilmaz, Jianfeng Gao, Noah Smith, and Yejin Choi (2019). "Robust Navigation with Language Pretraining and Stochastic Sampling". In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*. URL: `arXiv:1909.02244`.

Liao, Lizi, Yunshan Ma, Xiangnan He, Richang Hong, and Tat-Seng Chua (2018). "Knowledge-aware Multimodal Dialogue Systems". In: *Proceedings of the 26th ACM international conference on Multimedia*. Seoul, Republic of Korea, pp. 801–809. DOI: `10.1145/3240508.3240605`.

Locatello, Francesco, Ben Poole, Gunnar Raetsch, Bernhard Schölkopf, Olivier Bachem, and Michael Tschannen (2020). "Weakly-Supervised Disentanglement Without Compromises". In: *Proceedings of the 37th International Conference on Machine Learning*. Vol. 119. Proceedings of Machine Learning Research. PMLR, pp. 6348–6359.

Loshchilov, Ilya and Frank Hutter (2017). "SGDR: Stochastic Gradient Descent with Warm Restarts". In: *International Conference on Learning Representations*. URL: `https://openreview.net/forum?id=Skq89Scxx`.

Loshchilov, Ilya and Frank Hutter (2018). *Fixing Weight Decay Regularization in Adam*. URL: `https://openreview.net/forum?id=rk6qdGgCZ`.

Lu, Jiasen, Dhruv Batra, Devi Parikh, and Stefan Lee (2019). "ViLBERT: Pretraining Task-Agnostic Visiolinguistic Representations for Vision-and-Language Tasks". In: *Advances in Neural Information Processing Systems*. Ed. by H. Wallach, H.

Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett. Vol. 32. Curran Associates, Inc. URL: `https://proceedings.neurips.cc/paper/2019/file/c74d97b01eae257e44aa9d5bade97baf-Paper.pdf`.

Luong, Thang, Hieu Pham, and Christopher D. Manning (2015). "Effective Approaches to Attention-based Neural Machine Translation". In: *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. Lisbon, Portugal, pp. 1412–1421. DOI: `10.18653/v1/D15-1166`. URL: `http://aclweb.org/anthology/D15-1166`.

Luong, Thang, Ilya Sutskever, Quoc Le, Oriol Vinyals, and Wojciech Zaremba (2015). "Addressing the Rare Word Problem in Neural Machine Translation". In: *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Beijing, China: Association for Computational Linguistics, pp. 11–19. DOI: `10.3115/v1/P15-1002`. URL: `https://www.aclweb.org/anthology/P15-1002`.

Ma, Chih-Yao, Jiasen Lu, Zuxuan Wu, Ghassan AlRegib, Zsolt Kira, Richard Socher, and Caiming Xiong (2019a). "Self-Monitoring Navigation Agent via Auxiliary Progress Estimation". In: *Proceedings of the International Conference on Learning Representations (ICLR)*. URL: `https://arxiv.org/abs/1901.03035`.

Ma, Chih-Yao, Zuxuan Wu, Ghassan AlRegib, Caiming Xiong, and Zsolt Kira (2019b). "The Regretful Agent: Heuristic-Aided Navigation Through Progress Estimation". In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Majumder, Navonil, Soujanya Poria, Devamanyu Hazarika, Rada Mihalcea, Alexander F. Gelbukh, and Erik Cambria (2018). "DialogueRNN: An Attentive RNN for Emotion Detection in Conversations". In: *AAAI*.

Manjunatha, Varun, Nirat Saini, and Larry S. Davis (2019). "Explicit Bias Discovery in Visual Question Answering Models". In: *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pp. 9562–9571.

Maurer, Andreas and Massimiliano Pontil (2009). “Empirical Bernstein bounds and sample variance penalization”. In: *arXiv preprint arXiv:0907.3740*.

Minaee, Shervin, Yuri Y. Boykov, Fatih Porikli, Antonio J Plaza, Nasser Kehtarnavaz, and Demetri Terzopoulos (2021). “Image Segmentation Using Deep Learning: A Survey”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–1. DOI: 10.1109/TPAMI.2021.3059968.

Misra, Ishan, Ross Girshick, Rob Fergus, Martial Hebert, Abhinav Gupta, and Laurens van der Maaten (2018). “Learning by Asking Questions”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Mnih, Volodymyr, Adrià Puigdomènech Badia, Mehdi Mirza, Alex Graves, Tim Harley, Timothy P. Lillicrap, David Silver, and Koray Kavukcuoglu (2016a). “Asynchronous Methods for Deep Reinforcement Learning”. In: *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*. ICML’16. New York, NY, USA: JMLR.org, 1928–1937.

Mnih, Volodymyr, Adrià Puigdomènech Badia, Mehdi Mirza, Alex Graves, Tim Harley, Timothy P. Lillicrap, David Silver, and Koray Kavukcuoglu (2016b). “Asynchronous Methods for Deep Reinforcement Learning”. In: *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*. ICML’16. New York, NY, USA: JMLR.org, 1928–1937.

Mnih, Volodymyr, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin Riedmiller, Andreas K. Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Helen Antonoglou Ioannis King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis (2015). “Human-level control through deep reinforcement learning”. In: *Nature* 518, pp. 529–533. DOI: 10.1038/nature14236. URL: https://doi.org/10.1038/nature14236.

Moosavi-Dezfooli, Seyed-Mohsen, Alhussein Fawzi, and Pascal Frossard (2016). “DeepFool: A Simple and Accurate Method to Fool Deep Neural Networks”. In:

Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

Netzer, Yuval, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y. Ng (2011). "Reading Digits in Natural Images with Unsupervised Feature Learning". In: *NIPS Workshop on Deep Learning and Unsupervised Feature Learning 2011*. URL: `http://ufldl.stanford.edu/housenumbers/nips2011_housenumbers.pdf`.

Nguyen, Khanh and Hal Daumé III (2019). "Help, Anna! Visual Navigation with Natural Multimodal Assistance via Retrospective Curiosity-Encouraging Imitation Learning". In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*. URL: `https://arxiv.org/abs/1909.01871`.

Oberst, Michael and David Sontag (2019). "Counterfactual Off-Policy Evaluation with Gumbel-Max Structural Causal Models". In: *Proceedings of the 36th International Conference on Machine Learning*. Ed. by Kamalika Chaudhuri and Ruslan Salakhutdinov. Vol. 97. Proceedings of Machine Learning Research. Long Beach, California, USA: PMLR, pp. 4881–4890. URL: `http://proceedings.mlr.press/v97/oberst19a.html`.

Parvaneh, Amin, Ehsan Abbasnejad, Damien Teney, Javen Shi, and Anton van den Hengel (2020). "Counterfactual Vision-and-Language Navigation: Unravelling the Unseen". In: *Advances in Neural Information Processing Systems*. Vol. 33.

Pearl, Judea (2009). *Causality: Models, Reasoning and Inference*. 2nd. New York, NY, USA: Cambridge University Press. ISBN: 052189560X, 9780521895606.

Peng, Xingchao, Qinxun Bai, Xide Xia, Zijun Huang, Kate Saenko, and Bo Wang (2019). "Moment matching for multi-source domain adaptation". In: *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1406–1415.

Pennington, Jeffrey, Richard Socher, and Christopher Manning (2014). "GloVe: Global Vectors for Word Representation". In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Doha, Qatar:

Association for Computational Linguistics, pp. 1532–1543. DOI: 10.3115/v1/D14-1162. URL: https://www.aclweb.org/anthology/D14-1162.

Potamianos, A., E. Fosler-Lussier, E. Ammicht, and M. Perakakis (2007). “Information Seeking Spoken Dialogue Systems— Part II: Multimodal Dialogue”. In: *IEEE Transactions on Multimedia* 9.3, pp. 550–566.

Qin, Lianhui, Antoine Bosselut, Ari Holtzman, Chandra Bhagavatula, Elizabeth Clark, and Yejin Choi (2019). “Counterfactual Story Reasoning and Generation”. In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Radford, Alec, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever (2019). “Language Models are Unsupervised Multitask Learners”. In.

Ramakrishnan, Sainandan, Aishwarya Agrawal, and Stefan Lee (2018). “Overcoming Language Priors in Visual Question Answering with Adversarial Regularization”. In: *Proceedings of the 32Nd International Conference on Neural Information Processing Systems*. NIPS’18. Montreal, Canada: Curran Associates Inc., pp. 1548–1558.

Reddy, Siva, Danqi Chen, and Christopher D. Manning (2018). “CoQA: A Conversational Question Answering Challenge”. In: *rXiv:1808.07042v1*.

Ren, Pengzhen, Yun Xiao, Xiaojun Chang, Po-Yao Huang, Zhihui Li, Brij B. Gupta, Xiaojiang Chen, and Xin Wang (2021). “A Survey of Deep Active Learning”. In: *ACM Comput. Surv.* 54.9. ISSN: 0360-0300. DOI: 10.1145/3472291. URL: https://doi.org/10.1145/3472291.

Roth, Dan and Kevin Small (2006). “Margin-Based Active Learning for Structured Output Spaces”. In: *Machine Learning: ECML 2006*. Ed. by Johannes Fürnkranz, Tobias Scheffer, and Myra Spiliopoulou. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 413–424. ISBN: 978-3-540-46056-5.

Sachin Ravi, Hugo Larochelle (2017). “Optimization as a Model for Few-Shot Learning”. In: *International Conference on Learning Representations*.

Saha, Amrita, Mitesh M. Khapra, and Karthik Sankaranarayanan (2018). “Towards Building Large Scale Multimodal Domain-Aware Conversation Systems”. In: *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*. Ed. by Sheila A. McIlraith and Kilian Q. Weinberger. AAAI Press, pp. 696–704. URL: `https://www.aaai.org/ocs/index.php/AAAI/AAAI18/paper/view/17104`.

Savva, Manolis, Abhishek Kadian, Oleksandr Maksymets, Yili Zhao, Erik Wijmans, Bhavana Jain, Julian Straub, Jia Liu, Vladlen Koltun, Jitendra Malik, Devi Parikh, and Dhruv Batra (2019). “Habitat: A Platform for Embodied AI Research”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*.

Schulman, John, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov (2017). *Proximal Policy Optimization Algorithms*. arXiv: `1707.06347 [cs.LG]`.

Sener, Ozan and Silvio Savarese (2018). “Active Learning for Convolutional Neural Networks: A Core-Set Approach”. In: *International Conference on Learning Representations*. URL: `https://openreview.net/forum?id=H1aIuk-RW`.

Settles, Burr (2009). “Active learning literature survey”. In.

Shrestha, Robik, Kushal Kafle, and Christopher Kanan (2019). “Answer Them All! Toward Universal Visual Question Answering Models”. In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Simonyan, Karen and Andrew Zisserman (2014). “Very deep convolutional networks for large-scale image recognition”. In: *arXiv preprint arXiv:1409.1556*.

Singh, Amanpreet, Vivek Natarajan, Yu Jiang, Xinlei Chen, Meet Shah, Marcus Rohrbach, Dhruv Batra, and Devi Parikh (2018). “Pythia-a platform for vision & language research”. In: *SysML Workshop, NeurIPS*. Vol. 2018.

Singh, Amanpreet, Vivek Natarajan, Meet Shah, Yu Jiang, Xinlei Chen, Dhruv Batra, Devi Parikh, and Marcus Rohrbach (2019). “Towards VQA Models That Can Read”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 8317–8326. URL: `https://academic.microsoft.com/paper/2979382951`.

Sinha, Samarth, Sayna Ebrahimi, and Trevor Darrell (2019). “Variational Adversarial Active Learning”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*.

Socher, Richard, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts (2013). “Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank”. In: *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*. Seattle, Washington, USA: Association for Computational Linguistics, pp. 1631–1642. URL: `https://www.aclweb.org/anthology/D13-1170`.

Sordoni, Alessandro, Yoshua Bengio, Hossein Vahabi, Christina Lioma, Jakob Grue Simonsen, and Jian-Yun Nie (2015). “A Hierarchical Recurrent Encoder-Decoder for Generative Context-Aware Query Suggestion”. In: *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*. CIKM ’15. Melbourne, Australia: Association for Computing Machinery, 553–562. ISBN: 9781450337946. DOI: `10.1145/2806416.2806493`. URL: `https://doi.org/10.1145/2806416.2806493`.

Sukhbaatar, Sainbayar, arthur szlam, Jason Weston, and Rob Fergus (2015). “End-To-End Memory Networks”. In: *Advances in Neural Information Processing Systems 28*. Ed. by C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett. Curran Associates, Inc., pp. 2440–2448. URL: `http://papers.nips.cc/paper/5846-end-to-end-memory-networks.pdf`.

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

Swaminathan, Adith and Thorsten Joachims (2015). "Counterfactual Risk Minimization: Learning from Logged Bandit Feedback". In: *Proceedings of the 32nd International Conference on Machine Learning*. Ed. by Francis Bach and David Blei. Vol. 37. Proceedings of Machine Learning Research. Lille, France: PMLR, pp. 814–823.

Tai, Kai Sheng, Richard Socher, and Christopher D. Manning (2015). "Improved Semantic Representations From Tree-Structured Long Short-Term Memory Networks". In: *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Beijing, China: Association for Computational Linguistics, pp. 1556–1566. DOI: `10.3115/v1/P15-1150`. URL: `https://www.aclweb.org/anthology/P15-1150`.

Tan, Hao, Licheng Yu, and Mohit Bansal (2019). "Learning to Navigate Unseen Environments: Back Translation with Environmental Dropout". 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)*. Minneapolis, Minnesota: Association for Computational Linguistics, pp. 2610–2621. DOI: `10.18653/v1/N19-1268`. URL: `https://www.aclweb.org/anthology/N19-1268`.

Tang, Jianheng, Tiancheng Zhao, Chenyan Xiong, Xiaodan Liang, Eric Xing, and Zhiting Hu (2019). "Target-Guided Open-Domain Conversation". In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Florence, Italy: Association for Computational Linguistics, pp. 5624–5634. DOI: `10.18653/v1/P19-1565`. URL: `https://www.aclweb.org/anthology/P19-1565`.

Thomason, Jesse, Michael Murray, Maya Cakmak, and Luke Zettlemoyer (2020). "Vision-and-Dialog Navigation". In: *Proceedings of the Conference on Robot Learning*. Vol. 100. Proceedings of Machine Learning Research. PMLR, pp. 394–

406. URL: https://proceedings.mlr.press/v100/thomason20a.html.

Thorndike, L and Darryl Bruce (2017). *Animal intelligence: Experimental studies*. Routledge.

Tsai, T. J., A. Stolcke, and M. Slaney (2015). “A Study of Multimodal Addressee Detection in Human-Human-Computer Interaction”. In: *IEEE Transactions on Multimedia* 17.9, pp. 1550–1561.

Verma, Vikas, Alex Lamb, Christopher Beckham, Amir Najafi, Ioannis Mitliagkas, David Lopez-Paz, and Yoshua Bengio (2019a). “Manifold Mixup: Better Representations by Interpolating Hidden States”. In: *Proceedings of the 36th International Conference on Machine Learning*. Vol. 97. Proceedings of Machine Learning Research. Long Beach, California, USA: PMLR, pp. 6438–6447. URL: http://proceedings.mlr.press/v97/verma19a.html.

Verma, Vikas, Alex Lamb, Juho Kannala, Yoshua Bengio, and David Lopez-Paz (July 2019b). “Interpolation Consistency Training for Semi-supervised Learning”. In: *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*. International Joint Conferences on Artificial Intelligence Organization, pp. 3635–3641. DOI: 10.24963/ijcai.2019/504. URL: https://doi.org/10.24963/ijcai.2019/504.

Vries, Harm de, Kurt Shuster, Dhruv Batra, Devi Parikh, Jason Weston, and Douwe Kiela (2018). “Talk the Walk: Navigating New York City through Grounded Dialogue”. In: *CoRR* abs/1807.03367. arXiv: 1807.03367. URL: http://arxiv.org/abs/1807.03367.

Vries, Harm de, Florian Strub, A. P. Sarath Chandar, Olivier Pietquin, Hugo Larochelle, and Aaron C. Courville (2017). “GuessWhat?! Visual Object Discovery through Multi-modal Dialogue”. In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4466–4475.

Wang, D. and Y. Shang (2014). “A new active labeling method for deep learning”. In: *2014 International Joint Conference on Neural Networks (IJCNN)*, pp. 112–119. DOI: `10.1109/IJCNN.2014.6889457`.

Wang, Mei and Weihong Deng (2018). “Deep visual domain adaptation: A survey”. In: *Neurocomputing* 312, pp. 135–153. ISSN: 0925-2312. DOI: `https://doi.org/10.1016/j.neucom.2018.05.083`. URL: `https://www.sciencedirect.com/science/article/pii/S0925231218306684`.

Wang, Xin, Qiuyuan Huang, Asli Celikyilmaz, Jianfeng Gao, Dinghan Shen, Yuan-Fang Wang, William Yang Wang, and Lei Zhang (2019a). “Reinforced Cross-Modal Matching and Self-Supervised Imitation Learning for Vision-Language Navigation”. In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Wang, Xin, Wenhan Xiong, Hongmin Wang, and William Yang Wang (2018). “Look Before You Leap: Bridging Model-Free and Model-Based Reinforcement Learning for Planned-Ahead Vision-and-Language Navigation”. In: *The European Conference on Computer Vision (ECCV)*.

Wang, Xuewei, Weiyan Shi, Richard Kim, Yoojung Oh, Sijia Yang, Jingwen Zhang, and Zhou Yu (2019b). “Persuasion for Good: Towards a Personalized Persuasive Dialogue System for Social Good”. In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Florence, Italy: Association for Computational Linguistics, pp. 5635–5649. DOI: `10.18653/v1/P19-1566`. URL: `https://www.aclweb.org/anthology/P19-1566`.

Wei, Wei, Quoc Le, Andrew Dai, and Jia Li (2018). “AirDialogue: An Environment for Goal-Oriented Dialogue Research”. In: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Brussels, Belgium, pp. 3844–3854. URL: `http://aclweb.org/anthology/D18-1419`.

Wen, Tsung-Hsien, David Vandyke, Nikola Mrkšić, Milica Gašić, Lina M. Rojas-Barahona, Pei-Hao Su, Stefan Ultes, and Steve Young (2017). “A Network-based End-to-End Trainable Task-oriented Dialogue System”. In: *Proceedings of the*

15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers.

Wijmans, Erik, Samyak Datta, Oleksandr Maksymets, Abhishek Das, Georgia Gkioxari, Stefan Lee, Irfan Essa, Devi Parikh, and Dhruv Batra (2019). “Embodied Question Answering in Photorealistic Environments With Point Cloud Perception”. In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Williams, Ronald J. (1992). “Simple statistical gradient-following algorithms for connectionist reinforcement learning”. In: *Machine Learning* 8.3, pp. 229–256. DOI: `https://doi.org/10.1007/BF00992696`.

Woods, Walt, Jack Chen, and Christof Teuscher (2019). “Adversarial explanations for understanding image classification decisions and improved neural network robustness”. In: *Nature Machine Intelligence* 1.11, pp. 508–516.

Wortsman, Mitchell, Kiana Ehsani, Mohammad Rastegari, Ali Farhadi, and Roozbeh Mottaghi (2019). “Learning to Learn How to Learn: Self-Adaptive Visual Navigation Using Meta-Learning”. In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Wu, Qi, Damien Teney, Peng Wang, Chunhua Shen, Anthony Dick, and Anton van den Hengel (2017a). “Visual question answering: A survey of methods and datasets”. In: *Computer Vision and Image Understanding* 163. Language in Vision, pp. 21–40. ISSN: 1077-3142. DOI: `https://doi.org/10.1016/j.cviu.2017.05.001`. URL: `https://www.sciencedirect.com/science/article/pii/S1077314217300772`.

Wu, Qi, Damien Teney, Peng Wang, Chunhua Shen, Anthony Dick, and Anton van den Hengel (2017b). “Visual question answering: A survey of methods and datasets”. In: *Computer Vision and Image Understanding* 163. Language in Vision, pp. 21–40. ISSN: 1077-3142. DOI: `https://doi.org/10.1016/j.cviu.2017.05.001`. URL: `https://www.sciencedirect.com/science/article/pii/S1077314217300772`.

Wu, Yi, Yuxin Wu, Georgia Gkioxari, and Yuandong Tian (2018). *Building Generalizable Agents with a Realistic and Rich 3D Environment*. URL: `https://openreview.net/forum?id=rkaT3zWCZ`.

Xie, Cihang, Mingxing Tan, Boqing Gong, Jiang Wang, Alan Yuille, and Quoc V. Le (2020). "Adversarial Examples Improve Image Recognition". In: *arXiv preprint arXiv:1911.09665*.

Xing, Chen, Yu Wu, Wei Wu, Yalou Huang, and Ming Zhou (2018). "Hierarchical Recurrent Attention Network for Response Generation". In: *AAAI Conference on Artificial Intelligence*. URL: `https://www.aaai.org/ocs/index.php/AAAI/AAAI18/paper/view/16510`.

Xu, Lin, Qixian Zhou, Ke Gong, Xiaodan Liang, Jianheng Tang, and Liang Lin (2019). "End-to-End Knowledge-Routed Relational Dialogue System for Automatic Diagnosis". In: *AAAI*.

Yang, Yi, Zhigang Ma, Feiping Nie, Xiaojun Chang, and Alexander G. Hauptmann (2015). "Multi-Class Active Learning by Uncertainty Sampling with Diversity Maximization". In: *International Journal of Computer Vision*.

Yao, Rui, Guosheng Lin, Shixiong Xia, Jiaqi Zhao, and Yong Zhou (2020). "Video Object Segmentation and Tracking: A Survey". In: *ACM Trans. Intell. Syst. Technol.* 11.4. ISSN: 2157-6904. DOI: `10.1145/3391743`. URL: `https://doi.org/10.1145/3391743`.

Yoo, Donggeun and In So Kweon (2019). "Learning Loss for Active Learning". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

You, Quanzeng, Hailin Jin, Zhaowen Wang, Chen Fang, and Jiebo Luo (2016). "Image Captioning With Semantic Attention". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Zhang, Hongyi, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz (2018a). "Mixup: Beyond Empirical Risk Minimization". In: *International Conference on*

Learning Representations. URL: https://openreview.net/forum?id=r1Ddp1-Rb.

Zhang, Ruixiang, Tong Che, Zoubin Ghahramani, Yoshua Bengio, and Yangqiu Song (2018b). "MetaGAN: An Adversarial Approach to Few-shot Learning". In: *Proceedings of the 32Nd International Conference on Neural Information Processing Systems*. NIPS'18. Montr&#233;al, Canada: Curran Associates Inc., pp. 2371–2380. URL: http://dl.acm.org/citation.cfm?id=3327144.3327163.

Zhao, Zhong-Qiu, Peng Zheng, Shou-Tao Xu, and Xindong Wu (2019). "Object Detection With Deep Learning: A Review". In: *IEEE Transactions on Neural Networks and Learning Systems* 30.11, pp. 3212–3232. DOI: 10.1109/TNNLS.2018.2876865.

Zhou, Kaiyang, Yongxin Yang, Yu Qiao, and Tao Xiang (2021). "Domain Generalization with MixStyle". In: *International Conference on Learning Representations*. URL: https://openreview.net/forum?id=6xHJ37MVxxp.

Zue, Victor W. and James R. Glass (2000). "Conversational interfaces: advances and challenges". In: *Proceedings of the IEEE*. Beijing, China, pp. 1166–1180. DOI: 10.1109/5.880078.