

UCLA

UCLA Electronic Theses and Dissertations

Title

Multimodal Conversation Modeling via Neural Perception, Structure Learning, and Communication

Permalink

<https://escholarship.org/uc/item/3fn5f194>

Author

Zheng, Zilong

Publication Date

2021

Peer reviewed|Thesis/dissertation

UNIVERSITY OF CALIFORNIA

Los Angeles

Multimodal Conversation Modeling
via Neural Perception, Structure Learning, and Communication

A dissertation submitted in partial satisfaction
of the requirements for the degree
Doctor of Philosophy in Computer Science

by

Zilong Zheng

2021

© Copyright by
Zilong Zheng
2021

ABSTRACT OF THE DISSERTATION

Multimodal Conversation Modeling via Neural Perception, Structure Learning, and Communication

by

Zilong Zheng

Doctor of Philosophy in Computer Science

University of California, Los Angeles, 2021

Professor Song-Chun Zhu, Chair

Multimodal conversation modeling is an important and challenging problem when building conversational agents. Pioneer works mostly focus on end-to-end multimodal fusion techniques, which require large volumes of pairwise data and lacks interpretability. This dissertation aims at closing the loop of vision and language multimodal modeling from the perspectives of neural perception, structure learning, and communication. Specifically, it makes four major contributions:

1. We explicitly model the joint distribution of vision and language as a Gibbs distribution. Then, we propose an “analysis by synthesis” cooperative training schema that uses the learned joint distribution to sample from one modality to another, *e.g.* category to image, attribute to image, *etc.* Further, we argue that such a training paradigm can be explained in the cognitive theory, where the conditional generator is a fast-thinking initializer that provides a rough output and the sampling process is a slow-thinking solver that refines the output with detailed multimodal information.
2. We propose to view the multimodal dialogue as a graph, where each node is a round of dia-

logue and the edges represent the semantic dependencies among dialogue turns. Moreover, we propose an Expectation-Maximization (EM)-based algorithm that can both predict partially observed nodes and infer graph structures. We show that such an unsupervised structure learning paradigm can provide post-hoc interpretability to various multimodal dialogue tasks.

3. We present a crucial but barely discussed challenge – implicature and pragmatics – in the field of conversational reasoning. We show that human communicate based on their intents and beliefs, where implicatures commonly come along. Considering the missing gap in the current natural language community, we propose a dataset generation protocol based on Spatial-Temporal And-Or-Graphs (ST-AOGs). We show that most of the state-of-the-art language models result in a large performance gap compared with humans.
4. We present a human-robot collaboration task – bomb defusing game, that requires explanation to help human understand machine’s behavior. We argue that such explanations should be generated according to the user’s mental preferences, *i.e.* utilities. Therefore, we propose an explanation generation algorithm based on Hidden Markov Model (HMM), which considers the user’s mental utilities as a hidden variable that changes based on observations. We show that, compared with rule-based conversational system, our generated explanations are more natural and are helpful in gaining human trust.

The dissertation of Zilong Zheng is approved.

Kai-Wei Chang

Demetri Terzopoulos

Ying Nian Wu

Song-Chun Zhu, Committee Chair

University of California, Los Angeles

2021

*To my parents and Yifan Zhang ...
for their continual and unconditional support*

TABLE OF CONTENTS

1	Introduction	1
I	Multimodal Neural Perception	3
2	Multimodal Representation Learning with Deep Generative ConvNets .	4
2.1	Introduction	4
2.2	Related work	9
2.3	Cooperative conditional learning	11
2.3.1	Slow thinking solver	12
2.3.2	Fast thinking initializer	13
2.3.3	Cooperative training of initializer and solver	14
2.4	Theoretical underpinning	15
2.4.1	Kullback-Leibler divergence	15
2.4.2	Slow thinking solver	16
2.4.3	Fast thinking initializer	17
2.4.4	Objective shift: modified contrastive divergence	18
2.4.5	Mapping shift: distilling MCMC	18
2.5	Experiments	19
2.5.1	Experiment 1: Category \rightarrow Image	20
2.5.2	Experiment 2: Image \rightarrow Image	30
2.6	Conclusion	40

II Structure Learning **41**

3 Reasoning Visual Dialogue with Structural and Partial Observations . . . 42

3.1 Related Work 44

3.2 Our Approach 47

3.2.1 Dialogue as Markov Random Field 49

3.2.2 Inference with Partial Observation 50

3.2.3 MRF with Partial Observations 50

3.2.4 GNN with Partial Observations 52

3.2.5 Network Architecture 53

3.3 Experiments 56

3.3.1 Performance on VisDial v0.9 56

3.3.2 Performance on VisDial v1.0 59

3.3.3 Performance on VisDial-Q Dataset [DKG17, JLS18] 60

3.3.4 Diagnostic Experiments 61

3.4 Conclusion 62

III Communication with Theory of Minds **63**

4 GRICE: A Grammar-based Dataset for Recovering Implicature and Conversational Reasoning 64

4.1 Introduction 64

4.2 Related Work 68

4.3 Task Definition 70

4.4 Creating the GRICE Dataset 72

4.5	Experiments	76
4.5.1	Baseline Models	77
4.5.2	Evaluation and Results	78
5	Generating Explanations with Human Utility	82
5.1	Overview and Background	82
5.1.1	Background and Motivation	82
5.1.2	Overview	84
5.2	Methodology	87
5.2.1	Computational Model	87
5.2.2	Participants Description	96
5.2.3	Study Design/Procedure/Measurement	97
5.2.4	Instruments/Materials	99
5.2.5	Hypotheses	103
5.3	Results	104
5.4	Summary and Conclusions	105
6	Conclusion	107
	References	109

LIST OF FIGURES

2.1	Diagram of fast thinking and slow thinking conditional learning.	5
2.2	Learning step of cooperative training.	6
2.3	Network architecture of category-to-image.	21
2.4	Generated MNIST handwritten digits (left) and fashion MNIST images (right).	23
2.5	Image generation by the models at different training epochs.	24
2.6	Model analysis on fashion-MNIST dataset.	26
2.7	Generated Cifar-10 object images.	28
2.8	Style transfer on SVHN dataset.	30
2.9	Network architecture (image-to-image translation).	31
2.10	Generating images conditioned on architectural labels.	35
2.11	Sketch-to-photo face synthesis on CUHK dataset.	36
2.12	Results on edges \rightarrow shoes generation, compared to ground truth.	37
2.13	Results of photo inpainting on CMP Facade dataset.	39
3.1	An illustration of the visual dialogue task.	43
3.2	Illustration of GNN representation of visual dialogue.	48
3.3	A detailed illustration of our model.	51
3.4	Qualitative results of our model on VisDial v0.9 [DKG17].	58
4.1	An example of the conversation in the proposed GRICE dataset. Each round of dialogue includes a question, an answer that may contain implicature, and a recovered statement that converts the implicature to explicature. Different colors highlight coreference flows.	65

4.2	Examples of two tasks defined in GRICE dataset. (a) Given a multi-round open-dialogue, an algorithm is asked to perform (b) implicature recovery and (c) conversational reasoning in the form of QAs.	71
4.3	The graphical illustration of the grammar production rules for the GRICE dataset.	71
4.4	The candidate answers for the implicature recovery task are generated following four different strategies: 1. Statements that are similar to the ground-truth condition but with wrong coreferenced entities. 2. Random sampled true condition but with irrelevant facts. 3. Random sampled wrong facts from the conversational context. 4. Manually created statements that are close to the true condition but are in fact wrong.	75
4.5	Performance comparison between MemNN and with additional inference module (MemNN w/ inf) that explicitly recovers the implicature.	79
5.1	Algorithmic flow of the computational model.	87
5.2	Temporal Evolution of Explanation Generation.	94
5.3	User study flow.	97
5.4	User interface of the scout exploration game.	100
5.5	Example interfaces for the value function question and the behavior prediction question.	102

LIST OF TABLES

2.1	The Fréchet Inception Distance (FID) scores of different models trained on MNIST and fashion-MNIST dataset.	25
2.2	Comparison of computational time (in seconds) per epoch on fashion-MNIST dataset.	26
2.3	Inception scores of different models trained on Cifar-10 dataset.	29
2.4	Human perceptual tests for image-to-image synthesis.	36
2.5	Comparison with the baseline methods for image inpainting on the CMP Facade dataset and Paris streetview dataset.	38
2.6	Comparison of model complexity with the baseline methods for image inpainting on CMP Facade dataset.	38
3.1	Quantitative evaluation of discriminative methods on val split of VisDial v0.9.	57
3.2	Quantitative evaluation of discriminative methods on test-standard split of VisDial v1.0.	59
3.3	Quantitative evaluation on VisDial-Q dataset with VisDial-Q evaluation protocol.	60
3.4	Ablation study of the key components of our methods on VisDial v0.9 dataset.	62
4.1	Comparing GRICE with existing conversational datasets.	68
4.2	Categories and examples of different subtopics in GRICE dataset.	74
4.3	Distribution of implicature types (%).	74
4.4	Performance on implicature recovery task.	79
4.5	Performance on conversational reasoning task.	79
5.1	Notations adopted in the computational model.	89

ACKNOWLEDGMENTS

I would like to first express my sincere thanks to all the professors and teachers who gave me help:

My advisor, Dr. Song-Chun Zhu, for his insightful guidance, enlightening advice, endless encouragement, and selfless help throughout my Ph.D. study.

Dr. Ying Nian Wu, for his patience and valuable advice when I lost the direction, for his encouraging comments on my research work, and for his tutorials on advances of machine learning.

Dr. Ping Li, for his guidance on machine learning and writing skills during my internship, and all the weekly discussions that cover various topics.

Next, I would like to thank my collaborators and labmates:

Dr. Jianwen Xie, my first mentor in machine learning, for the careful tutorial on generative models and my first CVPR publication, as well as for all the collaboration throughout my PhD time.

Dr. Yixin Zhu, for leading me to think the insights of research work, guiding me to cognitive and fundamental research, teaching me various writing and reading techniques.

Dr. Wenguan Wang, for teaching me various deep learning techniques, sharing ideas with me no matter where you are, and for all the happy time that we experience together in LA restaurants.

Dr. Changsong Liu, Hanlin Zhu, for leading me into the VCLA lab, and discussions on natural language processing and dialogue managements.

Dr. Hangxin Liu, Dr. Tao Yuan, Dr. Siyuan Huang, Yixin Chen, Baoxiong Jia, Zeyu Zhang, for the time that we spent together during internship, conquering challenges when building intelligent conversational robots.

Dr. Siyuan Qi, Luyao Yuan, Shuwen Qiu, Ruiqi Gao, Yifei Xu, Lifeng Fan, Xu Xie, and

all the other labmates, for the discussions and collaborations that we had together in various AI topics.

Finally, I would like to give my thanks to those who always love and trust me and sacrifice a lot for me – my parents and my fiancée, Yifan Zhang.

Portions of this work were supported by DARPA XAI N66001-17-2-4029, ONR MURI N00014-16-1-2007, ONR MURI N00014-10-1-0933, and NSF IIS-1423305.

VITA

- 2017–2021 Graduate Research Assistant, Computer Science Department, UCLA.
- 2020 Teaching Assistant, Computer Science Department, UCLA.
- 2020 Research Intern, Baidu Research USA, Dr. Ping Li.
- 2017 M.S. (Computer Science), UCLA.
- 2017 Software Engineer Intern, Google.
- 2016 B.A. (Computer Science), University of Minnesota, Twin Cities.
- 2016 B.Eng. (Microelectronic Technology), University of Electronic Science and Technology of China.

PUBLICATIONS

* denotes joint first authors

GRICE: A Grammar-based Dataset for Recovering Implicature and Conversational Reasoning
Zilong Zheng, Shuwen Qiu, Yixin Zhu, Song-Chun Zhu. *Findings of ACL, ACL-IJCNLP*, 2021.

Cooperative Training of Fast Thinking Initializer and Slow Thinking Solver for Multi-Modal Conditional Learning. Jianwen Xie*, **Zilong Zheng***, Xiaolin Fang, Song-Chun Zhu, Ying Nian Wu. *TPAMI*, 2021.

Learning Triadic Belief Dynamics in Nonverbal Communication from Videos. Lifeng Fan, Shuwen Qiu, **Zilong Zheng**, Tao Gao, Song-Chun Zhu, Yixin Zhu. *CVPR*, 2021.

Generative PointNet: Deep Energy-Based Learning on Unordered Point Sets for 3D Generation, Reconstruction and Classification. Jianwen Xie, Yifei Xu, **Zilong Zheng**, Song-Chun Zhu, Ying Nian Wu. *CVPR*, 2021.

Learning Cycle-Consistent Cooperative Networks via Alternating MCMC Teaching for Unsupervised Cross-Domain Translation. Jianwen Xie*, **Zilong Zheng***, Xiaolin Fang, Song-Chun Zhu, Ying Nian Wu. *AAAI*, 2021.

Generative VoxelNet: Learning Energy-Based Models for 3D Shape Synthesis and Analysis. Jianwen Xie*, **Zilong Zheng***, Ruiqi Gao, Wenguan Wang, Song-Chun Zhu, Ying Nian Wu. *TPAMI*, 2020.

Joint Inference of States, Robot Knowledge, and Human (False-)Beliefs. Tao Yuan, Hangxin Liu, Lifeng Fan, **Zilong Zheng**, Tao Gao, Yixin Zhu, Song-Chun Zhu. *ICRA*, 2020.

Motion-Based Generator Model: Unsupervised Disentanglement of Appearance, Trackable and Intrackable Motions in Dynamic Patterns. Jianwen Xie*, Ruiqi Gao*, **Zilong Zheng**, Song-Chun Zhu, Ying Nian Wu. *AAAI*, 2020.

Reasoning Visual Dialogs with Structural and Partial Observations. **Zilong Zheng***, Wenguan Wang*, Siyuan Qi*, Song-Chun Zhu. *CVPR*, 2019.

Learning Dynamic Generator Model by Alternating Back-Propagation Through Time. Jianwen Xie*, Ruiqi Gao*, **Zilong Zheng**, Song-Chun Zhu, Ying Nian Wu. *AAAI*, 2019.

Learning Descriptor Networks for 3D Shape Synthesis and Analysis. Jianwen Xie*, **Zilong Zheng***, Ruiqi Gao, Wenguan Wang, Song-Chun Zhu, Ying Nian Wu. *CVPR*, 2018.

CHAPTER 1

Introduction

In the real world, information is presented in different forms, *e.g.* vision, sound, text, *etc.* These modalities are associated with each other, *i.e.* images with captions and tags, videos with visual and audio signals. Conversational systems, such as intelligent assistants and chat companion systems, often come with multiple sensory perceptions simultaneously. Therefore, it is important to research how to use information from different modalities effectively and meaningfully.

This dissertation focuses on multimodal conversation modeling, aiming at combining modalities with different statistical distribution into a joint representation in daily dialogues. Such modeling is nevertheless challenging due to the different statistical properties of different modalities. For example, visual inputs are continuous and differentiable, while languages are discrete and symbolic, and commonly require external commonsense knowledge. Most previous works on multimodal learning focus on data-driven approaches, which fuses information from different modalities in the hidden layer and evaluate throughout a downstream task, *e.g.* visual question answering (VQA), visual referring expression (VRE). We believe that such an end-to-end training paradigm is not the ultimate solution because (i) such paradigm requires large volumes of manually labeled pairwise data and is time consuming; (ii) the learned model commonly lacks interpretability and doesn't have explicit semantic meanings; (iii) the model is not robust as the failure of alignment can only be solved by adding additional training data; (iv) the model is task-dependent and cannot be applied to similar tasks.

In this dissertation, we propose to model multimodal conversation from three different perspectives: neural perception, structure learning, and communication. We aim at closing the loop in terms of multimodal representation: deep neural networks (DNN) \rightarrow graph structures \rightarrow symbolic communication.

In Chapter 2, we consider the multimodal modeling from the neural perception perspective, where we propose a joint modeling method that explicitly models the joint distribution of vision and language as a Gibbs distribution. Then, we propose an “analysis by synthesis” cooperative training schema that uses the learned joint distribution to sample from one modality to another, *e.g.* category to image, attribute to image, *etc.*

In Chapter 3, we consider the multimodal conversation modeling from the structure learning perspective, where we view the multimodal dialogue as a graph. In this graph, each node is a round of dialogue and the edges represent the semantic dependencies among dialogue turns. We show that such an unsupervised structure learning paradigm can provide post-hoc interpretability to various multimodal dialogue tasks.

In Chapters 4 and 5, we consider the problem of conversation from the cognitive communication perspective, *i.e.* communication with Theory of Mind (ToM). We argue that human speaks based on their intents and beliefs rather than the semantics. We further demonstrate that generating dialogues (*e.g.* explanations) considering human’s mental state, *i.e.* utility, can improve the communication efficiency and human’s trust.

This dissertation is intended to inspire more future work on building explainable models in the field of multimodal conversation modeling, while providing sufficient transparency, interpretability, and systematic generalization. Together with the recent boost on common-sense reasoning in both CV and NLP, we hope to shed some light on building a future intelligent conversational system.

Part I

Multimodal Neural Perception

CHAPTER 2

Multimodal Representation Learning with Deep Generative ConvNets

2.1 Introduction

When we learn to solve a problem, we can learn to directly map the problem to the solution. This amounts to *fast thinking*, which underlies reflexive or impulsive behavior, or muscle memory, and it can happen when one is emotional or under time constraint. We may also learn an objective function or value function that assigns values to candidate solutions, and we optimize the objective function by an iterative algorithm to find the most valuable solution. This amounts to *slow thinking*, which underlies planning, searching or optimal control, and it can happen when one is calm or has time to think through.

In this chapter, we study the supervised learning of the conditional distribution of a high-dimensional output given an input, where the output and input may belong to two different domains. For instance, the output may be an image, while the input may be a class label, a descriptive text, or an image from another domain. The input defines the problem, and the output is the solution. We also refer to the input as the source or condition, and the output as the target.

We solve this problem by learning two models, an initializer and a solver, cooperatively. The initializer generates the output directly by a non-linear transformation of the input as well as a noise vector, where the noise vector is to account for variability or uncertainty in the output. This amounts to fast thinking because the conditional generation is accomplished

by direct mapping. The solver learns an objective function in the form of a conditional energy function, so that the output can be generated by optimizing the objective function, or more rigorously by sampling from the conditional energy-based model (EBM), where the sampling is to account for variability and uncertainty. This amounts to slow thinking because the sampling is accomplished by an iterative algorithm Markov Chain Monte Carlo (MCMC) [Liu08, BZ20], such as Langevin Dynamics [Nea11]. We propose to learn the two models jointly, where the initializer serves to initialize the sampling process of the solver, and the solver refines the initial solution by an iterative sampling process. The solver learns from the difference between the refined solution and the observed solution, while the initializer learns from the difference between the initial solution and the refined solution.

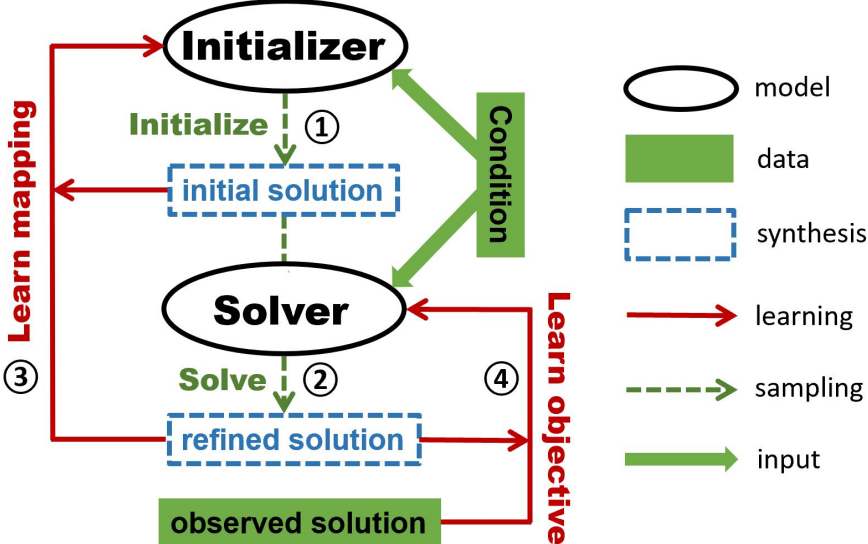


Figure 2.1: Diagram of fast thinking and slow thinking conditional learning.

Fig. 2.1 conveys the basic idea of cooperative learning, which iterates over two steps, a solving step and a learning step. The solving step consists of two stages: 1) Initialize: the initializer generates the initial solution according to the given condition by direct mapping, such as ancestral sampling; 2) Solve: the solver refines the initial solution according to the same condition by an iterative algorithm, such as Langevin sampling, which minimizes the objective function. The learning step also consists of two parts: 1) Learn-mapping: the

initializer updates its mapping by learning from how the solver refines its initial solution, for the purpose of providing better initial solution for the solver in the next iteration; 2) Learn-objective: the solver updates its objective function by shifting its high value region from the refined solution to the observed solution, for the sake of matching the refined solution to the observed one in terms of value in the next iteration.

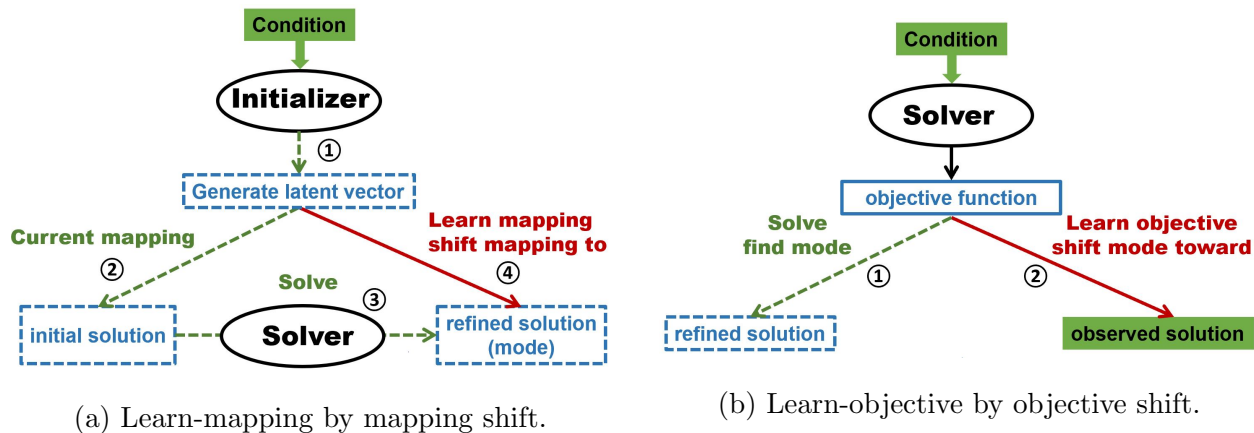


Figure 2.2: Learning step of cooperative training.

Fig. 2.2a illustrates Learn-mapping step. In the Initialization step, the initializer generates the latent noise vector, which, together with the input condition, is mapped to the initial solution. In the Learn-mapping step, the initializer updates its parameters so that it maps the input condition and the latent vector to the refined solution, in order to absorb the refinement made by the solver. Because the latent vector is known, it does not need to be inferred and the learning is easy. In other words, keeping the same mapping source, the initializer shifts its mapping target from the initial solution toward the refined solution.

Fig. 2.2b illustrates Learn-objective step. In the Solve step, the solver finds the refined solution at high value region around a mode of the objective function. In the Learn-objective step, the solver updates its parameters so that the objective function shifts its high value region around the mode toward the observed solution, so that in the next iteration, the refined solution will get closer to the observed solution.

The solver shifts its mode toward the observed solution, while inducing the initializer maps the input condition and the latent vector to its mode. Learning an initializer is like mimicking “how”, while learning a solver is like trying to understand “why” in terms of goal or value underlying the action.

Why slow thinking solver? The reason we need a solver in addition to an initializer is that it is often easier to learn the objective function than learning to generate the solution directly, since it is always easier to demand or desire something than to actually produce something directly. Because of its relative simplicity, the learned objective function can be more generalizable than the learned initializer. For instance, in an unfamiliar situation, we tend to be tentative, relying on slow thinking planning rather than fast thinking habit.

Efficiency. Even though we use the wording “slow thinking”, it is only relative to “fast thinking”. In fact, the slow thinking solver is usually fast enough, especially if it is jumpstarted by fast thinking initializer, and there is no problem scaling up our method to big datasets. Therefore the time efficiency of the slow thinking method is not a concern.

Student-teacher v.s. actor-critic. We may consider the initializer as a student model, and the solver as a teacher model. The teacher refines the initial solution of the student by a refinement process, and distills the refinement process into the student. This is different from the actor-critic relationship in (inverse) reinforcement learning [AN04, ZMB08, HE16] because the critic does not refine the actor’s solution by a slow thinking process.

Cooperative learning v.s. adversarial learning. Our framework, belonging to cooperative learning [XLG18a, XLG18b], jointly learns a conditional EBM as the slow thinking solver and a conditional generator as the fast thinking initializer. This is essentially different from the conditional generative adversarial net (cGAN) [GPM14, IZZ17, MO14], where a conditional discriminator is simultaneously learned to help train the conditional generator. Our framework simultaneously trains both models and keeps both of them after training, while cGAN discards its discriminator once the generator model is well trained. In other words, our framework trains both the slow thinking solver (*i.e.*, the EBM) and the fast think-

ing initializer (*i.e.*, the generator), while cGAN only desires a fast thinking model (*i.e.*, the generator). Thus, the advantage of our method over cGAN is that our method is equipped with a refinement process guided by the learned EBM.

We apply our learning method to various conditional learning tasks, such as class-to-image generation, image-to-image translation, image inpainting, etc. Our experiments show that the proposed method is effective compared to other methods, such as those based on GANs [GPM14].

Amortized computation and temporal difference learning. The solver is an iterative computing process. The initializer is an amortization of this process. The learning of the initializer can be considered temporal difference learning, where the finite steps of refinements produce the temporal difference to be distilled into the initializer.

Learning from external and internal data. The learning of the conditional energy function is from the training data, which we may call the external data. The learning of the initializer can be considered as learning from the internal data produced by the computational process of the solver.

Policy, value, and control. The initializer is similar to a policy network. The solver is similar to an iterative optimal control or planning process based on a value network. The conditional energy function is similar to a cost function.

Vector-valued initializer and scalar-valued conditional energy function. The initializer learns a mapping from an input to a high-dimensional output. The solver learns a scalar-valued conditional energy function. It is much easier to learn a scalar-valued function than a high-dimensional vector-valued mapping, so that the iterative refinement process guided by the learned energy function improves the initializer.

Contributions. This paper proposes a novel method for supervised learning of high-dimensional conditional distributions by learning a fast thinking initializer and a slow thinking solver. We show the effectiveness of our method on conditional image generation and

recovery tasks. Perhaps more importantly,

- We propose a different method for conditional learning than GAN-based methods. Unlike GANs, our method has a learned value function (i.e., the energy function in the conditional EBM) to guide a slow thinking process to refine the solution of the initializer (i.e., conditional generator). We demonstrate the benefit of such a refinement on various image synthesis tasks.
- The proposed framework is generic and can be applied to a broad range of artificial intelligence problems that can be modeled via a conditional learning framework, e.g., inverse optimal control, etc. The interaction between the fast thinking initializer and the slow thinking solver can be of interest to cognitive science.

2.2 Related work

The following themes are closely related to our research. We will briefly review each of them and connect them with our work.

Conditional Adversarial Learning Generative Adversarial Networks (GANs) [GPM14] proposed by Goodfellow et al. have demonstrated promising results of image generation in [RMC16], which belongs to unconditional learning, in which no supervision signals are used. With the success of adversarial learning, the conditional version of GAN (*i.e.*, conditional GAN or cGAN) [RAY16] has become a popular framework for supervised conditional learning, and it has been successfully applied to different scenarios that can be modeled in the context of conditional learning. For example, [MO14, DCF15] use conditional GANs for image synthesis based on class labels. [RAY16, ZXL17] study text-conditioned image synthesis. Other examples include image-to-image translation [IZZ17], semantic-image-to-photo translation [WLZ18a], super-resolution [LTH17], and video-to-video synthesis [WLZ18b], etc. Our work studies similar problems. The major difference between the conditional GAN and our

method is that ours is based on a conditional energy function that serves as an objective function and an iterative algorithm, which is the Langevin dynamics guided by this objective function. This iterative process corresponds to slow thinking. Existing adversarial learning methods do not involve this slow thinking refinement process.

Cooperative Learning Just as the conditional GAN is inspired by the original GAN [GPM14], our learning method is inspired by the recent work of generative cooperative networks (CoopNets) [XLG18a, XLG18b], where the models are unconditioned. Specifically, the CoopNets framework consists of an unconditional EBM and an unconditional latent variable model, and jointly trains both models via MCMC teaching [XLG18a], where the latent variable model learns to initialize the MCMC sampling of the EBM. While unconditioned generation is interesting, conditional generation and recovery is much more useful in applications. It is also much more challenging because we need to incorporate the input condition into both the initializer and the solver. Thus our method is a substantial generalization of the CoopNets [XLG18a], and our extensive experiments convincingly demonstrate the usefulness of our method, which in the meantime provides a different methodology from GAN-based methods. Our work is the first to study conditional cooperative learning, and propose the fast thinking and slow thinking framework as a conditional version of CoopNets.

Multi-modal Generative Learning Learning joint probability distribution of signals of different modalities enables us to recover or generate one modality based on other modalities. For example, [XYH05] learns a dual-wing harmoniums model for image and text data. [NKK11] learns stacked multimodal auto-encoder on video and audio data. [SS12] learns a multimodal deep Boltzmann machine for joint image and text modeling. Our work focuses on the conditional distribution of one modality given another modality, and our method involves the cooperation between two types of models.

Energy-based Generative ConvNets Our slow thinking solver is related to energy-based generative ConvNets [XLZ16, GLZ18, XZW17, XZW19, XZG18, NHZ19, NHH20], which are EBMs with energy functions parameterized by deep neural nets, and trained by

MCMC-based maximum likelihood learning. [XLZ16] is the first to learn EBMs parametrized by modern ConvNets by maximum likelihood estimation via Langevin dynamics, and also investigates ReLU [KSH12] with Gaussian reference in the proposed model that is called generative ConvNet. [GLZ18] proposes a multi-grid sampling and learning method for training generative ConvNets. The spatial-temporal generative ConvNet proposed in [XZW17, XZW19] further generalizes the generative ConvNet of images in [XLZ16] to modeling dynamic patterns, *e.g.*, videos or image sequences, by parameterizing the energy function with a bottom-up spatial-temporal ConvNet. [XZG18, XZG20] develops a volumetric version of the energy-based generative neural net, which is called generative VoxelNet, for 3D object patterns. Recently, [NHZ19] investigates training the energy-based generative ConvNet with a short-run MCMC. All models mentioned above are unconditioned EBMs, while our solver is a conditioned EBM jointly trained with a conditional latent variable model serving as an approximate conditional sampler.

Unsupervised Conditional Learning Some methods study unsupervised conditional learning, where the inputs and outputs are unpaired in the training set. For example, CycleGAN [ZPI17] jointly trains two GANs and enforces a cycle-consistency regularization between them to learn a two-way translator between two image collections in the absence of paired examples. AlignFlow [GCS20] adopts normalizing flow models [DKB14, DSB17] to solve this problem. Recently, CycleCoopNets [XZF21] tackles the unpaired translation problem based on the framework of cooperative learning. Our work belongs to supervised conditional learning, where the correspondence between source domain and target domain is given and used as supervision during training.

2.3 Cooperative conditional learning

Let Y be the D -dimensional output signal of the target domain, and C be the input condition of the source domain. Our goal is to learn the conditional distribution $p(Y|C)$ of the target

signal (solution) Y given the source signal C (problem) as the condition. We shall learn $p(Y|C)$ from the training dataset of the pairs $\{(Y_i, C_i), i = 1, \dots, n\}$ with the fast thinking initializer and slow thinking solver.

2.3.1 Slow thinking solver

The solver is based an objective function or value function $f(Y, C; \theta)$ defined on (Y, C) . $f(Y, C; \theta)$ can be parametrized by a bottom-up convolutional network (ConvNet) where θ collects all the weight and bias parameters. Serving as a negative energy function, $f(Y, C; \theta)$ defines a joint EBM [XLZ16]:

$$p(Y, C; \theta) = \frac{1}{Z(\theta)} \exp[f(Y, C; \theta)], \quad (2.1)$$

where $Z(\theta) = \int \exp[f(Y, C; \theta)] dY dC$ is the normalizing constant.

Fixing the source signal C , $f(Y, C; \theta)$ defines the value of the solution Y for the problem defined by C , and $-f(Y, C; \theta)$ defines the conditional energy function. The conditional probability is given by

$$\begin{aligned} p(Y|C; \theta) &= \frac{p(Y, C; \theta)}{p(C; \theta)} = \frac{p(Y, C; \theta)}{\int p(Y, C; \theta) dY} \\ &= \frac{1}{Z(C, \theta)} \exp[f(Y, C; \theta)], \end{aligned} \quad (2.2)$$

where $Z(C, \theta) = Z(\theta)p(C; \theta)$. The learning of this model seeks to maximize the conditional log-likelihood function

$$\mathcal{L}(\theta) = \frac{1}{n} \sum_{i=1}^n \log p(Y_i|C_i; \theta), \quad (2.3)$$

whose gradient $\mathcal{L}'(\theta)$ is

$$\sum_{i=1}^n \left\{ \frac{\partial}{\partial \theta} f(Y_i, C_i; \theta) - \mathbb{E}_{p(Y|C_i, \theta)} \left[\frac{\partial}{\partial \theta} f(Y, C_i; \theta) \right] \right\}, \quad (2.4)$$

where $\mathbb{E}_{p(Y|C; \theta)}$ denotes the expectation with respect to $p(Y|C, \theta)$. The identity underlying Eq. (2.4) is $\frac{\partial}{\partial \theta} \log Z(C, \theta) = \mathbb{E}_{p(Y|C, \theta)} \left[\frac{\partial}{\partial \theta} f(Y, C; \theta) \right]$.

The expectation in Eq. (2.4) is analytically intractable and can be approximated by drawing samples from $p(Y|C, \theta)$ and computing the Monte Carlo average. This can be solved by an iterative algorithm, which is a slow thinking process. One solver is the Langevin dynamics for sampling $Y \sim p(Y|C, \theta)$, which iterates the following step:

$$Y_{\tau+1} = Y_{\tau} + \frac{\delta^2}{2} \frac{\partial}{\partial Y} f(Y_{\tau}, C; \theta) + \delta U_{\tau}, \quad (2.5)$$

where τ indexes the time steps of the Langevin dynamics, δ is the step size, and $U_{\tau} \sim \mathcal{N}(0, I_D)$ is Gaussian white noise, D is the dimensionality of Y . A Metropolis-Hastings acceptance-rejection step can be added to correct for finite δ . The Langevin dynamics is gradient descent on the energy function, plus noise for diffusion so that it samples the distribution instead of being trapped in the local modes.

For each observed condition C_i , we run the Langevin dynamics according to Eq. (2.5) to obtain the corresponding synthesized example \tilde{Y}_i as a sample from $p(Y|C_i, \theta)$. The Monte Carlo approximation to $\mathcal{L}'(\theta)$ is

$$\mathcal{L}'(\theta) \approx \frac{\partial}{\partial \theta} \left[\frac{1}{n} \sum_{i=1}^n f(Y_i, C_i; \theta) - \frac{1}{n} \sum_{i=1}^n f(\tilde{Y}_i, C_i; \theta) \right]. \quad (2.6)$$

We can then update $\theta^{(t+1)} = \theta^{(t)} + \gamma_t \mathcal{L}'(\theta^{(t)})$.

Objective shift. The above gradient ascent algorithm is to increase the average value of the observed solutions versus that of the refined solutions, *i.e.*, on average, it shifts high value region or mode of $f(Y, C_i; \theta)$ from the generated solution \tilde{Y}_i toward the observed solution Y_i . The convergence of such a stochastic gradient ascent algorithm has been studied by [You99].

2.3.2 Fast thinking initializer

The initializer is of the following form:

$$X \sim \mathcal{N}(0, I_d), Y = g(X, C; \alpha) + \epsilon, \epsilon \sim \mathcal{N}(0, \sigma^2 I_D), \quad (2.7)$$

where X is the d -dimensional latent noise vector, and $g(X, C; \alpha)$ is a top-down ConvNet parameterized by α . The ConvNet g maps the observed condition C and the latent noise

vector X to the signal Y directly. Given high-dimensional C as the source signal, we can model g by an encoder-decoder structure: we first encode C into a latent vector Z , and then we map (X, Z) to Y by a decoder. Then, we can generate Y from the conditional generator model by a direct sampling, *i.e.*, first sampling X from its prior distribution, and then mapping (X, Z) into Y directly. We treat this as a fast thinking without iteration.

We can learn the initializer from the training pairs $\{(Y_i, C_i), i = 1, \dots, n\}$ by maximizing the conditional log-likelihood

$$\mathcal{L}(\alpha) = \frac{1}{n} \sum_{i=1}^n \log p(Y_i | C_i, \alpha), \quad (2.8)$$

where $p(Y|C, \alpha) = \int p(X)p(Y|C, X, \alpha)dX$. The learning algorithm iterates the following two steps: (1) sample X_i from $p(X_i|Y_i, C_i, \alpha)$ by Langevin dynamics; (2) update α by gradient descent on $\frac{1}{n} \sum_{i=1}^n \|Y_i - g(X_i, C_i; \alpha)\|^2$. See Algorithm 1 for details.

2.3.3 Cooperative training of initializer and solver

The initializer and the solver can be trained jointly as follows.

1. The initializer supplies initial samples for the MCMC of the solver. For each observed condition input C_i , we first generate $\hat{X}_i \sim \mathcal{N}(0, I_d)$, and then generate the initial solution $\hat{Y}_i = g(\hat{X}_i, C_i; \alpha) + \epsilon_i$. If the current initializer is close to the current solver, then the generated $\{\hat{Y}_i, i = 1, \dots, n\}$ should be a good initialization for the solver to sample from $p(Y|C_i, \theta)$, *i.e.*, starting from the initial solutions $\{\hat{Y}_i, i = 1, \dots, n\}$, we run Langevin dynamics for l steps to get the refined solutions $\{\tilde{Y}_i, i = 1, \dots, n\}$. These $\{\tilde{Y}_i\}$ serve as the synthesized examples from $p(Y|C_i)$ and are used to update θ in the same way as we learn the solver model in Eq. (2.6) for objective shifting.
2. The initializer then learns from the MCMC. Specifically, the initializer treats $\{(\tilde{Y}_i, C_i), i = 1, \dots, n\}$ produced by the MCMC as the training data. The key is that these $\{\tilde{Y}_i\}$ are obtained by the Langevin dynamics initialized from the $\{\hat{Y}_i, i = 1, \dots, n\}$, which are

generated by the initializer with *known* latent noise vectors $\{\hat{X}_i, i = 1, \dots, n\}$. Given $\{(\hat{X}_i, \tilde{Y}_i, C_i), i = 1, \dots, n\}$, we can learn α by minimizing $\frac{1}{n} \sum_{i=1}^n \|\tilde{Y}_i - g(\hat{X}_i, C_i; \alpha)\|^2$, which is a nonlinear regression of \tilde{Y}_i on (\hat{X}_i, C_i) . This can be accomplished by gradient descent

$$\Delta\alpha \propto -(\tilde{Y}_i - g(\hat{X}_i, C_i; \alpha)) \frac{\partial}{\partial \alpha} g(\hat{X}_i, C_i; \alpha). \quad (2.9)$$

Mapping shift: Initially $g(X, C; \alpha)$ maps (\hat{X}_i, C_i) to the initial solution \hat{Y}_i . After updating α , $g(X, C; \alpha)$ maps (\hat{X}_i, C_i) to the refined solution \tilde{Y}_i . Thus the updating of α absorbs the MCMC transitions that change \hat{Y}_i to \tilde{Y}_i . In other words, we distill the MCMC transitions of the refinement process into $g(X, C; \alpha)$.

Algorithm 1 presents a description of the conditional learning with two models. See Figs. 2.1 and 2.2 for illustrations. Both computations can be carried out by back-propagation, and the whole algorithm is in the form of alternating back-propagation.

In Algorithm 1, the conditional EBM is the primary model for conditional synthesis or recovery by MCMC sampling. The conditional generator model plays an assisting role to initialize the MCMC sampling.

2.4 Theoretical underpinning

This section presents theoretical underpinnings of the model and the learning algorithms presented in the previous section. Readers who are more interested in applications and experiments can jump to the next section.

2.4.1 Kullback-Leibler divergence

The Kullback-Leibler divergence between two distributions $p(x)$ and $q(x)$ is defined as $\text{KL}(p\|q) = \mathbb{E}_p[\log(p(X)/q(X))]$.

The Kullback-Leibler divergence between two conditional distributions $p(y|x)$ and $q(y|x)$

Algorithm 1 Cooperative Conditional Learning

Input: (1) training examples $\{(Y_i, C_i), i = 1, \dots, n\}$, (2) numbers of Langevin steps l .

Output: (1) learned parameters θ and α , (2) generated examples $\{\hat{Y}_i, \tilde{Y}_i, i = 1, \dots, n\}$.

1: $t \leftarrow 0$, initialize θ and α .

2: **while** *not converged* **do**

3: **Initialization by mapping:** For $i = 1, \dots, n$, generate $\hat{X}_i \sim \mathcal{N}(0, I_d)$, and generate the initial solution $\hat{Y}_i = g(\hat{X}_i, C_i; \alpha^{(t)}) + \epsilon_i$.

4: **Solve based on objective:** For $i = 1, \dots, n$, starting from \hat{Y}_i , run l steps of Langevin dynamics to obtain the refined solution \tilde{Y}_i , where each step follows Eq. (2.5).

5: **Learn-objective by objective shift:** Update $\theta^{(t+1)} = \theta^{(t)} + \gamma_t \mathcal{L}'(\theta^{(t)})$, where $\mathcal{L}'(\theta^{(t)})$ is computed according to Eq. (2.6).

6: **Learn-mapping by mapping shift:** Update $\alpha^{(t+1)} = \alpha^{(t)} + \gamma_t \Delta \alpha^{(t)}$, where $\Delta \alpha^{(t)}$ is computed according to Eq. (2.9).

7: Let $t \leftarrow t + 1$

8: **end while**

is defined as

$$\begin{aligned} \text{KL}(p\|q) &= \mathbb{E}_p \left[\log \frac{p(Y|X)}{q(Y|X)} \right] \\ &= \int \log \frac{p(y|x)}{q(y|x)} p(x, y) dx dy, \end{aligned} \quad (2.10)$$

where the expectation is over the joint distribution $p(x, y) = p(x)p(y|x)$.

2.4.2 Slow thinking solver

The slow thinking solver model is

$$\begin{aligned} p(Y|C; \theta) &= \frac{p(Y, C; \theta)}{p(C; \theta)} = \frac{p(Y, C; \theta)}{\int p(Y, C; \theta) dY} \\ &= \frac{1}{Z(C; \theta)} \exp[f(Y, C; \theta)], \end{aligned} \quad (2.11)$$

where

$$Z(C; \theta) = \int \exp [f(Y, C; \theta)] dY \quad (2.12)$$

is the normalizing constant and is analytically intractable.

Suppose the training examples $\{(Y_i, C_i), i = 1, \dots, n\}$ are generated by the true joint distribution $f(Y, C)$, whose conditional distribution is $f(Y|C)$. For large sample $n \rightarrow \infty$, the maximum likelihood estimation (MLE) of θ is to minimize the Kullback-Leibler divergence

$$\min_{\theta} \text{KL}(f(Y|C) \| p(Y|C; \theta)). \quad (2.13)$$

In practice, the expectation with respect to $f(Y, C)$ is approximated by the sample average. The difficulty with $\text{KL}(f(Y|C) \| p(Y|C; \theta))$ is that the $\log Z(C; \theta)$ term is analytically intractable, and its derivative has to be approximated by MCMC sampling from the model $p(Y|C; \theta)$.

2.4.3 Fast thinking initializer

The fast thinking initializer is

$$X \sim \mathcal{N}(0, I_d), Y = g(X, C; \alpha) + \epsilon, \epsilon \sim \mathcal{N}(0, \sigma^2 I_D). \quad (2.14)$$

We use the notation $q(Y|C; \alpha)$ to denote the resulting conditional distribution, which can be obtained by

$$q(Y|C; \alpha) = \int q(X)q(Y|X, C; \alpha)dX, \quad (2.15)$$

and is analytically intractable.

For large sample, the maximum likelihood estimation of α is to minimize the Kullback-Leibler divergence

$$\min_{\alpha} \text{KL}(f(Y|C) \| q(Y|C; \alpha)). \quad (2.16)$$

Again, the expectation with respect to $f(Y, C)$ is approximated by the sample average. The difficulty with $\text{KL}(f(Y|C) \| q(Y|C; \alpha))$ is that $\log q(Y|C; \alpha)$ is analytically intractable, and its derivative has to be approximated by MCMC sampling of the posterior $q(X|Y, C; \alpha)$.

2.4.4 Objective shift: modified contrastive divergence

Let $M(Y_1|Y_0, C; \theta)$ be the transition kernel of the finite-step MCMC that refines the initial solution Y_0 to the refined solution Y_1 . Let $(M_{\theta}q)(Y_1|C; \alpha) = \int M(Y_1|Y_0, C; \theta)q(Y_0|C; \alpha)dY_0$ be the distribution obtained by running the finite-step MCMC from $q(Y_0|C; \alpha)$.

Given the current initializer $q(Y|C; \alpha)$, the objective shift updates θ_t to θ_{t+1} , and the update approximately follows the gradient of the following modified contrastive divergence [Hin02, XLG18a]

$$\text{KL}(f(Y|C)\|p(Y|C; \theta)) - \text{KL}((M_{\theta_t}q)(Y|C; \alpha)\|p(Y|C; \theta)). \quad (2.17)$$

Compare Eq. (2.17) with the MLE Eq. (2.11), Eq. (2.17) has the second divergence term $\text{KL}((M_{\theta_t}q)(Y|C; \alpha)\|p(Y|C; \theta))$ to cancel the $\log Z(C; \theta)$ term, so that its derivative is analytically tractable. The learning is to shift $p(Y|C; \theta)$ or its high value region around the mode from the refined solution provided by $(M_{\theta_t}q)(Y|C; \alpha)$ toward the observed solution given by $f(Y|C)$. If $(M_{\theta_t}q)(Y|C; \alpha)$ is close to $p(Y|C; \theta)$, then the second divergence is close to zero, and the learning is close to MLE update.

2.4.5 Mapping shift: distilling MCMC

Given the current solver model $p(Y|C; \theta)$, the mapping shift updates α_t to α_{t+1} , and the update approximately follows the gradient of

$$\text{KL}((M_{\theta}q)(Y|C; \alpha_t)\|q(Y|C; \alpha)). \quad (2.18)$$

This update distills the MCMC transition M_{θ} into the model $q(Y|C; \alpha)$. In the idealized case where the above divergence can be minimized to zero, then $q(Y|C; \alpha_{t+1}) = (M_{\theta}q)(Y|C; \alpha_t)$. The limiting distribution of the MCMC transition M_{θ} is $p(Y|C; \theta)$, thus the cumulative effect of the above update is to lead $q(Y|C; \alpha)$ close to $p(Y|C; \theta)$.

Compare Sec. 2.4.5 to the MLE Eq. (2.14), the training data distribution becomes $(M_{\theta}q)(Y|C; \alpha_t)$ instead of $f(Y|C)$. That is, $q(Y|C; \alpha)$ learns from how M_{θ} refines it. The

learning is accomplished by mapping shift where the generated latent vector X is known, thus does not need to be inferred (or the Langevin inference algorithm can initialize from the generated X). In contrast, if we are to learn from $f(Y|C)$, we need to infer the unknown X by sampling from the posterior distribution.

In the limit, if the algorithm converges to a fixed point, then the resulting $q(Y|C; \alpha)$ minimizes $\text{KL}((M_\theta q)(Y|C; \alpha) \| q(Y|C; \alpha))$, that is, $q(Y|C; \alpha)$ seeks to be the stationary distribution of the MCMC transition M_θ , which is $p(Y|C; \theta)$.

If the learned $q(Y|C; \alpha)$ is close to $p(Y|C; \theta)$, $(M_{\theta_t} q)(Y|C; \alpha)$ is even closer to $p(Y|C; \theta)$. Then the learned $p(Y|C; \theta)$ is close to MLE because the second divergence term in Eq. (2.17) is close to zero.

2.5 Experiments

We test the proposed framework for conditional learning on a variety of vision tasks. According to the form of the conditional learning, we organize the experiments into two parts. In the first part (Experiment 1), we study conditional learning for a mapping from category (*i.e.*, one-hot vector) to image, *e.g.*, image generation conditioned on image class, while in the second part (Experiment 2), we study conditional learning for a mapping from image to image, *e.g.*, image-to-image translation. We propose a specific network architecture of our model in each experiment due to the different forms of input-output domains. Unlike the unconditioned cooperative learning framework [XLG18a, XLG18b], the conditioned framework needs to find a proper way to fuse the condition input C into both the bottom-up ConvNet f in the solver and the top-down ConvNet g in the initializer, for the sake of capturing accurate conditioning information. An improper design can cause not only unrealistic but also condition-mismatched synthesized results.

2.5.1 Experiment 1: Category \rightarrow Image

2.5.1.1 Network architecture

We start from learning the conditional distribution of an image given a category or class label. The category information is encoded as a one-hot vector. The network architectures of the models in this experiment are given as follows.

In the initializer, we can concatenate the one-hot vector C with the latent noise vector X sampled from $\mathcal{N}(0, I_d)$ as the input of the decoder $\Psi([X, C])$ to build a conditional generator $g(X, C; \alpha)$. The generator maps the input into image Y by several layers of deconvolutions. We call this setting “early concatenation”. See Fig. 2.3a(1) for an illustration. We can also adopt an architecture with “late concatenation”, where the concatenation happens in the intermediate layer of the initializer. Specifically, we can first sample the latent noise vector X from Gaussian noise prior $\mathcal{N}(0, I_d)$, and then decode X to an intermediate result with spatial dimension $b \times b$ by a decoder $\Psi_1(X)$. The decoder consists of several layers of deconvolutions, each of which is followed by batch normalization [IS15] and ReLU non-linear transformation. We then replicate the one-hot vector C spatially and perform a channel concatenation with the intermediate output. After that, we generate the target image Y from the concatenated result $[\Psi_1(X), C]$ by another decoder $\Psi_2([\Psi_1(X), C])$ that consists of several deconvolution layers. Batch normalization and ReLU layer are used between two consecutive deconvolution layers, and tanh non-linearity is added at the bottom layer. $g(X, C; \alpha)$ is the composition of Ψ_1 and Ψ_2 . See Fig. 2.3a(2) for an illustration.

To build the value function for the solver model, in the setting of “early concatenation”, we first replicate the condition one-hot vector C spatially and perform a depth concatenation with image Y , and then map them to a scalar by an encoder, $\Phi([Y, C])$, that consists of several layers of convolutions and ReLU non-linear transformations. The value function $f(Y, C; \theta)$ is designed as $\Phi([Y, C]) - \|Y\|^2/2s^2$. This corresponds to an exponential tilting

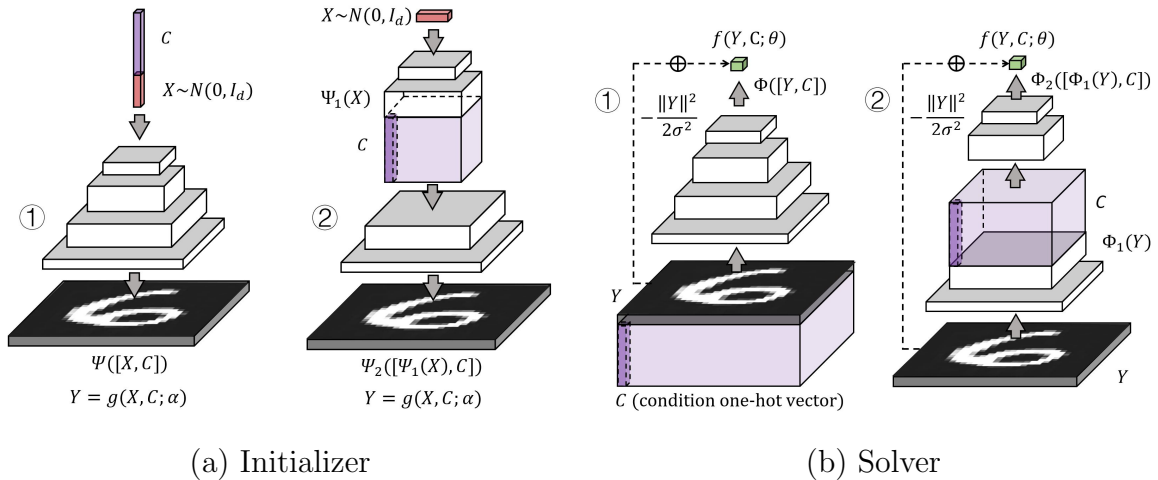


Figure 2.3: Network architecture of category-to-image.

form in [XLZ16],

$$p(Y, C; \theta) = \frac{1}{Z(\theta)} \exp [\Phi(Y, C; \theta)] p_0(Y), \quad (2.19)$$

where $p_0(Y)$ is Gaussian white noise distribution, *i.e.*, $p_0(Y) \propto \exp(-\|Y\|^2/2s^2)$, and s is a hyper-parameter for the standard deviation of p_0 . See Fig. 2.3b(1) for an illustration. As to the “late concatenation”, we first encode the image Y to an intermediate result with spatial dimension $a \times a$ by an encoder $\Phi_1(Y)$, which consists of several layers of convolutions and ReLU non-linear transformations, and then we replicate the one-hot vector C spatially and perform a depth concatenation with the intermediate result. The value function is defined by another encoder $\Phi_2([\Phi_1(Y), C])$ plus $-\|Y\|^2/2s^2$, in which the encoder takes as input the concatenated result $[\Phi_1(Y), C]$ and outputs a scalar by performing several layers of convolutions and ReLU non-linear transformations. See Fig. 2.3b(2) for an illustration.

2.5.1.2 Conditional image generation on grayscale images

We first test our model on two grayscale image datasets, such as MNIST [LBB98] and fashion-MNIST [XRV17]. The former is a dataset of handwritten digit images, and the latter is a dataset of fashion product images. Each of them consists of 70,000 28×28 images,

each of which is associated with a label from 10 classes. In each dataset, 60,000 examples are used for training and the rest are for testing. We learn our model on each of them respectively, conditioned on their class labels that are encoded as one-hot vectors. Since these two datasets are similar in number of classes, image size, data size, and image format (*i.e.*, grayscale), we use the same model for them.

We adopt the setting of “early concatenation” introduced in Sec. 2.5.1.1 for the initializer. To be specific, $g(X, C; \alpha)$ is a generator that maps the $1 \times 1 \times 138$ concatenated result (Note that the dimension of X is 128, and the size of C is 10.) to a 28×28 grayscale image by 4 layers of deconvolutions with kernel sizes $\{4, 4, 4, 4\}$, up-sampling factors $\{1, 2, 2, 2\}$ and numbers of output channels $\{256, 128, 64, 1\}$ at different layers. The last deconvolution layer is followed by a tanh operation, and each of the others is followed by batch normalization and ReLU operation.

We adopt the setting of “late concatenation” introduced in Sec. 2.5.1.1 for the solver. Specifically, $\Phi_1(Y)$ consists of 2 layers of convolutions with filter sizes $\{5, 3\}$, down-sampling factors $\{2, 2\}$ and numbers of output channels $\{64, 128\}$. The concatenated output is of size $7 \times 7 \times 138$. (Note that the number of the output channels of Φ_1 is 128, and the size of C is 10.) $\Phi_2([\Phi_1(Y), C])$ is a 2-layer ConvNet, where the first layer has 256 3×3 filters, and the last layer is a fully-connected layer with 100 filters.

We use Adam [KB15] to optimize the solver with initial learning rate 0.0008, $\beta_1 = 0.5$ and $\beta_2 = 0.999$, and the initializer with initial learning rate 0.0001, $\beta_1 = 0.5$ and $\beta_2 = 0.999$. The mini-batch size is 300. The number of paralleled MCMC chains is 300. The number of Langevin dynamics steps is $l = 16$. The step size δ of Langevin dynamics is 0.0008. The standard deviation of the residual in the initializer is $\sigma = 0.3$, and the standard deviation s of the reference distribution p_0 in the solver is 0.016. We run 1,600 epochs to train the model, where we disable the noise term in Langevin dynamics after the first 100 epochs.

Fig. 2.4 shows some of the generated samples conditioned on the class labels after training on the MNIST dataset and fashion-MNIST dataset. Each column is conditioned on one label

and each row is a different generated sample. The qualitative results show that our method can learn realistic conditional models.

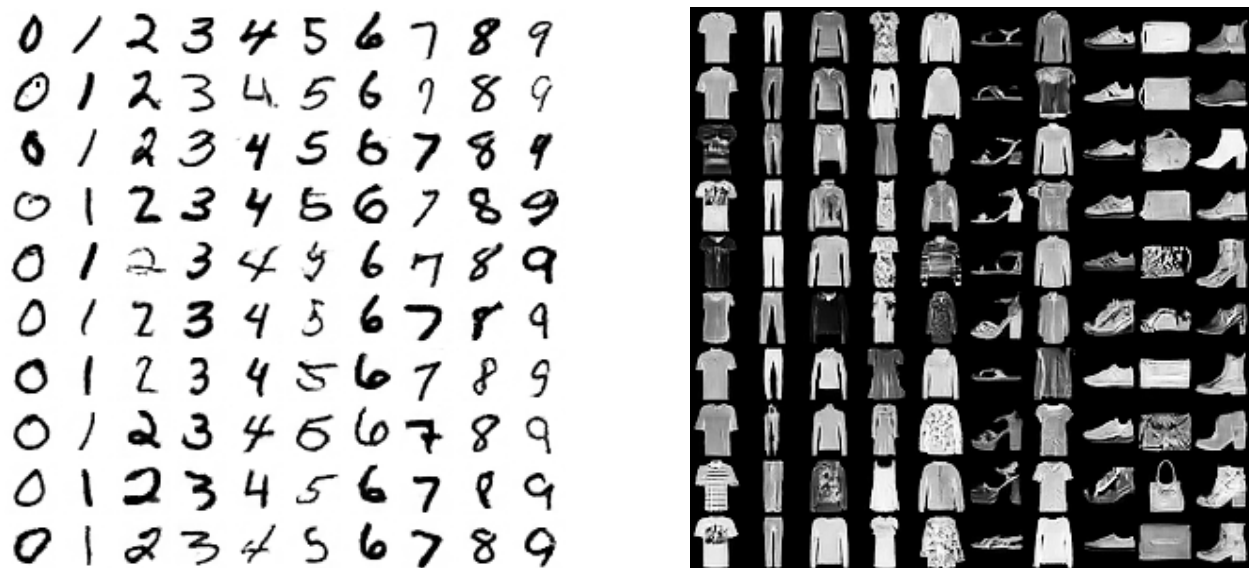


Figure 2.4: Generated MNIST handwritten digits (left) and fashion MNIST images (right).

To quantitatively evaluate the learned conditional distribution, we use FID [HRU17] score as a metric to measure the dissimilarity between the distributions of the observed and the synthesized examples. Specifically, we compute the distance between feature vectors extracted from observed and synthesized examples by a pre-trained Inception model [SVI16], with the following formula

$$\text{FID} = \|\tilde{\mu} - \mu\|^2 + \text{Tr} \left(\tilde{\Sigma} + \Sigma - 2(\tilde{\Sigma}\Sigma)^{1/2} \right),$$

where $V \sim \mathcal{N}(\mu, \Sigma)$ and $\tilde{V} \sim \mathcal{N}(\tilde{\mu}, \tilde{\Sigma})$ are the output feature vectors from Inception model of the observed and synthesized examples, respectively. We can fit a multi-variate Gaussian to feature vectors $\{V_i\}$ and $\{\tilde{V}_i\}$ to obtain means $\mu, \tilde{\mu}$ and variances $\Sigma, \tilde{\Sigma}$ for the observed and synthesized distributions respectively. A lower FID score implies better qualities of the synthesized images.

To compute FID score, we sample 10,000 examples from the learned conditional distribution by first sampling the class label C from the uniform prior distribution, and X from

$\mathcal{N}(0, I_d)$, then the initializer and the solver model cooperatively generate the synthesized example from the sampled C and X . Tab. 2.1 shows a comparison of FID scores of different methods on two datasets. Our method achieves better results than other conditional and unconditional baseline methods in terms of generation quality evaluated by FID. Those baselines include GAN-based, flow-based, and variational inference methods.

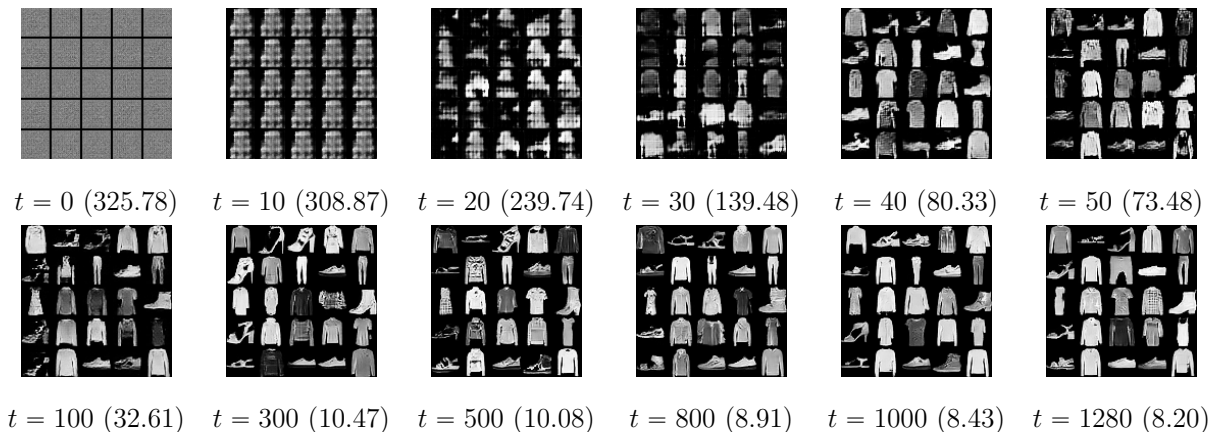


Figure 2.5: Image generation by the models at different training epochs.

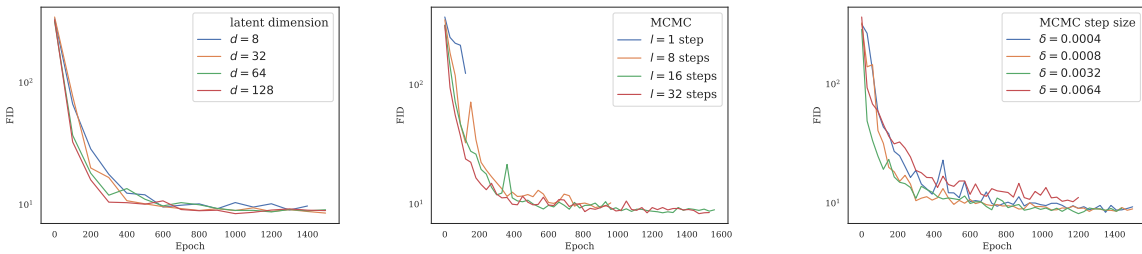
Fig. 2.5 displays some examples of the synthesized images at different training epochs along with the corresponding FID scores. The images shown are generated by the solver. The images at the same position of 5×5 image matrix of different training epochs share the same condition C , *i.e.*, the class label. We can find that as the cooperative training progresses, the synthesized images become more and more realistic and the FID scores become lower and lower. Additionally, the learned connection between the condition (*i.e.*, class label) and the target (*i.e.*, image) becomes more and more accurate in the sense that when the model converges, even though the appearances of the synthesized images vary at different epochs, they are always consistent with their input conditions.

We study the influences of different choices of some hyper-parameters, such as the number of dimension d of the latent space X in the initializer, the number of Langevin refinement steps l , and the step size δ of each Langevin. Fig. 2.6 depicts the influences of varying d , l and δ , respectively, while training on fashion-MNIST dataset. Each curve represents the testing

Table 2.1: The Fréchet Inception Distance (FID) scores of different models trained on MNIST and fashion-MNIST dataset.

	Model	MNIST	fashion-MNIST
unconditional	GLO [BJL17]	49.60	57.70
	VAE [KW14]	21.85	69.84
	BEGAN [BSM17]	13.54	15.90
	EBGAN [ZML17]	11.10	41.32
	GLANN [HLM19]	8.60	13.10
	WGAN [ACB17]	7.07	28.17
	LSGAN [MLX17]	6.75	14.72
	DCGAN [RMC16]	4.54	8.22
	InfoGAN [CDH16]	28.09	-
	GLF [XYA19]	5.80	10.30
conditional	CGlow [LLG19]	29.64	-
	CAGlow [LLG19]	26.34	-
	VCGAN [HZH19]	-	13.8
	CVAE-GAN [BCW17]	-	15.9
	CVAE [SLY15]	20.00	36.64
	ACGAN [OOS17]	12.55	49.11
	CGAN [MO14]	5.91	11.92
	CCoopNets (ours)	4.50	8.20

FID scores over training epochs. We observe that (1) the quality of synthesis decreases with decreasing d . (2) the more the number of MCMC refinement steps, the stabler the learning process, and the more time-consuming the refinement process of the solver. With a small l , e.g., 1 or 8, the cooperative learning tends to fail easily at the early stage of training because the slow-thinking solver distills an insufficient refinement process to the initializer such that the latter can not provide good initial solutions for the former. Fig. 2.6b shows



(a) number of latent dimension (b) number of Langevin steps (c) step size of Langevin refinement

Figure 2.6: Model analysis on fashion-MNIST dataset.

that the learning curves for $l = 1$ (in blue) and $l = 8$ (in orange) are terminated early due to failures occurred during training. Tab. 2.2 shows a comparison of computational time per epoch with different numbers of Langevin steps l and different numbers of latent dimensions d . The running times were recorded in a PC with an Intel i7-6700k CPU and a Titan Xp GPU. A choice of $l = 16$ or 32 appears reasonable. The influence of d on the computational time is not significant. (3) A large Langevin step size allows the model to learn faster to generate high quality images, at the cost of arriving on a sub-optimal synthesis of images. A smaller Langevin step size may allow the model to generate more realistic images but it may take more Langevin steps.

Table 2.2: Comparison of computational time (in seconds) per epoch on fashion-MNIST dataset.

	$l = 1$	$l = 8$	$l = 16$	$l = 32$	$l = 64$
$d = 8$	8.98	20.38	26.88	46.74	86.93
$d = 32$	9.23	20.21	27.04	46.95	86.95
$d = 64$	9.12	20.10	27.55	47.22	87.06
$d = 128$	9.37	20.50	27.76	48.62	86.92

2.5.1.3 Conditional image generation on Cifar-10

We also test the proposed framework on Cifar-10 [Kri09] object dataset, which contains 10-class 60,000 training images of 32×32 pixels. Compared with the MNIST dataset, Cifar-10 contains training images with more complicated visual patterns.

As to the initializer, we adopt the “late concatenation” setting. Specifically, $\Psi_1(X)$ is a decoder that maps 100-dimensional X (*i.e.*, $1 \times 1 \times 100$) to an intermediate output with spatial dimension 8×8 by 2 layers of deconvolutions with kernel sizes $\{4, 5\}$, up-sampling factors $\{1, 2\}$ and numbers of output channels $\{256, 128\}$ at different layers from top to bottom, respectively. The condition C is a 10-dimensional one-hot vector to represent the class. $\Psi_2([\Psi_1(X), C])$ is a generator that maps the $8 \times 8 \times 138$ concatenated result to a $32 \times 32 \times 3$ image by 2 layers of deconvolutions with kernel sizes $\{5, 5\}$, up-sampling factors $\{2, 2\}$ and numbers of output channels $\{64, 3\}$ at different layers.

We adopt the “late concatenation” setting for the solver. $\Phi_1(Y)$ consists of 2 layers of convolutions with filter sizes $\{5, 3\}$, down-sampling factors $\{2, 2\}$ and numbers of output channels $\{64, 128\}$. The concatenated output is of size $8 \times 8 \times 138$. $\Phi_2([\Phi_1(Y), C])$ is a 2-layer bottom-up ConvNet, where the first layer has 256 3×3 filters, and the last layer is a fully connected layer with 100 filters.

We use the Adam for optimization. The initial learning rates for the solver and initializer are 0.002 and 0.0064, respectively. The joint models are trained with mini-batches of size 300. The number of paralleled MCMC chains is also 300. The number of Langevin dynamics steps is 8. The step size δ of Langevin dynamics is 0.0008. We run 2,000 epochs to train the model, where we disable the noise term in Langevin dynamics in the last 1,500 ones.

Fig. 2.7 shows the generated object patterns. Each row is conditioned on one category. The first two columns display some typical training examples, while the rest columns show generated images conditioned on labels. We evaluate the learned conditional distribution by computing the inception scores of the generated examples. Tab. 2.3 compares our framework

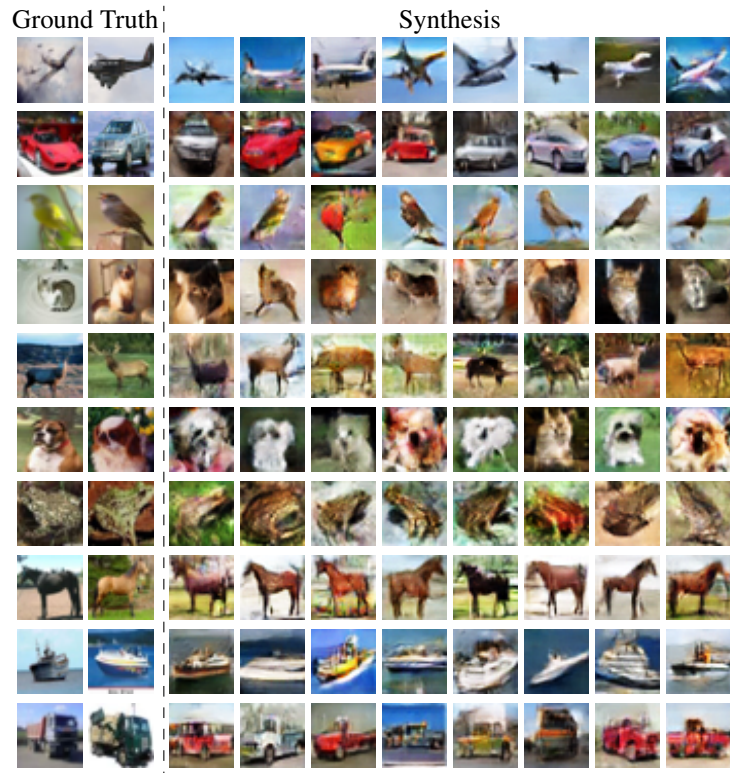


Figure 2.7: Generated Cifar-10 object images.

against two baselines, which are two conditional models based on GANs. The proposed model performs better than the baselines. We also found that in the proposed method, the solution provided by the initializer is indeed further refined by the solver in terms of inception score.

2.5.1.4 Disentangling style and category

To test the inference power of the fast-thinking initializer, which is trained jointly with the slow-thinking solver, we apply the learned initializer to a task of style transfer from an unseen testing image in one category onto other categories. The network architectures of initializer and solver are similar to those used in Sec. 2.5.1.2, except that the training images in this experiment are RGB images and they are of size 32×32 pixels. With the learned initializer, we first infer the latent variables X corresponding to that testing image. We then fix the inferred latent vector, change the category label C , and generate the different categories of

Table 2.3: Inception scores of different models trained on Cifar-10 dataset.

	Model	Inception score
unconditional	PixelCNN [OKK16]	4.60
	PixelIQN [ODM18]	5.29
	DCGAN [RMC16]	6.40
	WGAN-GP [GAA17]	6.50
	ALI [DBP17]	5.34
conditional	CGAN [SGZ16]	6.58
	Conditional SteinGAN [WL16]	6.35
	Initializer (ours)	6.63
	Solver (ours)	7.30

images with the same style as the testing image by the learned model. Given a testing image Y with known category label C , the inference of the latent vector X can be performed by directly sampling from the posterior distribution $p(X|Y, C; \alpha)$ via Langevin dynamics, which iterates

$$X_{\tau+1} = X_{\tau} + sU_{\tau} + \frac{s^2}{2} \left[\frac{1}{\sigma^2} (Y - g(X_{\tau}, C; \alpha)) \frac{\partial}{\partial X} g(X_{\tau}, C; \alpha) - X_{\tau} \right]. \quad (2.20)$$

If the category label of the testing image is unknown, we need to infer both C and X from Y . Since C is a one-hot vector, in order to adopt a gradient-based method to infer C , we adopt a continuous approximation by reparametrizing C using a softMax transformation on the auxiliary continuous variables A . Specifically, let $C = (c_k, k = 1, \dots, K)$ and $A = (a_k, k = 1, \dots, K)$, we reparametrize $C = v(A)$ where $c_k = \exp(a_k) / \sum_k \exp(a'_k)$, for $k = 1, \dots, K$, and assume the prior for A to be $N(0, I_K)$. Then the Langevin dynamics for sampling $A \sim p(A|Y, X)$ iterates

$$A_{\tau+1} = A_{\tau} + sU_{\tau} + \frac{s^2}{2} \left[\frac{1}{\sigma^2} (Y - g(X_{\tau}, v(A); \alpha)) \frac{\partial}{\partial A} g(X, v(A_{\tau}); \alpha) - A \right]. \quad (2.21)$$

Fig. 2.8 shows 10 results of style transfer on SVHN dataset [NWC11]. For each testing



Figure 2.8: Style transfer on SVHN dataset.

image Y , we infer X and C by sampling $[X, C] \sim p(X, C|Y)$, which iterates (1) $X \sim p(X|Y, C)$, and (2) $C = v(A)$ where $A \sim p(A|Y, X)$, with randomly initialized X and C . We then fix the inferred latent vector X , change the category label C , and generate images from the combination of C and X by the learned initializer. This experiment demonstrates the effectiveness of our model in style and category disentanglement.

2.5.2 Experiment 2: Image \rightarrow Image

2.5.2.1 Network architecture

We study learning conditional distributions for image-to-image translation by our framework. The network architectures of the models in this experiment are discussed as follows.

As to the initializer, a straightforward design is presented below: we first sample X from the Gaussian noise prior $N(0, I_d)$, and we encode the condition image C via an encoder

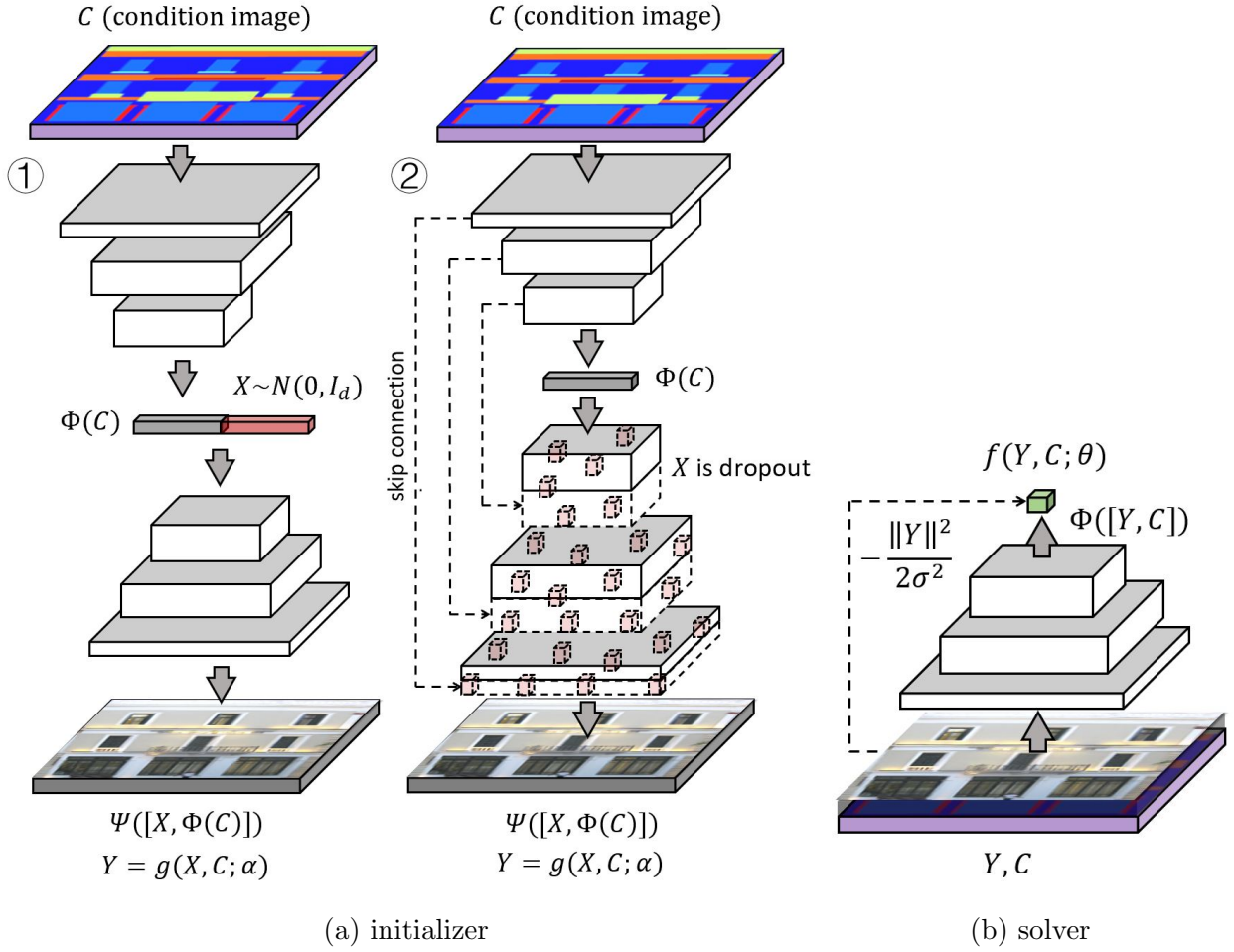


Figure 2.9: Network architecture (image-to-image translation).

$\Phi(C)$. The image embedding $\Phi(C)$ is then concatenated to the latent noise vector X . After this, we generate target image Y by a decoder $\Psi([X, \Phi(C)])$. The initializer $g(X, C; \alpha)$ is the composition of Φ and Ψ . With Gaussian noise X , the initializer will produce stochastic outputs as a distribution. See Fig. 2.9a(1) for an illustration of the structure. However, in the initial experiments, we found that this design was ineffective in the sense that the generator learned to ignore the noise and produce deterministic outputs. Inspired by [IZZ17], we design the initializer by following a general shape of the U-Net [RFB15] with the form of dropout [SHK14], applied on several layers, as noise that accounts for stochasticity in this experiment. A U-Net is an encoder-decoder structure with skip connections added between

each layer j and layer $M - j$, where M is the number of layers. Each skip connection performs a concatenation between all channels at layer j and those at layer $M - j$. In the task of image-to-image translation, the input and output images usually differ in appearance but share low-level information. For example, in the case of translating sketch image to photo image, the input and output images are roughly aligned in outline except that they have different colors and textures in appearance. The addition of skip connections allow a direct transfer of low-level information across the network. Fig. 2.9a(2) illustrates the U-Net structure with dropout as the initializer for image-to-image translation.

As to the design of the solver model, we first perform channel concatenation on target image Y and condition image C , where both images are of the same size. The value function $f(Y, C, \theta)$ is then defined by an encoder $\Phi([Y, C])$ plus $-\|Y\|^2/2s^2$, in which $\Phi([Y, C])$ maps the 6-channel “image” to a scalar by several convolutional layers. Leaky ReLU layers are added between two consecutive convolutional layers. Fig. 2.9b shows an illustration of the network architecture of the solver.

2.5.2.2 Semantic labels \rightarrow Scene images

The experiments are conducted on CMP Facade dataset [TS13] where each building facade image is associated with an image of architectural labels. The condition image and the target image are of the size of 256×256 pixels with RGB channels. Data are randomly split into training and testing sets.

In the initializer, the encoder Φ consists of 8 layers of convolutions with a filter size 4, a subsampling factor 2, and the numbers of channels $\{64, 128, 256, 512, 512, 512, 512, 512\}$ at different layers. Batch normalization and leaky ReLU (with slope 0.2) layers are used after each convolutional layer except that batch normalization is not applied after the first layer. The output of Φ is then fed into Ψ , which consists of 8 layers of deconvolutions with a kernel size 4, an up-sampling factor 2, and the numbers of channels $\{512, 512, 512, 512, 256, 128, 64, 3\}$ at different layers. Batch normalization, dropout with a dropout rate of 0.5, and ReLU layer

are added between two consecutive deconvolutional layers, and a tanh non-linearity is used after the last layer. The U-Net structure used in this experiment is a connection of the encoder Φ and the decoder Ψ , along with skip connections added to concatenate activations of each layer j and layer $M - j$. (M is the total number of layers.) Therefore, the numbers of output channels of Ψ in the U-Net are $\{1024, 1024, 1024, 1024, 512, 256, 128, 3\}$. The dropout that is applied to each layer of Ψ implies an implicit latent vector X in the initializer. Such an implicit X is too complicated to infer. However, there is no need to infer this X with the cooperative training, which can get around the difficulty of the inference of any complicated forms of latent factors by MCMC teaching. In other words, in each iteration, the learning of the initializer $\Psi([X, \Phi(C)])$ is based on how the MCMC changes the initial examples generated by the initializer from the condition image C and the randomness X due to dropout.

In the solver model, we first perform a channel concatenation on target image Y and condition image C , where both images are of size $256 \times 256 \times 3$. The value function is then defined by a 4-layer encoder $\Phi([Y, C])$, which maps a 6-channel “image” to a scalar as the value score by 3 convolutional layers with numbers of channels $\{64, 128, 256\}$, filter sizes $\{5, 3, 3\}$ and subsampling factors $\{2, 2, 1\}$ at different layers (from bottom to top), and one fully connected layer with 100 single filters. Leaky ReLU layer is used between two consecutive convolutional layers.

Adam is used to optimize the solver with an initial learning rate 0.007, and the initializer with an initial learning rate 0.0001. We set the mini-batch size to be 1. The number of paralleled MCMC chains is also 1. We run 15 Langevin steps with a step size $\delta = 0.002$. The standard deviation of the residual in the initializer is $\sigma = 0.3$. The standard deviation of the reference distribution in the solver is $s = 0.016$. We run 3,000 epochs to train our model.

We adopt random jitter and mirroring for data augmentation in the training stage. As to random jitter, we first resize the input images from 256×256 to 286×286 , and then

randomly crop image patches with a size 256×256 .

In this task, we found it beneficial to feed both the refined solutions and the observed ground truth solutions to the initializer, when we update the initializer at each iteration. The solver’s job remains unchanged, but the initializer is tasked to not only learn from the solver $\{\tilde{Y}_i\}$ but also to be near the ground truth solutions $\{Y_i\}$. We add an extra ℓ_1 loss to penalize the distance between the output of the initializer and the ground truth solution. [IZZ17] also finds this strategy effective in training a GAN-based conditional model for image-to-image translation.

As to the computational time, compared with GAN-based method, our framework has additional $l = 15$ steps of Langevin. However, the Langevin is based on gradient, whose computation can be powered by back-propagation, so it is not significantly time-consuming. To be concrete, our method costs 32.7s, while GAN-based method costs 30.9s per epoch for training in a PC with an Intel i7-6700k CPU and a Titan Xp GPU in this experiment.

Fig. 2.10 shows some qualitative results of generating building facade images from the semantic labels. The first column displays 5 semantic label images that are unseen in the training data. The second column displays the corresponding ground truth images for reference. The results by a baseline method, pix2pix [IZZ17], are shown in the third row for comparison. pix2pix is a conditional GAN method for image-to-image mapping. Since its generator also uses a U-Net and is paired up with a ℓ_1 loss, for a fair comparison, our initializer adopts exactly the same U-Net structure as in [IZZ17]. The fourth to sixth columns are results generated by some variants of the conditional GAN method, including cVAE-GAN [ZZP17], cVAE-GAN++ [ZZP17] and BicycleGAN [ZZP17]. The seventh and eighth rows show the generated results conditioned on the semantic label images shown in the first row by the learned initializer and solver, respectively. We can easily observe qualitative improvements by comparing the outputs of the solver with the ones of the initializer.

We perform human perceptual tests for evaluating the visual quality of synthesized images. We randomly select 30 different human users to participate in these tests. We compare



Figure 2.10: Generating images conditioned on architectural labels.

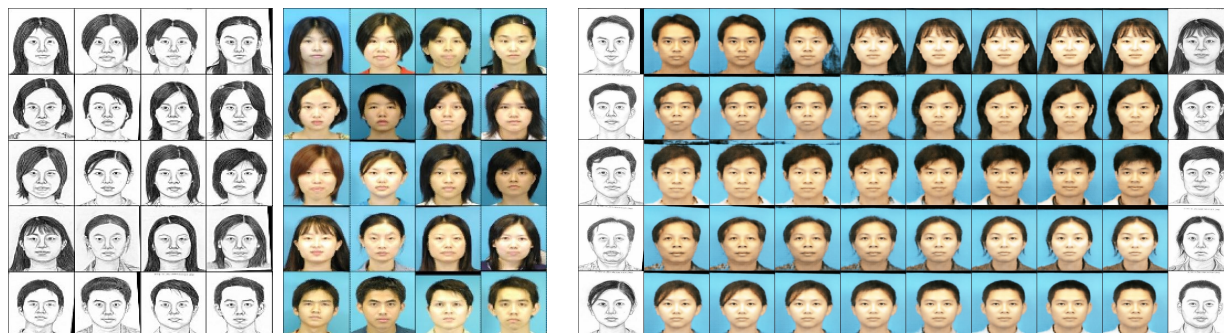
two methods in each test, where each participant is first presented two images at a time, which are results generated by two different methods given the same conditional input, and then asked which one looks more like a real image. We have total 36 pairwise comparisons in each test for each participant. We evaluate each method by the ratio that the images generated by the method are preferred. As shown in Tab. 2.4, the results generated by our method are considered more realistic by the human subjects.

2.5.2.3 Sketch images \rightarrow Photo images

We next test the model on CUHK Face Sketch database (CUFS) [WT09], where for each face, there is a sketch image drawn by an artist based on a photo of the face. We learn to recover the color face images from the sketch images by the proposed framework. The

Table 2.4: Human perceptual tests for image-to-image synthesis.

Methods	Preference Ratio
CCoopNets (ours) / cVAE-GAN [ZZP17]	0.625 / 0.375
CCoopNets (ours) / cVAE-GAN++ [ZZP17]	0.687 / 0.313
CCoopNets (ours) / BicycleGAN [ZZP17]	0.628 / 0.372
CCoopNets (ours) / pix2pixel [IZZ17]	0.720 / 0.280



(a) Recover faces from sketches

(b) Sketch interpolation

Figure 2.11: Sketch-to-photo face synthesis on CUHK dataset.

network design and hyperparameter setting are similar to the one we used in Sec. 2.5.2.2, except that the mini-batch size and the number of paralleled MCMC chains are set to be 4.

Fig. 2.11a displays the face synthesis results conditioned on the sketch images. Columns 1 through 4 show some sketch images as input conditions, while columns 5 through 8 show the corresponding recovered images obtained by sampling from the learned conditional distribution. From the results, we can see that the generated facial appearance (color and texture) in each output image is not only reasonable but also consistent with the input sketch face image in the sense that the face identity in each sketch image remains unchanged after being translating to a photo image.

Fig. 2.11b demonstrates the learned manifold of sketch images (condition) by showing 5 examples of interpolation. For each row, the sketch images at the two ends are first



Figure 2.12: Results on edges \rightarrow shoes generation, compared to ground truth.

encoded into the embedding by $\Phi(C)$, and then each face image in the middle is obtained by first interpolating the sketch embedding, and then generating the images using the initializer with a fixed dropout, and eventually refining the results by the solver via finite-step Langevin dynamics. Even though there is no ground truth sketch images for the intervening points, the generated faces appear plausible. Since the dropout X is fixed, the only changing factor is the sketch embedding. We observe smooth changing of the generated faces.

We conduct another experiment on UT Zappos50K dataset [TS13] for photo image recovery from edge image. The dataset contains 50k training images of shoes. Edge images are computed by HED edge detector [XT15] with post processing. We use the same model structure as the one in the last experiment. Fig. 2.12 shows some qualitative results of synthesizing shoe images from edge images.

2.5.2.4 Image Inpainting

Table 2.5: Comparison with the baseline methods for image inpainting on the CMP Facade dataset and Paris streetview dataset.

Model	CMP Facades		Paris streetview	
	PSNR	SSIM	PSNR	SSIM
cVAE-GAN [ZZP17]	19.43	0.68	16.12	0.72
cVAE-GAN++ [ZZP17]	19.14	0.64	16.03	0.69
BicycleGAN [ZZP17]	19.07	0.64	16.00	0.68
pix2pix [IZZ17]	19.34	0.74	15.17	0.75
CCoopNets (ours)	20.47	0.77	21.17	0.79

Table 2.6: Comparison of model complexity with the baseline methods for image inpainting on CMP Facade dataset.

Model	Size	Time
	# of parameters	sec / epoch
cVAE-GAN [ZZP17]	60.85M	12.06
cVAE-GAN++ [ZZP17]	64.30M	18.40
BicycleGAN [ZZP17]	64.30M	25.60
pix2pix [IZZ17]	57.89M	12.62
CCoopNets (ours)	55.84M	22.43

We also test our method on the task of image inpainting by learning a mapping from an occluded image (256×256 pixels), where a mask with the size of 128×128 pixels is centrally placed onto the original version, to the original image. We use Paris streetview [PKD16] and the CMP Facade dataset. In this case, C is the observed part of the input image, and Y is the unobserved part of the image. The network architectures for both initializer and solver,

along with hyperparameter setting, are similar to those we used in Sec. 2.5.2.2. To recover the occluded part of the input images, we only update the pixels of the occluded region in the Langevin dynamics.

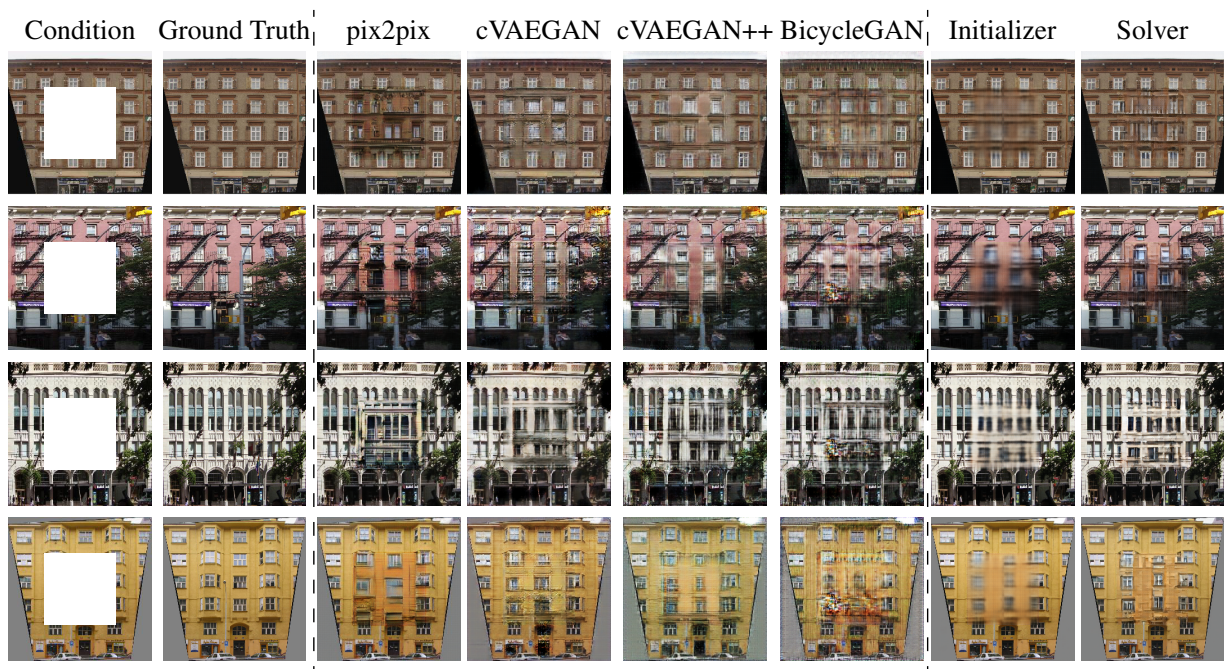


Figure 2.13: Results of photo inpainting on CMP Facade dataset.

We compare our method with some baselines, including pix2pix, cVAE-GAN, cVAE-GAN++ and BicycleGAN. Tab. 2.5 shows quantitative results where the recovery performance is measured by the peak signal-to-noise ratio (PSNR) and structural similarity measures (SSIM), which are computed between the occlusion regions of the generated example and the ground truth example. The batch size is one. Our method outperforms the baseline methods using adversarial training or variational inference in this recovery task. Tab. 2.6 reports a comparison of model complexity with the baseline methods on CMP Facade dataset in terms of number of model parameters and running time.

Fig. 2.13 shows a comparison of qualitative results of different methods on CMP Facade dataset. Each row displays one example. The first image is the testing image with a hole that needs to be recovered. The second image is the ground truth image. The third to

sixth images are the inpainting results obtained by pix2pix, cVAE-GAN, cVAE-GAN++ and BicycleGAN, respectively. The seventh and the last images are the results recovered by the initializer and the solver, respectively.

2.6 Conclusion

Solving a challenging problem usually requires an iterative algorithm. This amounts to slow thinking. The iterative algorithm usually requires a good initialization to jumpstart it so that it can converge quickly. The initialization amounts to fast thinking. For instance, reasoning and planning usually require iterative search or optimization, which can be initialized by a learned computation in the form of a neural network. Thus integrating fast thinking initialization and slow thinking sampling or optimization is very compelling. This paper addresses the problem of high-dimensional conditional learning and proposes a cooperative learning method that couples a fast thinking initializer and a slow thinking solver. The initializer initializes the iterative optimization or sampling process of the solver, while the solver in return teaches the initializer by distilling its iterative algorithm into the initializer. We demonstrate the proposed method on a variety of image synthesis and recovery tasks. Compared to GAN-based method, such as conditional GANs, our method is equipped with an extra iterative sampling and optimization algorithm to refine the solution, guided by a learned objective function. This may prove to be a powerful method for solving challenging conditional learning problems.

Part II

Structure Learning

CHAPTER 3

Reasoning Visual Dialogue with Structural and Partial Observations

Visual dialogue has drawn increasing research interests at the intersection of computer vision and natural language processing. In such tasks, an image is given as context input, associated with a summarizing caption and a dialogue history of question-answer pairs. The goal is to answer questions posed in natural language about images [DKG17], or recover a follow-up question based on the dialogue history [JLS18]. Despite its significance to artificial intelligence and human-computer interaction, it poses a richer set of challenges (see an example in Fig. 3.1) – requiring representing/understanding a series of multi-modal entities, and reasoning the rich semantic relations/structures among them. An ideal inference algorithm should be able to find out the underlying semantic structure and give a reasonable answer based on this structure.

Fig. 3.1 shows an example of visual dialogue, where the left side is context image, middle is image caption and dialogue history and the right side is the underlying semantic dependencies between nodes in the dialogue (darker green links indicate higher dependencies). A main difference between Visual Question Answering (VQA) and Visual Dialogue is that dialogues have more complex semantic dependencies. An ideal inference algorithm should be able to find out the underlying semantic structure and give a reasonable answer based on this structure.

Previous studies have explored this task through embedding rich features from image representation learned from convolutional neural networks and language (*i.e.*, question-answer

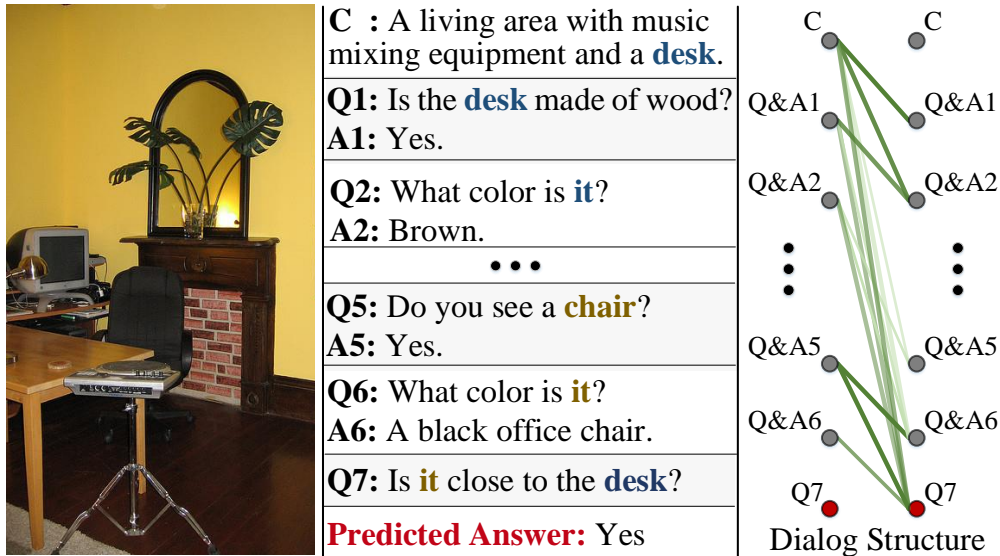


Figure 3.1: An illustration of the visual dialogue task.

pairs, caption) representations learned from recurrent sequential models. Their impressive results well demonstrate the importance of mining and fusing multi-modal information in this area. However, they largely neglect the key role of the rich relational information in dialogue. Although a few [ZWS18, WWS18a] leveraged co-attention mechanisms to capture cross-modal correlations, their reasoning ability is still quite limited. They typically concatenate the multi-modal features together and directly project the concatenated feature into the answer feature space by a neural network. On one hand, their reasoning process does not fully utilize the rich relational information in this task due to their monolithic vector representations of dialogue. On the other hand, their feed-forward network designs fail to deeply and iteratively mine and reason the information from different dialogue entities over the inherent dialogue structures.

In this work, we consider the problem of recovering the dialogue structure and reasoning about the question/answer simultaneously. We represent the dialogue as a graph, where the nodes are dialogue entities and the edges are semantic dependencies between nodes. Given the dialogue history as input, we have a partial observation of the graph. Then we formulate

the problem as inferring about the values of unobserved nodes (*e.g.*, the queried answer) and the graph structure.

The challenge of the problem is that there is no label for dialogue structures. For each individual dialogue, we need to recover the underlying structure in an unsupervised manner. The node values could then be inferred iteratively with the graph structure: we can reason about the nodes based on the graph structure, and further improve the structure based on the node values. To tackle this challenge, the insight is that a graph structure essentially specifies a joint probability distribution for all the nodes in the graph. Therefore we can view the queried dialogue entities as missing values in the data, the dialogue structure as unknown parameters of the distribution. Specifically, we encode the dialogue as a Markov Random Field (MRF) where some nodes are observed, and the goal is to infer the edge weights between nodes as well as the value of unobserved nodes. We formulate a solution based on the Expectation-Maximization (EM) algorithm, and provide a graph neural network (GNN) approach to approximate this inference.

Our model provides a unified framework which is applicable to diverse dialogue settings (detailed in Sec. 3.3). Besides, it provides extra post-hoc interpretability to show the dialogue structures through an implicit learning manner. We evaluate the performance of our method on VisDial v0.9 [DKG17], VisDial v1.0 [DKG17] and VisDial-Q [JLS18] datasets. The experimental results prove that our model is able to automatically parse the dialogue structure and infer reasonable answer, and further achieves promising performance.

3.1 Related Work

Image Captioning aims to annotate images with natural language at the scene level automatically, which has been a long-term active research area in computer vision community. Early work [OKB11, GWH14] typically tackled this task as a retrieval problem, *i.e.*, finding the best fitting caption from a set of predetermined caption templates. Modern

methods [MXY15, KF15, VTB15] were mainly based on a CNN-RNN framework, where the RNN leverages the CNN-representation of an input image to output a word sequence as the caption. In this way, they were freed from dependence of the pre-defined, expression-limited caption candidate pool. After that, some methods [XBK15, AHB18, LXP17] tried to integrate the vanilla CNN-RNN architecture with neural attention mechanisms, like semantic attention [LXP17], and bottom-up/top-down attention [AHB18], to name a few representative ones. Another popular trend [GGH17, PKK17, JKF16, CJS18, MXH18, LPC18, CZY18] in this area focuses on improving the discriminability of caption generations, such as stylized image captioning [GGH17, CZY18], personalized image captioning [PKK17], and context-aware image captioning [JKF16, CJS18].

Visual Question Answering focuses on providing a natural language answer given an image and a free-form, open-ended question. It is a more recent (dated back to [MF14, AAL15]) and challenging task (need to access information from both the question and image). With the availability of large-scale datasets [RKZ15, AAL15, GMZ15, GKS17, JHM17], numerous VQA models were proposed to build multimodal representations upon the CNN-RNN architecture [GMZ15, RKZ15], and recently extended with differentiable attentions [XBK15, LYB16, YHG16, ZGB16, AHB18, MDS18]. Rather than above classification-based VQA models, there were some other work [SSH16, JJM16, TAH18, BFZ18] leveraged answer representations into the VQA reasoning, *i.e.*, predicting whether or not an image-question-answer triplet is correct. Teney *et al.* [TLH17] proposed to solve VQA with graph-structured representations of both visual content and questions, showing the advantages of graph neural network in such structure-rich problems. Narasimhan *et al.* [NLS18] applied graph convolution networks for factual VQA. However, there are some notable differences between our model and [TLH17, NLS18] in the fundamental idea and theoretical basis, besides the specific tasks. First, we model the visual dialogue task as a problem of inference over a graph with partially observed data and unknown dialogue structures. This is one step further than propagating information over a fixed graph structure. Second, we emphasize both

graph structure inference (in a unsupervised manner) and unobserved node reasoning. Last, the proposed model provides an end-to-end network architecture to approximate the EM solution and offers a new glimpse into the visual dialogue task.

Visual dialogue refers to the task of answering a sequence of questions about an input image [DKG17, VSC17]. It is the latest vision-and-language problem, after the popularity of image captioning and visual question answering. It requires to reason about the image, the on-going question, as well as the past dialogue history. [DKG17] and [VSC17] represented two early attempts towards this direction, but with different dialogue settings. In [DKG17], a VisDial dataset is proposed and the questions in this dataset are free-form and may concern arbitrary content of the images. Differently, in [VSC17], a ‘Guess-What’ game is designed to identify a secret object through a series of yes/no questions. Following [DKG17], Lu *et al.* [LKY17] introduced a generator-discriminator architecture, where the generator are improved using a perceptual loss from the pre-trained discriminator. In [SLH17], a neural attention mechanism, called Attention Memory, is proposed to resolve the current reference in the dialogue. Das *et al.* [DKM17] then extended [DKG17] with an ‘image guessing’ game, *i.e.*, finding a desired image from a lineup of images through multi-round dialogue. Reinforcement Learning (RL) was used to tackle this task. Later methods to visual dialogue include applying Parallel Attention to discover the object through dialogue [ZWS18], learning a conditional variational auto-encoder for generating entire sequences of dialogue [MSD18], unifying visual question generation and visual question answering in a dual learning framework [JLS18], combining RL and Generative Adversarial Networks (GANs) to generate more human-like answers [WWS18a]. In [JLS18], a discriminative visual dialogue model was proposed and a new evaluation protocol was designed to test the questioner side of visual dialogue. More recently, [KMP18a] used a neural module network to solve the problem of visual coreference resolution.

Graph Neural Networks [GMS05, SGT09] draw a growing interest in the machine learning and computer vision communities, with the goal of combining structural representation

of graph/graphical models with neural networks. There are two main stream of approaches. One stream is to design neural network operations to directly operate on graph-structured data [DMI15, NAK16, MBM16, SK17, DBV16, KW17]. Another stream is to build graphically structured neural networks to approximate the learning/inference process of graphical models [LSR15, SSF16, LTB16, FXW18, BPL16, GSR17, WXS18, COW16]. Our method falls into this category. Some of these methods [LSR15, SSF16, BPL16, GSR17, KFW17, QWJ18] implement every graph node as a small neural network and formulate the interactions between nodes as a message propagation process, which is designed to be end-to-end trainable. Some others [ZJR15, LSR15, LLL15, LSV16, COW16] tried to integrate CRFs and neural networks in a fully differentiable framework, which is quite meaningful for semantic segmentation.

In this work, for the first time, we generalize the task of visual dialogue to such a setting that we have partial observation of the nodes (*i.e.*, image, caption and dialogue history), and the graph structure (relations in dialogue) needs to be automatically inferred. In this setting, the answer is the essentially unobserved node needs to be inferred based on the dialogue graph, where the graph structure describes the dependencies among given dialogue entities. We propose an essential neural network approach as an approximation to the EM solution of this problem. The proposed GNN is significantly different from most previous GNNs, which consider problems that the node features are observable, and usually a graph structure is given.

3.2 Our Approach

We begin by describing the visual dialogue task setup as introduced by Das *et al.* [DKG17]. Formally, a visual dialogue agent is given a dialogue tuple $D = \{(I, C, H_t, Q_t)\}$ as input, including an image I , a caption C , a dialogue history till round $t-1$, $H_t = \{(Q_k, A_k), k = 1, \dots, t-1\}$, and the current question Q_t being asked at round t . The visual dialogue agent

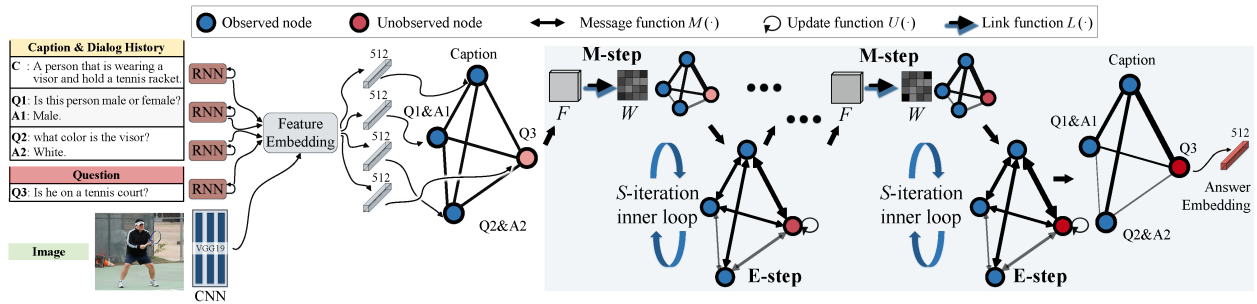


Figure 3.2: Illustration of GNN representation of visual dialogue.

is required to return a response A_t to the question Q_t , by ranking a list of 100 candidate answers.

In our approach, we represent the entire dialogue by a graph, and we solve for the optimal queried answer by a GNN as an approximate inference (see Fig. 3.2). In this graph, the dialogue entities $H_t = \{(Q_k, A_k), k = 1, \dots, t-1\}$, Q_t , and A_t are represented as nodes. The graph structure (*i.e.*, edges) represents the semantic dependencies between those nodes. The joint distribution of all the question and answer nodes are described by a Markov Random Field, where the values for some nodes are observed (*i.e.*, the history questions & answers, the current question). The node value for the current answer is unknown, and the model needs to infer its value as well as the graph structure encoded by the edge weights in this MRF.

The joint distribution in this MRF over all the nodes is specified by its potential functions and the graph structure. The potential functions can be learned in the training phase to maximize the likelihood of the training data, and used for inference in the testing phase. However, we cannot learn a fixed graph structure for all dialogues since they are different from dialogue to dialogue. For dialogues in both training and testing, we need to automatically infer the semantic structures.

In addition, because there is no label (also is hard to obtain) for such graph structures, our model needs to infer them in an unsupervised manner. Viewing the input nodes (*i.e.*,

the history questions & answers, the current question) as observed data, the queried answer node as missing data, we adopt an EM algorithm to recover both the distribution parameter (the edge weights) and the missing data (the current answer). In this algorithm, the edge weights and the queried answer node are inferred to maximize the expected log likelihood. Finally, we resemble this inference process by a GNN approach, in which the node values and edge weights are updated in an iterative fashion.

3.2.1 Dialogue as Markov Random Field

We model a dialogue as an MRF, in which the nodes represent questions and answers and the edges encode semantic dependencies. Specifically, in a fully connected MRF model, the joint probability of all the nodes \mathbf{v} is:

$$p(\mathbf{v}) = \frac{1}{Z} \exp \left\{ -\sum_i \phi_u(v_i) - \sum_{(i,j) \in E} \phi_p(v_i, v_j) \right\}, \quad (3.1)$$

where Z is a normalizing constant, $\phi_u(v_i)$ is the unary potential function, and $\phi_p(v_i, v_j)$ is the pairwise potential function.

In our task, we want to learn a general potential function for all dialogues. We also want to maintain soft relations between nodes (*i.e.*, a connectivity between 0 and 1) instead of just binary relations. Hence we generalize the above form to an MRF with $0 \sim 1$ weighed edges:

$$\begin{aligned} p(\mathbf{v}|W) &= \frac{1}{Z} \exp \left\{ -\sum_i w_i \phi_u(v_i) - \sum_{i,j} w_{i,j} \phi_p(v_i, v_j) \right\} \\ &= \frac{1}{Z} \exp \left\{ -\text{Tr}(W^T \Phi(\mathbf{v})) \right\}, \end{aligned} \quad (3.2)$$

where w_i and $w_{i,j}$ are the weights that compose the edge weight matrix W . Note that we write $\Phi(\mathbf{v})$ the potential matrix as a compact form of all the potentials between nodes, where $\Phi_{i,i} = \phi_u(v_i)$ and $\Phi_{i,j} = \phi_p(v_i, v_j)$.

3.2.2 Inference with Partial Observation

Next we briefly review EM as a typical approach to do inference with missing data. Suppose we have observed data \mathbf{x} and unobserved data \mathbf{z} , whose joint distribution is parametrized by θ . The goal is to infer the most likely parameter θ and random variable \mathbf{z} . The EM algorithm optimizes the expected log likelihood:

$$Q(\theta, \theta^{\text{old}}) = \int_{\mathbf{z}} p(\mathbf{z}|\mathbf{x}, \theta^{\text{old}}) \log p(\mathbf{x}, \mathbf{z}|\theta) dz. \quad (3.3)$$

An EM algorithm is an iterative process of two steps: expectation (E-step) and maximization (M-step). In the E-step, the above expected likelihood is computed. In the M-step, the parameter θ is optimized to maximize this objective:

$$\theta = \underset{\theta}{\operatorname{argmax}} Q(\theta, \theta^{\text{old}}). \quad (3.4)$$

The EM iteration always increases the observed data likelihood and terminates when a local minimum is found. However, the expected log likelihood Eq. (3.3) is often intractable. In the visual dialogue task, to compute this quantity we need to enumerate all possible answers to the current question in the entire language space. In practice, we can use an surrogate objective in the E-step, in which we compute the plug-in approximation [Vaa00] by a maximum a posteriori (MAP) estimate:

$$\tilde{Q}(\theta, \theta^{\text{old}}) = \max_{\mathbf{z}} p(\mathbf{z}|\mathbf{x}, \theta^{\text{old}}) \log p(\mathbf{x}, \mathbf{z}|\theta). \quad (3.5)$$

Then in the M-step we update the θ according to this surrogate objective.

3.2.3 MRF with Partial Observations

In the visual dialogue task, the question & answer history and the current question is given, hence we know the values for those nodes in the MRF. The task is to find out the missing value of the current answer node and the underlying semantic structure. Suppose in an MRF, we observe some nodes in the graph and we do not know the edge weights W . Denote

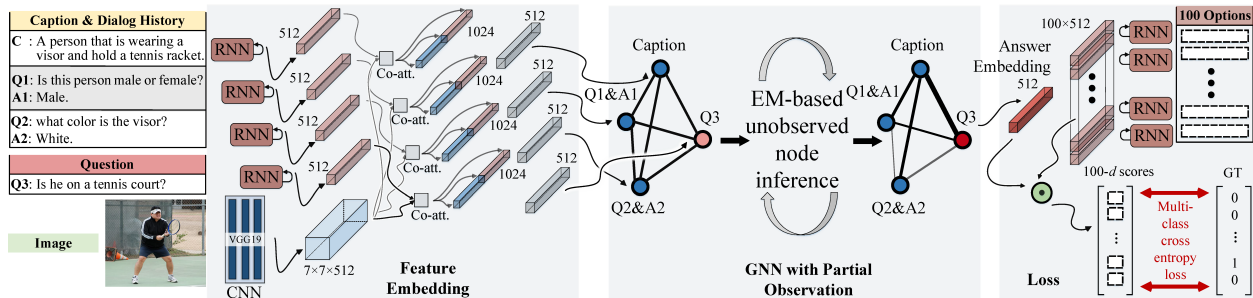


Figure 3.3: A detailed illustration of our model.

the observed nodes as \mathbf{x} and the unobserved nodes as \mathbf{z} , where $\mathbf{v} = \mathbf{x} \cup \mathbf{z}$ and $\mathbf{x} \cap \mathbf{z} = \emptyset$. Here the weight matrix W parametrizes the joint distribution of \mathbf{x} and \mathbf{z} , hence it can be viewed as the θ in the previous section. To jointly infer W the graph structure (*e.g.*, the semantic dependencies) and \mathbf{z} the missing values (*e.g.*, the queried answer), we run an EM algorithm:

E-step: We compute $\mathbf{z}^* = \operatorname{argmax}_{\mathbf{z}} p(\mathbf{z}|\mathbf{x}, W^{\text{old}})$ to obtain $\tilde{Q}(\theta, \theta^{\text{old}})$ in Eq. (3.5). This is achieved by a max-product loopy belief propagation [WF01]. At every iteration, each node sends a (different) message to each of its neighbors and receives a message from each neighbor. After receiving message from neighbors, the belief $b(v_i)$ for each node v_i is updated by the max-product update rule:

$$b(v_i) = \alpha \phi_u(v_i) \prod_{v_j \in \mathcal{N}(v_i)} m_{ji}(v_i), \quad (3.6)$$

where α is a normalizing constant, $\mathcal{N}(v_i)$ denotes the neighbor nodes of v_i , and $m_{ji}(v_i)$ is the message from v_j to v_i . The message is given by:

$$m_{ji}(v_i) = \max_{v_j} w_{ij} \phi_p(v_i, v_j) \prod_{v_k \in \mathcal{N}(v_j) \setminus v_i} m_{kj}(v_j). \quad (3.7)$$

where $\mathcal{N}(v_j) \setminus v_i$ indicates the all the neighboring nodes of v_j except v_i .

M-step: Based on the estimated \mathbf{z}^* in the E-step, we want to find the edge weights that

maximizes the objective Eq. (3.5):

$$\begin{aligned}
W &= \operatorname{argmax}_W \tilde{Q}(W, W^{\text{old}}) \\
&= \operatorname{argmax}_W p(\mathbf{z}^* | \mathbf{x}, W^{\text{old}}) \log p(\mathbf{x}, \mathbf{z}^* | W) \\
&= \operatorname{argmax}_W \log p(\mathbf{x}, \mathbf{z}^* | W).
\end{aligned} \tag{3.8}$$

The M-step together with E-step forms a coordinate descent algorithm in the objective function $\tilde{Q}(W, W^{\text{old}})$. This algorithm contains two loops: an outer loop of inferring \mathbf{z} and θ alternatively, and an inner loop of inferring the missing values \mathbf{z} by iterative belief propagation.

Note that in the partial observed case, for the E-step we fix the observed nodes $v_x \in \mathbf{x}$ and only update the unobserved nodes $v_z \in \mathbf{z}$. Hence we also only need to compute messages from observed nodes to unobserved nodes. The message passing and belief update process iterate until convergence. When the iteration terminates, we obtain an MAP estimate \mathbf{z}^* for the missing values, conditioned on the observed nodes \mathbf{x} and current estimated edge weights W .

3.2.4 GNN with Partial Observations

We design a GNN for the visual dialogue task guided by the above formulations. The network is structured as an MRF, in which the caption and each question/answer pair is represented as a node embedding, and the semantic relations are represented by edges. The model contains three different neural modules: message functions, update functions, and link functions. These modules are called iteratively to emulate the above EM algorithm.

E-step: We perform a neural message passing/belief propagation [GSR17] for an approximate inference of missing values \mathbf{z}^* . This process emulates the belief propagation in the E-step. For each node, we use an hidden state/embedding to represent its value. During belief propagation, the observed variables \mathbf{x} and the edge weights W are fixed. The hidden states of the unobserved nodes are iteratively updated by communicating with other nodes.

Specially, we use message functions $M(\cdot)$ to summarize messages to nodes coming from other nodes, and update functions $U(\cdot)$ to update the hidden node states according to the incoming messages.

At each iteration step s , the update function computes a new hidden state for a node after receiving incoming messages:

$$h_{v_i}^s = U(h_{v_i}^{s-1}, m_{v_i}^s), \quad (3.9)$$

where h_v^s is the hidden state/embedding for node v . m_v^s is the summarized incoming message for node v at s -th iteration. The messages are given by:

$$m_{v_i}^s = \sum_{v_j \in \mathcal{N}(v_i)} w_{ij} M(h_{v_i}^{s-1}, h_{v_j}^{s-1}). \quad (3.10)$$

The message passing phase runs for S iterations towards convergence. At the first iteration, the node hidden states h_v^0 are initialized by node features F_v .

M-step: Based on the updated hidden states of all the nodes in the E-step, we update the edge weights W by link functions. A link function $L(\cdot)$ estimates the connectivity w_{ij} between two nodes v_i and v_j based on their current hidden states:

$$w_{ij} = L(h_{v_i}, h_{v_j}). \quad (3.11)$$

3.2.5 Network Architecture

At each round of the dialogue, we aim to answer the query question based on the image, caption, and the question & answer (QA) history. For dialogue round t , we construct $t+1$ nodes in which one node represents the caption, $t-1$ nodes represents the history of $t-1$ rounds of QAs, and one last node represents the answer to the current query question. The embedding for each node is initialized by fusing the image feature and the language embedding of the corresponding sentence(s). As shown in Fig. 3.3, for the caption node we extract the language embedding of the caption, and fuse it with the image feature as an initialization. For the last node representing the queried answer, we use the corresponding

question embedding fused with the image feature to initialize the hidden state. For the rest nodes, the hidden states are initialized by fusing the QA embedding and the image feature. The fusing of language embeddings and image features are achieved by co-attention techniques [LYB16], and more details are introduced in Sec. 3.3. The goal of our approach is to infer the hidden state of the queried answer by the emulated EM algorithm.

After initializing the node hidden states with feature embeddings, we start the iterative inference by first estimating the edge weights. The edge weights are estimated by Eq. (3.11), where the link function is given by a dot product between transformed hidden states:

$$w_{ij} = L(h_{v_i}, h_{v_j}) = \langle \text{fc}(h_{v_i}), \text{fc}(h_{v_j}) \rangle \quad (3.12)$$

where $\langle \cdot, \cdot \rangle$ denotes dot product, and $\text{fc}(\cdot)$ denotes multiple fully connected layers with Rectified Linear Units (ReLU) in between the layers.

Using $M(h_{v_i}^{s-1}, h_{v_j}^{s-1}) = h_{v_j}^{s-1}$ as the message function, the summarized message from all neighbor nodes is computed as $m_{v_i}^s = \sum_{v_j \in \mathcal{N}(v_i)} w_{ij} h_{v_j}^{s-1}$. To stabilize the training of the update function, we normalize the sum of weights for edges coming into one node to 1 by a softmax function. Then the node hidden state is updated by a Gated Recurrent Unit (GRU) [CMG14]:

$$h_{v_i}^s = U(h_{v_i}^{s-1}, m_{v_i}^s) = \text{GRU}(h_{v_i}^{s-1}, m_{v_i}^s). \quad (3.13)$$

Here the GRU is chosen for two reasons. First, Eq. (3.9) has a natural recurrent form. GRU is one type of Recurrent Neural Networks (RNN) that known to be more computationally efficient than Long short-term memory (LSTM). Second, Li *et al.* [LTB16] has shown that GRU performs well in GNNs as update functions.

The algorithm stops after several iterations of the outer loop for EM, in which the edge weights W and the node hidden states h_v are updated alternatively. Inside each iteration, an inner loop is performed to update the node hidden states. The inner loop emulates the E-step, where a belief propagation is performed. The algorithm is illustrated in Algorithm 2. For the visual dialogue task, the set of unobserved nodes include only the node that represents

Algorithm 2 EM for Graph Neural Network

Input: Extracted features F_{v_x} for observed nodes $v_x \in \mathbf{x}$

Output: Graph structure W , node embeddings h_{v_z} for unobserved nodes $v_z \in \mathbf{z}$

```
1: /* Initialization */
2: for each observed node  $v_x \in \mathbf{x}$  do
3:   Initialize  $h_{v_x}$  to be  $F_{v_x}$ 
4: end for
5: for each unobserved node  $v_z \in \mathbf{z}$  do
6:   Initialize  $h_{v_z}$  to be the question embedding
7: end for
8: /* Expectation-Maximization: outer loop */
9: while not converged do
10:  /* M-step */
11:  for each node pair  $(v_i, v_j)$  do
12:     $w_{ij} = L(h_{v_i}, h_{v_j}) = \langle \text{fc}(h_{v_i}), \text{fc}(h_{v_j}) \rangle$ 
13:  end for
14:  /* E-step: inner loop for message passing */
15:  for step  $s$  from 1 to  $S$  do
16:    for each  $v_z \in \mathbf{z}$  do
17:      /* Compute incoming message for  $v_i$  */
18:       $m_{v_z}^s = \sum_{v_j \in \mathcal{N}(v_z)} w_{zj} h_{v_j}^{s-1}$ 
19:      /* Update embedding for unobserved  $v_i$  */
20:       $h_{v_z}^s = U(h_{v_z}^{s-1}, m_{v_z}^s) = \text{GRU}(h_{v_z}^{s-1}, m_{v_z}^s)$ 
21:    end for
22:  end for
23: end while
```

the current queried answer.

Finally, we regard the hidden state of the last node as the embedding of the queried

answer. To choose one answer from the pre-defined options provided by the dataset, we compute $\langle h_v, h_o \rangle$ where h_v is the node hidden state from the last node and h_o is the language embedding for an option. A softmax activation function is applied to those dot products, and a multi-class cross entropy loss is computed to train the GNN.

3.3 Experiments

3.3.1 Performance on VisDial v0.9

Dataset: We first evaluate the proposed approach on VisDial v0.9 [DKG17], which was collected via two Amazon Mechanical Turk (AMT) subjects chatting about an image. The first person is allowed to see only the image caption, and instructed to ask questions about the hidden image to better understand the scene. The second worker has access to both image and caption, and is asked to answer the first person’s questions. Both are encouraged to talk in a natural manner. Their conversation is ended after 10 rounds of question answering. VisDial v0.9 contains a total of 1,232,870 dialogue question-answer pairs on MSCOCO images [LMB14]. It is split into 80K for `train`, 3K for `val` and 40K as the `test`, in a manner consistent with [DKG17].

Evaluation Protocol: We follow [DKG17] to evaluate individual responses at each round ($t = 1, 2, \dots, 10$) in a retrieval setup. Specifically, at test time, every question is coupled with a list of 100 candidate answer options, which a VisDial model is asked to return a sorting of the candidate answers. The model is evaluated on standard retrieval metrics [DKG17]: Recall@1, Recall@5, Recall@10, Mean Reciprocal Rank (MRR), and Mean Rank of human response. Lower value for MR and higher values for all the other metrics are desirable.

Data Preparation: To pre-process the data, we first resize each image into 224×224 resolution, and use the output of the last pooling layer (*pool5*) of VGG-19 [SZ15] as the image feature ($512 \times 7 \times 7$). For the text data, *i.e.*, caption, questions and answers, we convert digits to words, and remove contractions, before tokenizing. The captions, questions, answers

Methods	MRR \uparrow	R@1 \uparrow	R@5 \uparrow	R@10 \uparrow	Mean \downarrow
LF [DKG17]	0.5807	43.82	74.68	84.07	5.78
HRE [DKG17]	0.5846	44.67	74.50	84.22	5.72
HREA [DKG17]	0.5868	44.82	74.81	84.36	5.66
MN [DKG17]	0.5965	45.55	76.22	85.37	5.46
SAN-QI [YHG16]	0.5764	43.44	74.26	83.72	5.88
HieCoAtt-QI [LYB16]	0.5788	43.51	74.49	83.96	5.84
AMEM [SLH17]	0.6160	47.74	78.04	86.84	4.99
HCIAE-NP-ATT [LKY17]	0.6222	48.48	78.75	87.59	4.81
SF [JLS18]	0.6242	48.55	78.96	87.75	4.70
SCA [WWS18a]	0.6398	50.29	80.71	88.81	4.47
Ours	0.6285	48.95	79.65	88.36	4.57

Table 3.1: Quantitative evaluation of discriminative methods on val split of VisDial v0.9.

longer than 40, 20, 20 words respectively are truncated. All the texts in the experiment are lowercased. Each word is then turned into a vector representation with a look-up table, whose entries are 300- d vectors learned along other parameters during training. Thus for caption, each question and answer, we have the sequences of word embedding with size of 40×300 , 20×300 , and 20×300 , respectively. The embedding of the caption, question or answer, is passed through a two-layered LSTM with 512 hidden states and the output state is used as our final text embeddings. We use the same LSTM and word embedding matrix across question, history, caption and options.

Implementation Details: We use 2 layers of fully connected layer in Eq. (3.12). The update function $U(\cdot)$ in Eq. (3.9) is implemented as a one-layer GRU with 512 hidden states. We use a single Titan Xp GPU to train the network with a batch size of 32. In the experiments, we use the Adam optimizer with a base learning rate of 1e-3 further decreasing to 5e-5. The training converges after ~ 5 epochs.

Quantitative Results: We compare our method with several state-of-the-art discriminative dialogue models, *i.e.*, LF [DKG17], HRE [DKG17], HREA [DKG17], MN [DKG17], SAN-QI [YHG16], HieCoAtt-QI [LYB16], AMEM [SLH17], HCIAE-NP-ATT [LKY17], SF [JLS18], and SCA [WWS18a]. Tab. 3.1 summarizes the quantitative results of above competitors and


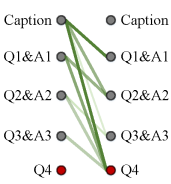

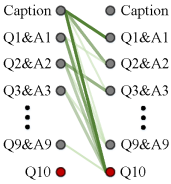

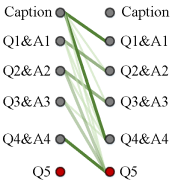
Image+Caption	Question	Human Answer	Ours	Dialog Sturcture
 <p>A hot dog covered in mustard and cheese sits next to French fries.</p>	Q1: Are hot dogs in bun? Q2: Are they on plate? Q3: Are they steak fries? Q4: Are they on table? Q5: Is it inside? Q6: Is hot dog foot long? Q7: Does it look good? Q8: Any ketchup? Q9: Any mustard? Q10: Any drink?	A1: Yes, there's only 1 though. A2: No, it's in cardboard container. A3: No, they are shoestring fries. A4: They seem to be. A5: It's hard to say, all I can see is food. A6: No, it looks like normal size dog. A7: It looks delicious. A8: No, not that I can see. A9: Yes! Lots of mustard. A10: There are no drinks.	A1: Yes. A2: No, it's in cardboard container. A3: No, they are shoestring fries. A4: It is on flat surface but table is not visible. A5: It's hard to say, all I can see is food. A6: It's looking up like it's confused. A7: It looks delicious. A8: No, not that I can see. A9: Yes! Lots of mustard. A10: There are no drinks.	
 <p>A living room with white furniture and a TV surrounded by book.</p>	Q1: Is this a large room? Q2: Is there a couch? Q3: Are there any people? Q4: It the couch large? Q5: Is there a window? Q6: Is the couch watch? Q7: Is the couch white, sorry? Q8: Is there a rug? Q9: Is it day time? Q10: Is there a TV?	A1: Yes, it's very large. A2: Yes, there is. A3: There are no people. A4: The couch is very large. A5: There is a large window as well. A6: I don't understand that question. A7: Yes, almost everything is white. A8: There is a rug. A9: Yes, it is daytime. A10: There is a flat screen TV.	A1: Yes. A2: No, only small chairs. A3: No, there isn't. A4: The couch is very large. A5: There is a large window as well. A6: I don't understand that question. A7: Yes, almost everything is white. A8: There is a rug. A9: I would say yes can't see out the window. A10: There is a flat screen TV.	
 <p>A young man and woman sitting at a table.</p>	Q1: Are they in a home or restaurant? Q2: What color table? Q3: Is it large or small? Q4: How old do they look? Q5: Do they look like a couple? Q6: Any other people? Q7: Any plates? Q8: What color? Q9: What food? Q10: What kind of cake?	A1: Restaurant, or maybe an event hall. A2: It has a white tablecloth. A3: Only part of the table can be seen. A4: Early to mid 20s. A5: It's possible but I'm not totally sure. A6: Yes, at the table behind them. A7: Yes. A8: White. A9: A piece of cake and fruit. A10: Chocolate.	A1: Restaurant, or maybe an event hall. A2: It has a white tablecloth. A3: Only part of the table can be seen. A4: I mid thirties, 2 late 30's. A5: I'm not sure. A6: Yes, at the table behind them. A7: I don't see any. A8: White. A9: Something within bacon. A10: Chocolate.	

Figure 3.4: Qualitative results of our model on VisDial v0.9 [DKG17].

our model. Our model consistently outperforms most approaches, highlighting the importance of understanding the dependencies in visual dialogue. Specifically, our $R@k$ ($k = 1, 5, 10$) is at least 0.4 point higher than SF. Our method only performs slightly worse than SCA, which adopts adversarial learning techniques.

Qualitative Results: Fig. 3.4 shows some qualitative results of our model. We summarize three key observations: **(i)** We compare our machine selected answer with human answer and show that our model is capable of selecting meaningful yet different answers compared with the ground-truth answer. **(ii)** We present our inferred dialogue structure according to the edge weight between each pair of nodes. We show that the edge weight is relatively high when the correlation between the node pairs is strong. **(iii)** Tab. 3.1 and Fig. 3.4 illustrate the interpretable and grounded nature of our model. As seen, the suggested model successfully captures the relations in dialogue and attend to dialogue fragments which are relevant to current question.

Methods	MRR \uparrow	R@1 \uparrow	R@5 \uparrow	R@10 \uparrow	Mean \downarrow	NDCG \uparrow
LF [DKG17]	0.5542	40.95	72.45	82.83	5.95	0.4531
HRE [DKG17]	0.5416	39.93	70.45	81.50	6.41	0.4546
MN [DKG17]	0.5549	40.98	72.30	83.30	5.92	0.4750
LF-Att [DKG17]	0.5707	42.08	74.82	85.05	5.41	0.4976
MN-Att [DKG17]	0.5690	42.42	74.00	84.35	5.59	0.4958
Ours	0.6137	47.33	77.98	87.83	4.57	0.5282

Table 3.2: Quantitative evaluation of discriminative methods on test-standard split of VisDial v1.0.

3.3.2 Performance on VisDial v1.0

Dataset: Then we test our model on the newest version of VisDial dataset [DKG17]: VisDial v1.0, which is collected in a similar way of VisDial v0.9. For VisDial v1.0, all the VisDial v0.9 (*i.e.*, 1,232,870 dialogue question-answer pairs on MSCOCO images [LMB14]) is used for **train**, extra 20,640 and 8,000 dialogue question-answer pairs are used for **val** and **test**, respectively.

Evaluation Protocol: In addition to the five evaluation metrics (*i.e.*, Recall@1, Recall@5, Recall@10, MRR, and Mean Rank of human response) used in VisDial v0.9, an extra metric, Normalized Discounted Cumulative Gain (NDCG), is involved for a more comprehensive quantitative performance study. Higher value for NDCG is better.

Quantitative Results: Five discriminative dialogue models (*i.e.*, LF [DKG17], HRE [DKG17], MN [DKG17], LF-Att [DKG17], MN-Att [DKG17]) were included in our experiments. Tab. 3.2 presents the overall quantitative comparison results. As seen, the suggested model consistently gaining promising results.

Methods	MRR \uparrow	R@1 \uparrow	R@5 \uparrow	R@10 \uparrow	Mean \downarrow
SF-QI [JLS18]	0.3021	17.38	42.32	57.16	14.03
SF-QIH [JLS18]	0.4060	26.76	55.17	70.39	9.32
Ours (<i>w/o iter</i>)	0.3977	25.69	54.52	70.33	9.38
Ours (<i>const. graph</i>)	0.4025	26.08	55.30	70.83	9.24
Ours (<i>full, 3 iter</i>)	0.4126	27.15	56.47	71.97	8.86

Table 3.3: Quantitative evaluation on VisDial-Q dataset with VisDial-Q evaluation protocol.

3.3.3 Performance on VisDial-Q Dataset [DKG17, JLS18]

Dataset: VisDial Dataset [DKG17] provides a solid foundation for assessing the performance of a visual dialogue system answering questions. To test the questioner side of visual dialogue, Jain *et al.* [JLS18] further propose a VisDial-Q dataset, which is built upon VisDial v0.9 [DKG17]. The dataset splitting is the same as VisDial v0.9.

Evaluation Protocol: VisDial-Q dataset is accompanied with a retrieval based ‘*VisDial-Q evaluation protocol*’, analogous to the ‘*VisDial evaluation protocol*’ in VisDial dataset detailed before. A visual dialogue system is required to choose one out of 100 next questions for a given question-answer pair. Similar methodology in [DKG17] is adopted to collect the 100 follow-up question candidates. Therefore, the metrics described in Sec. 3.3.1: Recall@ k , MRR, and Mean Rank, are also used for quantitative evaluation.

Data Preparation: We use the same text embedding techniques as we used for Sec. 3.3.1. Different from VisDial task, the first round of QA pair is given to predict next round of question. Thus the maximum round of dialogue in the VisDial-Q task is set as 9. Similar as we illustrate in Sec. 3.2.5, we construct $t+1$ node with caption and previous history as the first t nodes and the expected question as the last node. We initialize our question node with language embedding of the caption and set the language embedding of corresponding sentence as the embedding of the rest of nodes.

Quantitative Results: We follow the same protocol described in [JLS18] to evaluate our model. Tab. 3.3 shows the quantitative results for comparative methods and our ablative model variants. The ablative models include i) our model with constant graph (all edge weights are 1), and ii) our model without the EM iterations. Our full model with 3 EM iterations outperforms the comparative method in all evaluation metrics. Particularly, we can see that our model with constant graph has a similar performance to the comparative method. This shows the effectiveness of our EM-based inference process. Experiment results on this dataset also shows the generality of our approach: it can infer the underlying dialogue structure and reason accordingly about unobserved nodes (next question or current answer).

3.3.4 Diagnostic Experiments

To assess the effect of some essential component of our model, we implement and test several variants: **(i)** constant graph that fixes edge weight between each pair of nodes to be 1; **(ii)** graph without EM iteration; and **(iii)** graph with n EM iterations. Tab. 3.4 shows the quantitative evaluations of these model variants on VisDial v0.9 [DKG17]. We summarize our observations here: **(a)** model without EM iterations performs the worst among all variants. This shows the importance of iteratively updating the node embeddings. **(b)** In our experiments, message passing with 3 iterations shows the best performance of our proposed model. **(c)** model using constant graph (3 iterations) performs better than worse than the model without EM iterations, since it allows iterative updates of node embeddings. However, it is outperformed by other iterative models with a dynamic structure, since all incoming messages are treated equally. This shows the importance of edge weights: they filter out misleading messages while allowing information flow.

Methods	MRR \uparrow	R@1 \uparrow	R@5 \uparrow	R@10 \uparrow	Mean \downarrow
Ours (3 iter).	0.6285	48.95	79.65	88.36	4.57
<i>const. graph.</i>	0.6197	47.91	78.99	87.77	4.74
<i>w/o iter.</i>	0.6162	46.73	78.41	87.26	4.84
<i>2 iter.</i>	0.6213	48.18	78.97	87.81	4.75
<i>4 iter.</i>	0.6237	48.41	79.20	87.95	4.68

Table 3.4: Ablation study of the key components of our methods on VisDial v0.9 dataset.

3.4 Conclusion

In this paper, we develop a novel model for the visual dialogue task. The backbone of this model is a GNN, in which each node represents a dialogue entity and the edge weights represent the semantic dependencies between nodes. An EM-style inference algorithm is proposed for this GNN to estimate the latent relations between nodes and the missing values of unobserved nodes. Experiments are performed on the VisDial and VisDial-Q dataset. Results show that our method is able to find and utilize underlying dialogue structures for dialogue inference in both tasks, demonstrating the generality and effectiveness of our method.

Part III

Communication with Theory of Minds

CHAPTER 4

GRICE: A Grammar-based Dataset for Recovering Implicature and Conversational rEasoning

4.1 Introduction

“When a diplomat *says* yes, he *means* ‘perhaps’; when he *says* perhaps, he *means* ‘no’; when he says no, he is not a diplomat.” —Voltaire, quoted in Spanish in Escandell [Esc96] [KP20]

Voltaire’s above quote is an epitome of a crucial aspect of conversation; the meaning of the very same word or token varies according to its context and goes *beyond* what we *literally* say, which is the central character of the field of pragmatics. Such a high-level comprehension of *utterance* is more than traditional semantics and logic; it is often believed to involve the construction of the speaker’s intents, beliefs, and social institutes [Gri75, KP20]. For instance (see Fig. 4.1), when asked “did you see the apples?”, one would not merely say “yes” or “no”; instead, one should provide an answer that is cooperative, truthful, informative, relevant, and perspicuous [Dav16] based on the inferred speaker’s intent and belief. Consequently, in the above example, a person would instead answer the actual location without mentioning any positive or negative words. Such a teleological account echoes Grice’s core insight that “language use is a form of rational action; hence, technical tools for reasoning about rational action should elucidate linguistic phenomena” [GF16].

In stark contrast, such a goal-directed perspective of conversational reasoning has been

Alice: Did **you** see the **apples**?

Bob: There is a basket in the **dining room**.

(The **apples** are in the dining room.)

Alice: How many?

Bob: There are at least two.

(I am not sure how many **apples** are there.)

Alice: Did **you** put **them** **there**?

Bob: **I** was in the **kitchen**.

(I didn't put the **apples** in the dining room.)

Alice: Are all the oranges **there**?

Bob: Some are there.

(Not all the **oranges** are in the kitchen.)

Alice: What about the pears?

Bob: They are in the living room.

(The **pears** are not in the kitchen.)

Figure 4.1: An example of the conversation in the proposed GRICE dataset. Each round of dialogue includes a question, an answer that may contain implicature, and a recovered statement that converts the implicature to explicature. Different colors highlight coreference flows.

largely ignored in the modern literature of Natural Language Processing (NLP) (but see [DR95, NBG18] as exceptions). The recent development of open-ended dialogue systems has a clear trend that adopts state-of-the-art deep learning or deep reinforcement learning methods, fueled by hardware accelerations and massive sets of labeled data. However, the inspiring progress was recently challenged by researchers [SHL18, YCC18]; there remain valid concerns that systems are simply imitating human responses by regressing a large amount of training data without truly understanding it. Although we see an emerging field

of conversational reasoning (*e.g.*, [MSK19, CWL20]), existing work fails to account for the pragmatics perspective within conversations: human speakers usually do not speak their thoughts or intentions *directly*; it has to be inferred from the conversational context.

To fill the gap between the current open-dialogue systems and the future humanlike dialogue systems, we design a new open-dialogue dataset generation protocol, which we refer as Grammar-based dataset for Recovering Implicature and Conversational rEasoning (GRICE), in homage to H. P. Grice for his influential theory in explaining and predicting conversational implicatures [Gri75]. Specifically, our design follows four principles.

First, we design the GRICE dataset with a focus of *conversational implicature* [Gri75], “one of the single most important ideas of pragmatics” [Lev85]. Naturally, the ability to successfully perform **implicature recovery** in conversation [Bor09] would be a suitable indicator of a system’s performance; we adopt it as part of our evaluation protocols. To recover conversational implicature into explicit ones with only information and context in the dialogue, an ideal model should reason about the dialogue context and the relations among dialogue entities.

Second, we emphasize the comprehension of the *conversational context* and adopt the **conversational reasoning** as part of the evaluation protocols. Again, we take the conversation in Fig. 4.1 as the example: When the speaker says “I was in the kitchen,” what she really means is that she was not in the dining room and therefore could not put the apples there. The same response would have the opposite meaning when the question becomes “Were you in the kitchen?”. Such a swift switch according to its dialogue context is a quintessential demonstration that human communication is a context-dependent endeavor [Fet17] and a dynamic construct, which relates communicators and the language that they use in a dialectical manner [Bat00].

Third, we build the GRICE dataset by incorporating five different types of implicature; see details in Sec. 4.4. To resolve these types of implicature, the algorithm ought to make a proper prediction or inference of intents and beliefs by representing and reasoning about

triadic relations [Sax06]: the speaker’s belief, the addressee’s belief, and what they have or communicate in common.

Fourth, in comparison to pioneering work Facebook bAbi [WBC15] and its follow-up work ToM [NBG18] that evaluate different aspects of reasoning with a set of toy tasks, the proposed GRICE dataset does not sacrifice crucial characteristics of modern open-dialogue systems. On the contrary, by integrating pragmatics and implicature in conversation, we hope to shed light on some challenging issues in open-ended dialogue:

- *Coreference* resolution [CZC17, KMP18b] refers to finding all expressions that refer to the same entity in the conversation. The significance of resolving coreference becomes even more profound in conversations with implicature; Fig. 4.1 gives an example and highlights the coreference flows in different colors.
- *Commonsense* reasoning [SRC19, THL19, SCH17] received an increasing attention in NLP. Notably, the Winograd [LDM12] and WinoGrande [SBB20] challenges have been proposed to examine commonsense reasoning. For conversations with implicature, commonsense reasoning reflects a crucial concept of *relevance*. For instance, to understand the conversation “A: I am out of petrol. B: There is a garage around the corner.”, one needs to have the commonsense about “a garage could store petrol” to resolve implicature.
- *Logic*-based methods were once thought to be the “ideal language” approach to the semantics of human language [Rus03], but were later challenged by [Wit53, Wit69] and [Gri75]. However, this disagreement should not prohibit the central role of logical forms in reasoning tasks. In fact, it would be interesting to investigate if the modern end-to-end trainable methods could benefit from logical forms in conversational reasoning.

The remainder of this paper is organized as follows: We review related work on dialogue dataset, implicature, and conversational reasoning in Sec. 4.2. In Sec. 4.3, two tasks are defined for evaluations. We present detailed design, generation, and analysis of the GRICE dataset in Sec. 4.4. By introducing two evaluation protocols, we provide the performance of baseline models with discussions of results and future directions in Sec. 4.5.

Table 4.1: Comparing GRICE with existing conversational datasets.

Dataset	Task	Context	Source Domain
Ubuntu [LPS15]	Next Utterances Prediction	Dialogue	Ubuntu Chat logs
PERSONA-CHAT [ZDU18]	Next Utterances Prediction	Dialogue	Persona
Douban [WWX17]	Next Utterances Prediction	Dialogue	Open Domain
MuTual [CWL20]	Next Utterances Prediction	Dialogue	Listening Comprehension
DREAM [SYC19]	Question Answering	Dialogue	English Language Exams
CoQA [RCM19]	Conversational QA	Paragraph	Literature
GRICE (ours)	Implicature recovery & Question Answering	Dialogue	Open Domain with implicature

4.2 Related Work

Dialogue Datasets Dialogue datasets have been focusing on predicting the next most-likely response by imitating the teacher’s responses (human corpus) [LPS15, ZDU18, WWS18b]. However, as pointed out by [CWL20], prior datasets and associated methods lack proper explicit reasoning modules; it later becomes evident that such reasoning modules serve as the scaffold in building a humanlike conversational agent. Note that a model’s reasoning capability is minimal if it simply converts various reasoning challenges into a categorization problem when predicting the utterances; it still tends to choose the most frequent answer given the training set, without truly making sense of the context and underlying meaning.

To the best of our knowledge, the proposed GRICE dataset is the first open-dialogue dataset that explicitly integrates implicature; see a detailed comparison in Tab. 4.1. We hope our design would encourage or necessitate future models to make explicit reasoning on conversational contexts, commonsense, and agent’s intents and beliefs. The most similar dataset in terms of the format is DREAM by [SYC19], a conversational dataset with a question-answering (QA) task. However, the design of this dataset does not require much reasoning; answers can be directly extracted. The most similar dataset in terms of the task

is CoQA by [RCM19], which considers pragmatics and QAs over literature paragraphs; the proposed GRICE dataset differs by reasoning over the *dialogue context* between two agents.

Implicature Implicature has been extensively studied in the field of linguistics and philosophy since the inception of pragmatics; Grice’s four maxims [Gri75]—quality, quantity, relevance, and manner—founded the principles of the interpretation of conversational implicature. Two neo-Gricean typologies of conversational implicature include [HW04]’s Q- and R-implicature and [Lev85]’s Q-, I-, and M-implicature. The relevance theory developed by [SW86] offers an alternative account than Gricean and neo-Gricean theory. In general, although these doctrines provide crucial insights into the field, they focus more on philosophical debates over toy examples, without proposing computational solutions or validating the ideas on large-scale natural language datasets.

Recently, a few computational models have been proposed (*e.g.*, [FG12, GS13]); however, these models assume the space of utterance and possible semantic meanings are finite or given, so that models only need to pick up one over others based on the shared context. Other models focus on more specific tasks; for instance, recovering the direct meaning from the indirect answer using scalar adjectives [MMP10, DB13], conducting analysis on the ironic implicature behind simile [VH10].

By generating paired sentences in a semi-automatic fashion with human annotations, [JWB20] recently devise a dataset with a focus on scalar implicature [Hir85]. In comparison, the proposed GRICE dataset has a much more natural setup and broader scope by combining the multi-round open-dialogue with conversational implicature. Additionally, leveraging a grammar representation for fine-grained control, the GRICE dataset is generated in a fully automated fashion without human annotations. We hope such a design could boost researchers in implicature, pragmatics, and conversational reasoning at a large scale.

Conversational Reasoning In the past four years, we have witnessed an increasing interest in conversational reasoning in various contexts. OpenDialKG [MSK19] incorporates external knowledge graphs to the dialogue context to provide extra entities as responses. Visual Dialog [WWS18b, ZWQ19, DKG17] takes images as external multi-modalities to jointly reason with dialogue context to generate visually grounded responses. MuTual [CWL20] modifies English reading comprehension to select the next best response by machine reasoning.

However, prior efforts have ignored the fact that humans commonly do not directly speak out answers. The proposed GRICE dataset is a complement of prior conversational reasoning tasks; it focuses on implicature with conversational reasoning, which does not reject multi-modalities as they could be a source of commonsense knowledge.

4.3 Task Definition

To evaluate how well a model “understands” the dialogue presented in the proposed GRICE dataset, we devise two tasks: the implicature recovery task and the conversational reasoning task, wherein the latter task depends on the successful completion of the former task. Below, we introduce the setup and evaluation protocol of each task.

Task 1: Implicature Recovery Formally, an n -round dialogue occurred between two agents is represented by a sequence of QA-pairs $\{(Q_1, A_1), (Q_2, A_2), \dots, (Q_n, A_n)\}$, where Q_i is the question raised by the first agent, A_i is the response provided by the second agent, which may contain an implicature. To complete this task, a model is asked to identify if A_i is a statement containing implicature, and if this is true, to resolve the implicature to its explicit form, *i.e.*, to perform implicature recovery.

The implicature recovery is evaluated in the form of multiple choices: For each utterance, the ground-truth condition (with implicature) and its explicit form are given when generating

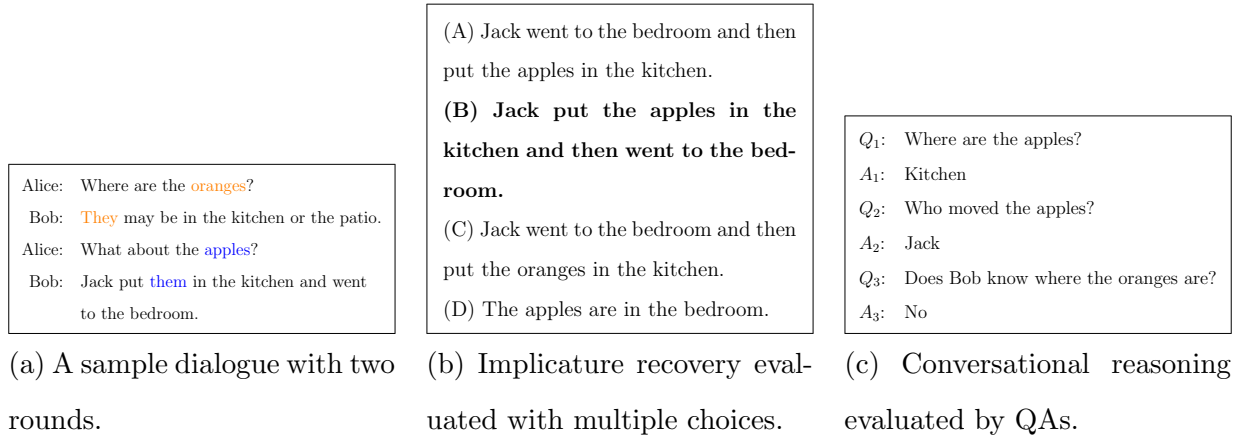


Figure 4.2: Examples of two tasks defined in GRICE dataset. (a) Given a multi-round open-dialogue, an algorithm is asked to perform (b) implicature recovery and (c) conversational reasoning in the form of QAs.

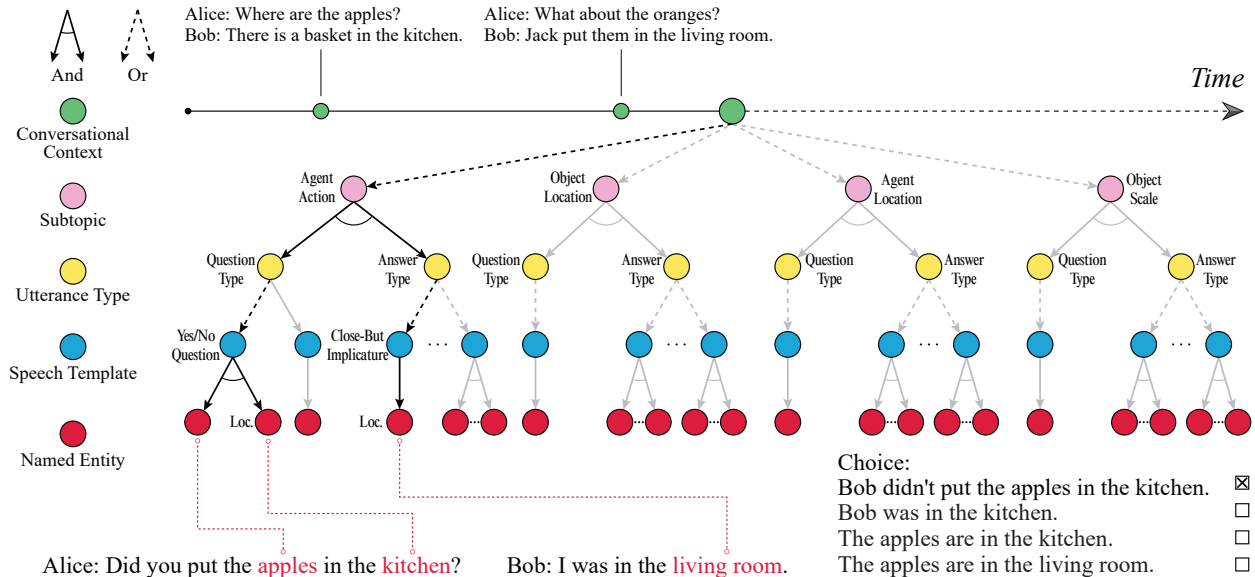


Figure 4.3: The graphical illustration of the grammar production rules for the GRICE dataset.

the dialogue; the explicit form, which not only recovers the implicature but also resolves coreferences in the utterance, serves as the correct answer in the multiple choices. We then sample three possible answers from the candidate pools, given a set of manually defined *speech templates* (see details in Sec. 4.4). Figs. 4.2a and 4.2b show an example: The last

utterance by Bob implicates (by the word “then”) the temporal order between “put them in the kitchen” and “went to the living room.” Thus, the correct implicature recovery should resolve “them” as “the apples” and recover the correct temporal order.

Two strategies developed by existing work could be adopted to address this task. One strategy is to train a model that directly chooses an answer from the candidate answers. Another more challenging strategy is to train a generator that chooses the answer by computing the log-likelihood scores and ranking the candidate answers as done in [DKG17]. To quantitatively evaluate the performance, we use the standard response selection metrics [LPS15, WWX17, CWL20]: Top 1 Recall (R@1) and Mean Reciprocal Rank (MRR) [Voo99].

Task 2: Conversational Reasoning To evaluate the open-ended conversational reasoning, we follow the protocols specified in [WBC15] and [NBG18] with comprehensive QAs. For each dialogue, we generate questions by randomly sampling the *conversational contexts* (see details in Sec. 4.4), and each question could be answered by a single word; see some examples in Fig. 4.2c.

4.4 Creating the Grice Dataset

Representation We adopt a structural grammar model—Temporal And-Or Graph (T-AOG) [QJZ20, TPZ13]—to represent the dialogue context due to its expressiveness of hierarchical dialogue structure and temporal-dependent dialogue flow. We represent one *turn* of the dialogue as an AOG [BHW79, BHW81, Pea84, ZM07] that has a hierarchy of five levels: conversational context, subtopic, utterance type, speech template, and named entity. These AOGs are connected w.r.t. temporal constraints in order to assemble the T-AOG.

Formally, an AOG (*i.e.*, each turn of the dialogue) has two sets of non-terminal vertex: (i) a set of And-nodes, wherein each node represents the decomposition of a larger concept (*e.g.*, subtopics) into smaller components (*e.g.*, utterance types), and (ii) a set of Or-nodes, wherein

each node branches to an alternative decomposition (*e.g.*, a conversational context could have different types of subtopics), enabling the model to reconfigure the overall dialogue. An instance of AOG can be sampled by selecting a child node for each of the Or-nodes, resulting in a parse graph.

Fig. 4.3 illustrates an example of AOG. Specifically, the root node of one dialogue turn is an Or-node, representing the current conversational context. Represented by an And-node, each child node of the root node denotes a subtopic of the current dialogue turn. The subtopic is composed of a set of utterance types, further decomposed into speech templates filled by named entities. Instantiating an AOG by selecting Or-nodes would produce a complete utterance of a dialogue turn and pose constraints on the next dialogue turn.

Conversational Context We follow [WBC15] to represent dialogue context by a simulated world with various dialogue entities: *objects*, *locations*, and *agents*. We randomly initialize a world for each dialogue snippet by (i) positioning objects in locations with a random scalar (one, two, ...), (ii) randomly setting a location for each agent as the “previous agent location,” and (iii) for each $\langle object \rangle$ in $\langle location \rangle$, randomly selecting an $\langle agent \rangle$ in $\langle location \rangle$ to denote that “ $\langle agent \rangle$ put the $\langle object \rangle$ in the $\langle location \rangle$.”

Subtopic In this dataset, we focus on four different subtopics: *agent_location*, *agent_action*, *object_location* and *object_scale*; see examples in Tab. 4.2. Specifically, *agent_location* queries the location of some $\langle agent \rangle$. The example in Tab. 4.2 implicates that “Jack was in the kitchen.” Similarly, *object_location* queries the location of some $\langle object \rangle$. *Agent_action* queries the previous action taken by some $\langle agent \rangle$ on some $\langle object \rangle$. Typically, the action can be identified as an $\langle agent \rangle$ put $\langle object \rangle$ in the $\langle location \rangle$. *Object_scale* queries the quantity of some $\langle object \rangle$. In particular, an algorithm should also be able to reason about the strength among the quantifying phrases, such as *at least*, *some*, and *all*. a typical example shown in Tab. 4.2 implicates that “Bob does not know if all the apples are in the kitchen.”

Subtopic	Example		Train	Dev
agent_location	Alice: Where was Jack? Bob: I saw him in the kitchen.			
agent_action	Alice: Did you put the apples in the kitchen? Bob: I was in the bedroom.	Explicit Answer	27.3	29.6
object_location	Alice: Where can I find the apples? Bob: They are in the kitchen, if not the living room.	Implicature	72.7	70.4
		Relevance	9.9	9.3
		Strengthening	22.5	22.9
		Limiting	6.3	6.4
		Ignorance	23.5	21.2
object_scale	Alice: Are all the apples in the kitchen? Bob: At least four are there.	Close-But	10.5	10.8

Table 4.3: Distribution of implicature types (%).

Table 4.2: Categories and examples of different subtopics in GRICE dataset.

Utterance Type Utterance type concerns how to generate a QA-pair correctly. For questions, query types of each subtopic are manually defined. For example, the question regarding `agent_location` can be categorized into yes/no question (“were you in the kitchen?”) or where question (“where were you?”). For answers, we focus on five different types of implicature [Hua17, HW04, Dav16]: *relevance*, *strengthening*, *limiting*, *ignorance*, and *close-but*; see Supplementary Material for detailed definitions and examples.

Diversity We follow [WBC15] to use a simple automated grammar to makes the conversation more natural and diverse: We assign a set of synonyms for each verb; *e.g.*, we randomly replace (i) *put* with *left*, *dropped*, or *placed*, and (ii) *went* with *travelled*, *journeyed*, or *walked*.

Since coreference is a crucial feature in the conversational context in GRICE dataset, we

track agents, objects, and locations mentioned in previous conversations and replace them with deixis in the following conversational context.

Additionally, we build a set of follow-up questions for each type of dialogue actions to challenge the model’s ability in reasoning about the omission in utterances. Take Fig. 4.2 as an example; the question “What about the apples?” should be interpreted or recovered as “Where are the apples?” during the reasoning procedure.

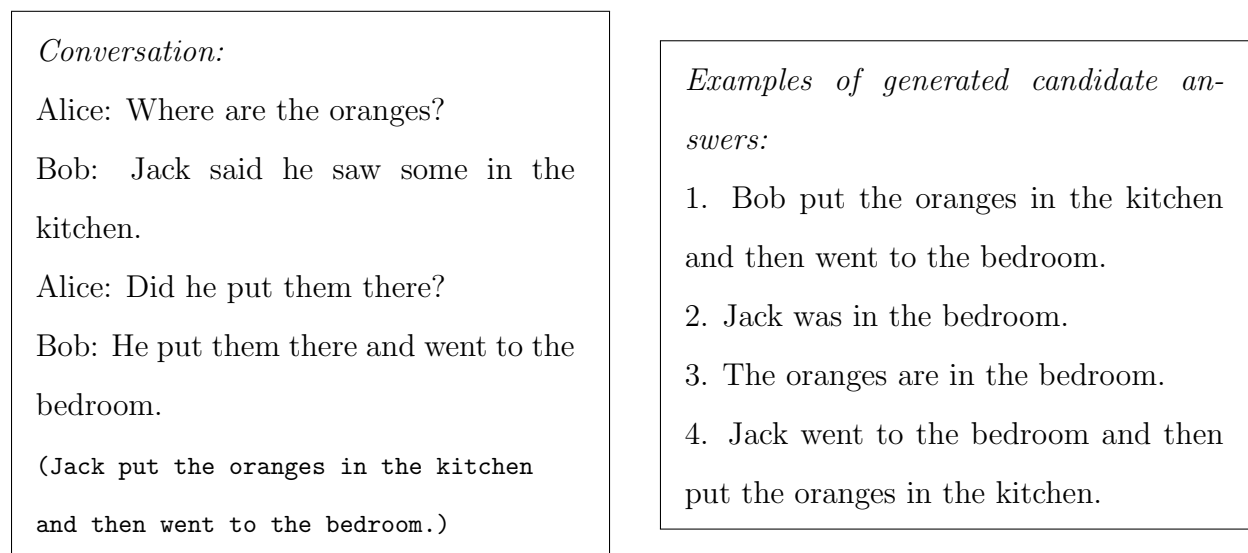


Figure 4.4: The candidate answers for the implicature recovery task are generated following four different strategies: 1. Statements that are similar to the ground-truth condition but with wrong coreferenced entities. 2. Random sampled true condition but with irrelevant facts. 3. Random sampled wrong facts from the conversational context. 4. Manually created statements that are close to the true condition but are in fact wrong.

Candidate Answer Generation To generate candidate answers for each round of dialogue for the implicature recovery task, we define four different strategies tailored to produce challenging candidates. Among all four candidate answers, besides the ground-truth condition in its explicit form, the other three candidate answers are randomly sampled from the candidate pool, composed by applying the following strategies; see Fig. 4.4 for examples of

each strategy:

1. Statements that are similar to the ground-truth condition but with wrong coreferenced entities.
2. Randomly sampled true condition but with irrelevant facts.
3. Randomly sampled wrong facts from the current conversational context.
4. Manually created statements that are close to the true condition but are in fact wrong.

Questions We follow [WBC15] to generate questions about the dialogue context. After sampling the dialogue turns and finalizing the dialogue context, we query current dialogue states in terms of agent locations, agent actions, object locations, and object scales. Inspired by [NBG18], we further add the belief query (*e.g.*, “does Bob know where the oranges are?”) to test the model’s capability of belief reasoning. See Fig. 4.2 for examples.

4.5 Experiments

We randomly sample 6,000 dialogues as the train set and additional 4,000 dialogues as the dev set to evaluate baseline models; each dialogue contains 10 dialogue turns and 3 questions. Detailed distributions of implicature types are listed in Tab. 4.3. For the test set, we sample 1,000 dialogues in each implicature category, resulting in a total of 5,000 dialogues. Each test dialogue contains 3–5 dialogue turns and one question on implicature. All data is clean and noiseless.

Setup We model both tasks as a query over the conversational context. Specifically, for the implicature recovery task, we define $h_t = (Q_t, A_t)$ as the queried sequence and the $H_t = \{(Q_1, A_1), \dots, (Q_{t-1}, A_{t-1})\}$ as the past dialogue context. Then the task is to predict the explicit form $E_t = f(h_t, H_t)$. For the conversational reasoning task, we treat the entire history as the input context and the question as the query sequence. The task is then modeled

as a Sequence-to-Vector framework that maps the query with its context to the vocabulary space. We implement all the models in PyTorch and trained using ADAM [KB15] with a learning rate of 0.001 for 40 epochs.

4.5.1 Baseline Models

We evaluate 5 representative baseline models for both tasks on the GRICE dataset. The baseline models are chosen on the basis of performing well on synthetic language datasets (*e.g.* Facebook bAbi) or similar tasks and easy adoption to perform conversational reasoning tasks. We additionally test the performance of transformer-based language models that are claimed to have strong reasoning capabilities.

LSTM We start with a simple dual LSTM model: one LSTM to encode the history context as a long context sequence, and another LSTM to encode the queried sequence. A simple MLP fuses two encoded vectors to predict answers.

Recurrent Entity Network (EntNet) EntNet [HWS16] is an RNN-based memory-augmented architecture, capable of capturing the sequential nature and learning relevant entities with their properties by gated recurrent units and weight matrices. Our implementation is based on its official open-sourced code¹.

Relation Network (RelNet) [SRB17] propose a neural model for relational reasoning. The algorithm considers each pair of sentences together with the question as inputs. Our implementation is based on its official open-sourced code².

¹<https://github.com/jimfleming/recurrent-entity-networks>

²<https://github.com/siddk/relation-network>

Memory Network (MemNN) We follow [WCB14] to build a memory network³ that takes each round of history context as a supporting fact and stores it in the memory bank; the algorithm is expected to learn to refer the memory when predicting answers. Specifically, we use an LSTM to encode each round of history and compute the association matrix between the queried sequence and the memory bank. We apply a softmax to the association matrix to get attended weight of the dialogue history. Finally, we compute the attended dialogue history embedding and combine it with the queried embedding using a simple MLP to predict answers.

Transformer-based Language Model Fine-tuning transformer-based language models (*e.g.*, GPT [RNS18] and BERT [DCL18]) has shown superior performance on conversational reasoning tasks [SYC19]. We use BERT-base-uncased⁴ as our pre-trained model and apply it to the conversational reasoning task by adding a single linear layer to generate answers from the target vocabulary set.

Human Performance We randomly selected 100 dialogues and assigned them to 40 human subjects in a between-subject design; 20 subjects for the implicature recovery tasks, and another 20 subjects for the conversational reasoning task.

4.5.2 Evaluation and Results

Implicature Recovery We start by evaluating the performance of the baseline models on the implicature recovery task. As discussed in Sec. 4.3, we evaluate under two different settings to predict the implicature recovery results: the discriminative setting and the generative setting (marked by “-Gen”). For the discriminative setting, we take the encoder output and compute the similarity score with each candidate answer to predict the final choice. For

³<https://github.com/facebook/MemNN>

⁴<https://github.com/huggingface/transformers>

Model	Dev		Test	
	R@1	MRR	R@1	MRR
LSTM	81.92	0.9046	83.54	0.9145
EntNet	89.07	0.9445	91.15	0.9523
RelNet	93.02	0.9623	95.33	0.9602
MemNN	96.76	0.9833	97.29	0.9862
LSTM-Gen	62.28	0.7763	65.02	0.7784
MemNN-Gen	86.29	0.9305	88.79	0.9418
Human	99.00	-	98.50	-

Table 4.4: Performance on implicature recovery task.

Model	Accuracy (%)	
	Dev	Test
LSTM	59.77	55.82
EntNet	57.91	53.17
RelNet	63.02	65.50
MemNN	64.66	67.32
BERT	67.21	71.06
MemNN w/ inf	69.24	73.12
Human	98.50	97.50

Table 4.5: Performance on conversational reasoning task.

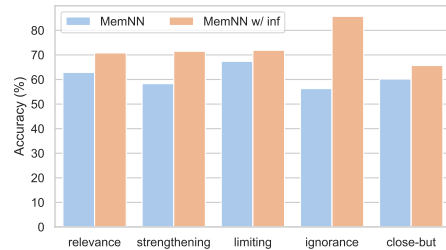


Figure 4.5: Performance comparison between MemNN and with additional inference module (MemNN w/ inf) that explicitly recovers the implicature.

the generative setting, we train the encoder-decoder framework using the teacher-forcing algorithm by minimizing the negative log-likelihood between the generated answers and the ground-truths. Overall, the generative setting is more challenging than the discriminative one; see Tab. 4.4 for results on dev and test sets.

Conversational Reasoning We follow [WBC15] and [NBG18] on performance evaluation of the conversational reasoning task, measured by the accuracy score in the vocabulary space; see Tab. 4.5 for the results of all the baseline models on the dev and test sets.

Analysis Comparing human performance and the model performance in Tabs. 4.4 and 4.5, we see a consistent and competent performance in human subjects, whereas the model performance of the conversational reasoning task drops significantly even after a relatively good performance on the implicature recovery task. This contrast indicates that the models that perform well on the implicature recovery task may not really “understand” the conversational context to be used in the following conversational reasoning task.

To further test this hypothesis, for the implicature recovery task, we additionally pre-train an inference encoder that predicts the explicit/recovered answer under the generative

settings (MemNN w/ inf), given the previous dialogue history. This additional inference model is further appended into the basic model and fused to predict the final answer. Such a setting would be a reasonable test to see how well a model could perform if they explicitly incorporate the recovered implicature from the implicature recovery task to solve the later conversational reasoning task. As shown in both Tab. 4.5 and Fig. 4.5, we observe that the conversational reasoning performance improves an average 5% with this additional inference module; for certain implicature types, it boosts the performance for more than 25%. Note that it even outperforms the previous state-of-the-art model that fine-tunes the pre-trained Bert model, indicating the significance of incorporating an explicit module of implicature recovery for pragmatic reasoning in conversation.

General Discussions Taken together, the results show that the existing models do exhibit a certain level of reasoning capability, though weak. Furthermore, the performance gap between the implicature recovery task and conversational reasoning task leaves us many mysteries. Humans seem to be reasonably consistent in solving both tasks, whereas current models are not. One possible explanation is that the computational model is able to fit the relatively confined space of the implicature recovery task based on the training data, but fails to incorporate such knowledge for the more open-ended conversational reasoning task. This possible explanation is further backed up by the above experiment with an additional inference module. All these observations call for future research for investigations.

Although the proposed GRICE dataset incorporates the triadic relations among agents and additional challenges (coreference, commonsense, *etc.*) existing in modern dialogue systems, it is difficult to directly evaluate these aspects on an open-ended dialogue system, especially with implicature. One may use an indirect metric: Whether the system performance would improve after integrating such modules. Moving forward, we call for future research to design more direct evaluation metrics in addition to the present implicature recovery and conversational reasoning tasks.

More importantly, how could we properly leverage the knowledge extracted during the implicature recovery task for the following conversational reasoning task? [Lev95] argues that human conversation depends on intention-ascription, where inferences must be made way beyond the data, therefore forming an *abductive* process. A possible and promising future direction would be using a neural-symbolic solver, capable of handling noisy inputs using neural-network modules and reasoning about the answers in a logic-like style.

CHAPTER 5

Generating Explanations with Human Utility

5.1 Overview and Background

5.1.1 Background and Motivation

Background In the past decade, we have witnessed the growth of machine’s learning capability to handle noisy real-world inputs with impressive performance, fueled by large datasets. Meanwhile, the community has realized the necessity of machine interpretability [ZZ18, ZWZ18] for safety-critical applications. Intrinsically, most of the existing models are not designed to simultaneously maximize the performance and explainability [GA19], hence resulting in a need for a trade-off between the performance and explainability. This trade-off often leads to a debate between the black-box models *vs* the white-box models: Models with high performance usually lack explainability, whereas models with relatively high explainability often perform poorly in real-world scenarios.

Recent trends in neural-symbolic approaches [YWG18, MGK18, LHH20, PMS16] refute the above need for the trade-off; a hybrid model could possess high performance in complex reasoning task while maintaining relatively high interpretability. Significantly, a robot system [EGL19] has recently demonstrated the efficacy of such an approach using a large-scale, between-subject study. The finding echoes the above conclusion: Forms of explanation that are best suited to foster trust do not necessarily correspond to the model components contributing to the best task performance; by integrating model components to enhance both task execution and human trust, a machine system could achieve both high task performance

and high human trust. Crucially, it also shows that the means of delivering explanations matters: Providing high-level semantics meaning is not sufficient to boost human trust; such explanations should not decouple from the participants’ observations of the robot’s task execution.

Motivation Despite the above progress, existing systems demonstrating specific levels of explanations are still rudimentary in terms of the forms of explanations. Existing systems mostly emphasize *hierarchical decompositions* (either spatial or temporal) of the systems’ inner decision-making process, either by visualizing the saliency/attention maps of deep neural network’s layers [ZWZ18, ZZ18, AWZ20, ZWW20], or by tracing top-down/bottom-up process of the graph/tree structures [LZS18, EGL19, EQZ19, EMQ20, ZRH20, ZZZ20]. Thus, the explanations and interpretability are primarily *machine-centric*; the process only unfolds the model for a human user to probe or inspect. Critically, human users’ active interactions or inputs with the systems rarely change the behavior of the machine’s decision-making process, and the machine’s responses are primarily based on pre-computed and stored information. We call this the *passive machine—active human* paradigm, wherein an active human user may *query* the state of the machine to *passively* acquire the explainable information.

We argue that human-machine teaming should follow a different and more user-friendly paradigm, which we call the *active machine—active human* [QLZ20] paradigm. In such a new paradigm, the machine would adopt the human user’s input and change its behavior in *real-time* so that the system and the human user would *cooperatively* achieve a common task. Hence, such a cooperation-oriented human-machine teaming would require the machine to possess a certain level of ToM: A machine would behave like a human agent to *actively* infer the human user’s belief, desire, and goals [YLF20, GGZ20]. The system’s design is no longer limited to display its decision-making process, but further to understand human’s needs to cooperate, therefore forming a *human-centric* process. Critically, the essence to establish

such a cooperation lies in the *shared agency* [TSZ20, SZZ20] or *common mind* [Tom10].

Motivated to build an Explainable Artificial Intelligence (XAI) system with the aforementioned characteristics, capable of understanding human user’s beliefs, design, and goals, we move from conventional explanation tasks on function approximation (*e.g.*, classification) to tasks involving sequential decision-making. These decision-making tasks include extensive human-machine teaming, dealing with complex constraints over problems intractable to the human’s inferential capabilities. By resolving the discrepancy between robot’s and human’s expectations and mental models, we hope the XAI system could assist the human user to discover the provenance of various artifacts of a system’s decision-making process over long-term interactions even as the physical world evolves [GA19, CSK20]. We believe this research direction is the prerequisite for generic human-machine teaming.

5.1.2 Overview

Game Design We devise a human-machine teaming system presented as a collaborative game, in which the human user needs to work together with a group of robot scouts to accomplish some tasks and optimize the group gain. In this game, the human user and robot scouts communicate on a constrained channel: Only the robot team directly interacts with the physical world; the human user does not directly access the physical world or direct control over robot scouts’ behavior. Meanwhile, only the human user has access to the ground-truth value function of the task (*e.g.*, minimize overall time); the robot team has to infer this value function through human-machine teaming. Such a setting realistically mimics real-world human-machine teaming tasks, as many systems perform autonomously in dangerous settings under human users’ supervision.

The XAI system is expected to provide appropriate explanations to justify its behaviors and gain human user’s trust and reliance. This process is achieved by actively inferring the human user’s mental model (*i.e.*, value and utility as the instantiation of the belief, desire, and goals) during the game. Therefore, the system’s explanation generation is a *bidirectional*

dialogue framework: The XAI system needs to both “speak” and “listen”—explaining what it has done and plans to do based on its inference of the human user’s value and utility. In the meantime, the human user is tasked to command robot scouts to reach the destination while maximizing the team’s score. Hence, the human user’s evaluation of the XAI system is also a *bidirectional* process: The human user has to infer the goal of robot scouts and check if it aligns with the given value function of the task. Ultimately, if the XAI system works well, the robot scout value function should align well with the ground-truth value function given only to the human user, and the human user should gain high trust from the XAI system. Our methodology studies XAI in a full-blown communication system, a combination of our previous XAI work on autonomy, including theory-of-mind, communicative learning, value-alignment, and causal reasoning for effective explanation generation.

Our design encourages natural human-machine teaming and bidirectional reasoning as both parties have crucial but private information at the beginning of the game. The robot scouts possess information about the map but lack access to the human user’s value function, which determines mission goals, hindering the robot scouts’ ability to make proper decisions that reflect the human user’s intent. Meanwhile, the human user, who knows the task’s value function that governs the decision-making process, lacks direct access to the environment. By allowing constrained communication to fulfill human-machine collaboration, the robot scouts can make sporadic action proposals to the human user, and the human user provides a binary accept/reject feedback, which the robot scouts will then utilize to infer the human user’s value function and adjust robot scouts’ behaviors accordingly. Based on adjusted behavior, the human user will provide ratings or the trust/reliance of the XAI system. In our setting, the communication’s main purpose is to align the value function between the human user and the robot scouts. For a fast alignment, the robot scouts need to know when and how to make proposals, such that feedback from the user is most informative to estimate the value function correctly. To obtain instructive feedback from the human user, the robot scouts must establish a shared agency or common mind—what the human user knows and

believes, what the human user intends to do, and what are aligned and misaligned. Only based on this shared agency could the robot scouts provide explanations that properly justify previous actions and current proposals.

Besides the value alignment process, our design also involves estimation of human user’s utility, *i.e.*, the human user’s preference of the forms of explanations. In contrast to the objective value function given to the human user, this utility-driven human user’s preference is subjective and more likely to be individually different. We argue that a properly modeling of such an individual difference plays a crucial role in gaining human trust and reliance. The human user’s value function and utility together form the human user’s mental state.

Game Setting Our collaborative game, Robot Scout Exploration Game, has a minimal design and involves one human commander and three robot scouts. The game’s objective is to find a safe path on an unknown map from the base (located at the bottom right corner of the map) to the destination (located at the upper left corner of the map). The map is represented as a partially observed 20×20 tile board, with each tile potentially holding one of the various devices and remain unobserved until a robot scout moves closer to it.

We define a set of goals for the robot scouts to pursue as they find the path to reach the destination, including (i) saving time used to reach the destination, (ii) investigating suspicious devices on the map, and (iii) exploring tiles, and (iv) collecting resources. The game’s performance is measured by the accomplishment of these goals by the robot scouts and their relative importance (weights), defined as the human user’s value function. Again, this value function is only revealed to the human user, not the robot scouts.

One comparable but different setting to our human-machine teaming framework is the inverse Reinforcement Learning (RL). Inverse RL aims to recover an underlying reward function given *pre-recorded* expert demonstrations in a *passive* learning setting. In contrast, the agent (the collective form of all robot scouts) in our system is designed to *interactively* learn from *scarce* supervisions given by the human user. Crucially, our design requires

the agent to actively infer human user’s mental model (value and utility) to *cooperatively* accomplish a task, a unique proper of *human-centric* learning scheme. In a nutshell, the agent is tasked to perform value-alignment by inferring human user’s mental model, actively make proposals, and evaluate the human user’s feedback, requiring complicated and recursive mind modeling of the human user.

5.2 Methodology

In this section, explain the details of your study and any justifications necessary for your approach.

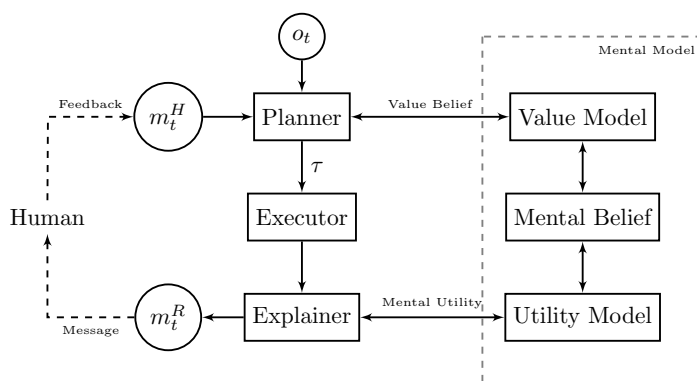


Figure 5.1: Algorithmic flow of the computational model.

5.2.1 Computational Model

In this section, we provide an overview of the game flow and its associated computational model. Throughout the paper, we use R and H to denote the robot scouts and the human user, respectively. θ is the parameters of the value function, s the physical state, $b(\cdot)$ the belief over latent variables, and $x = (b_s, b_\theta, b_v)$ the mental state (value and utility) of the human user. m is the message used for human-machine communication. BU stands for the

Algorithm 3 High-level game flow.

```
1: Set  $t = 1$ , initialize  $s^t$ , agent's mental state  $x_0^R$ ;  
2: while stop condition is not satisfied do  
3:    $o_t \sim O(s_t)$   
4:    $\widehat{x}_t^R = BU_1(x_{t-1}^R, o_t)$  // collect observation from the environment  
5:    $m_t^R \sim \lambda_R(\widehat{x}_t^R)$  // update belief given observation  
6:    $x_t^R = BU_2(\widehat{x}_{t-1}^R, m_t^R, m_t^H)$  // generate message (proposal & explanation) to the user  
7:    $\mathbf{a}_t^R \sim \pi(x_t^R)$  // update belief given user feedback  
8:    $s_{t+1} \sim T(s_t, \mathbf{a}_t^R)$  // agent's policy  
9:    $t = t + 1$  // state transition  
10: end while
```

belief update sub-processes, where BU_1 is on the physical state, and BU_2 is on the value function. λ_R manages the generation of the messages to the user, including proposal and various modes of explanations. Other notations (o, t, O, T , and π) follow standard partially observable Markov decision process (POMDP) [SV10] definitions; see Tab. 5.1 for a summary of the notations.

Every round of the game starts with the robot scouts receiving observations from the environment and making a task plan based on their current mental state. Next, they generate messages (proposals and/or explanations) to the human commander for feedback, using which they make final moving decisions for this round and execute their actions. A high-level game flow is sketched in Algorithm 3, and the computation pipeline for one round of human-machine teaming is shown in Fig. 5.1. Since the game directly displays the most probable map information to the human user, we assume the communication from the agent to the human user is noise-free. Hence, after laying out the formulation of the agent policy (see Sec. 5.2.1.1), we focus on how the agent updates belief over human user's value function (BU_2) (see Sec. 5.2.1.2) and how the communication messages are generated (λ_R) (see Sec. 5.2.1.3).

5.2.1.1 Agent Policy

Suppose the robot scouts already know about the human user’s value function, the game simplifies to a POMDP setting, solvable by planning-based methods [SV10]. Let τ_i denote the plan proposed by the i -th scout and $\tau = \{\tau_1, \dots, \tau_K\}$ as the complete plan of the scout group, where K is the number of scouts in the group. When making a plan, the scouts

Table 5.1: Notations adopted in the computational model.

Notation	Description	Remark	Notation	Description	Remark
$s \in S$	Physical State	N/A	$m^E \in M^E$	Robot’s explanation	N/A
$o \in O$	Observation	N/A	$m^P \in M^P$	Robot’s proposal	$\mathcal{T} \subset M^P$
$t \in T$	Time Step	N/A	$m^R \in M^R$	Robot’s message	$m^R = (m^P, m^E)$
$\theta \in \Theta$	Human’s value function	N/A	$fb \in FB$	Proposal feedback	$m^H(fb) \in \{0, 1\}^K$
$v \in \Upsilon$	Human’s utility function	N/A	$ss \in SS$	Satisfactory Score	$SS \subset \mathbb{Z}^+$
\mathbf{a}^R	Joint action of all scouts	$\mathbf{a}^R = (a_1^R, \dots, a_K^R)$	$m^H \in M^H$	Human’s message	$m^H = (fb, ss)$
b	Belief over hidden variables	$b(\cdot)$ means the belief function	λ_R	Robot’s communication policy	$X^R \times M^R \longrightarrow [0, 1]$
$x^R \in X^R$	Robot’s mental state	$x^R = (b(s), b(\theta), b(v))$			
T	Physical State Transition Model	$S \times \mathbf{A}^R \times S \longrightarrow [0, 1]$			
π	Agent Policy	$X^R \times \mathbf{A}^R \longrightarrow [0, 1]$			
$\tau \in \mathcal{T}$	Group motion plan	$\mathcal{T} = (\mathbf{A}^R \times O)^*$			
		τ_i means the i -th scout’s plan. $\tau_i \in \tau$.			

follow:

$$\begin{aligned} \operatorname{argmax}_{\tau} \mathbb{E}_{s \sim b(s), \theta \sim b(\theta)} [\theta^T f(\tau, s)] &= \operatorname{argmax}_{\tau} \mathbb{E}_{s \sim b(s)} [f(\tau, s)]^T \mathbb{E}_{\theta \sim b(\theta)} [\theta] \\ &\approx \operatorname{argmax}_{\tau} \bar{\theta}^T \left(\frac{1}{N_S} \sum_{n=1}^{N_S} f(\tau, s_n) \right) = \operatorname{argmax}_{\tau} \bar{\theta}^T \overline{f(\tau)}, \end{aligned} \quad (5.1)$$

where $f(\tau, s)$ is the fluent [NC36] when the game terminates given the state s and the scouts' plan τ , and the above equation takes the hard-max for plan selection. Given the dynamics of the game, f can be forward simulated in our planner, such that the expectation of $f(\tau, s)$ can be approximated using the Monte Carlo method with state samples. Instead of computing the full distribution, the agent only needs to keep track of the mean of the belief over human user's value function as we are using a linear model to calculate the gain of the game; we use $\bar{\theta}$ to denote the mean of $b(\theta)$. We can use the Boltzman rationality model to convert the planning problem described in Eq. (5.1) to a stochastic process, *i.e.*:

$$p(\tau; \bar{\theta}) = \frac{\exp(\beta_1 \bar{\theta}^T \overline{f(\tau)})}{\sum_{\tau' \in \mathcal{T}} \exp(\beta_1 \bar{\theta}^T \overline{f(\tau')})}, \quad (5.2)$$

where $\beta_1 \geq 0$. This conversion facilitates the inference of the human user's value function by enabling gradient-based optimization methods to learn $\bar{\theta}$. After a plan τ is determined, the joint action of all robot scouts is the first action of the plan, $\mathbf{a}^R = (\tau_1[0], \dots, \tau_K[0])$.

5.2.1.2 Value Function Estimation by Modeling ToM

Since the human user's value function is unknown to the scouts and has to be learned through interaction, our problem setting poses an additional challenge for classic POMDP solvers. To estimate the human user's value function during the communication, we integrated ToM into our computation model and developed a closed-form learning algorithm. Our algorithm leverages the assumption that, given a cooperative human user, the accepted plans are more likely to have a performance advantage over the rejected ones.

Belief Update with Level-1 ToM We use $m^H(fb)$ to denote the human user's feedback, which is a binary code with the i -th bit indicating the acceptance or rejection of the proposal

from the i -th scout. Assuming the human user is following the above decision-making process, the likelihood function of human user's feedback is:

$$p(m^H(fb)|\tau; \bar{\theta}) = \prod_{i=1}^K p(\tau_i; \bar{\theta})^{m^H(fb)_i} (1 - p(\tau_i; \bar{\theta}))^{(1-m^H(fb)_i)}, \quad (5.3)$$

where $p(\tau_i; \bar{\theta}) = \sum_{\tau \in \mathcal{T}, \tau_i \in \tau} p(\tau; \bar{\theta})$. Given this likelihood function, we can learn the mean of the parameter of value function $\bar{\theta}$, following the MLE derivation by maximizing $\log p(m^H(fb)|\tau; \bar{\theta})$ w.r.t. $\bar{\theta}$. Since $\bar{\theta} > 0$ and $\|\bar{\theta}\|_1 = 1$, this MLE process can be calculated by the projected stochastic gradient ascent algorithm [Nes03]. Hence, we have a closed-form derivation for $\partial \log p(m^H(fb)|\hat{\tau}; \bar{\theta}) / \partial \bar{\theta}$:

$$\begin{aligned} \frac{\partial \log p(m^H(fb)|\hat{\tau}; \bar{\theta})}{\partial \bar{\theta}} = & \beta_1 \sum_{i=1}^K \left[\mathbf{1}(m^H(fb)_i = 1) \left(\sum_{\tau \in \mathcal{T}, \hat{\tau}_i \in \tau} \frac{\exp(\beta_1 \bar{\theta}^T \overline{f(\tau)})}{\sum_{\tau' \in \mathcal{T}, \hat{\tau}_i \in \tau'} \exp(\beta_1 \bar{\theta}^T \overline{f(\tau')})} \overline{f(\tau)} \right) + \right. \\ & \left. \mathbf{1}(m^H(fb)_i = 0) \left(\sum_{\tau \in \mathcal{T}, \hat{\tau}_i \notin \tau} \frac{\exp(\beta_1 \bar{\theta}^T \overline{f(\tau)})}{\sum_{\tau' \in \mathcal{T}, \hat{\tau}_i \notin \tau'} \exp(\beta_1 \bar{\theta}^T \overline{f(\tau')})} \overline{f(\tau)} \right) - \mathbb{E}_{\tau \sim p(\tau; \bar{\theta})} [\overline{f(\tau)}] \right], \end{aligned} \quad (5.4)$$

where the two indicator functions select which summation to take conditioned on the feedback of the i -th proposal. The summation over weighted fluents, despite the overwhelming form, can be interpreted as the expected fluents in accord to the accepted/rejected plans. Hence, the intuition of this gradient is the difference between the expected fluents from plans without the accept/rejected proposals and the expected fluents from all the plans.

Belief Update with Level-2 ToM The above belief update mechanism assumes the human user will provide feedback to the proposals based on the intrinsic value of the proposals, *i.e.*, the expected return of the proposed plans given the underlying parameters of the value function. However, this is unlikely to be the case, as completely rational agents do not exist. Thus, we need to properly model level-2 ToM: With the explanation generated by the XAI system (see Sec. 5.2.1.3 for details), we further assume that the human user will be cooperative and provide feedback to best accelerate the parameter learning. Suppose the human

user provides feedback based on the improvement brought by the feedback, we have

$$q(m^H(fb)|\theta^*, \bar{\theta}, \tau) = \frac{\exp(-\beta_2 \|\bar{\theta} + \eta_t \frac{\partial \log p(m^H(fb)|\tau; \bar{\theta})}{\partial \theta} - \theta^*\|^2)}{\sum_{m^H(fb) \in FB} \exp(-\beta_2 \|\bar{\theta} + \eta_t \frac{\partial \log p(m^H(fb)|\tau; \bar{\theta})}{\partial \theta} - \theta^*\|^2)}, \quad (5.5)$$

where $\beta_2 \geq 0$ controls the extremeness of the softmin, η_t is the learning rate at time t , and θ^* is the ground-truth parameters of the value function possessed by the human user. The intuition of this equation is: The feedback from the human user is sampled from a softmin distribution of the distance between the updated parameters given the feedback and the ground-truth parameters. The smaller the distance is, the larger the improvement brought by that feedback, and the larger the improvement is, the more likely the feedback is provided. Further analysis of the above distance can be found in [LDL18]. Here, we use a softmin instead of hardmin in the data selection process. Integrating this feedback function into our parameter learning algorithm, we can derive a new parameter update function:

$$\bar{\theta}^{t+1} = \bar{\theta}^t + \eta_t g(m^H(fb)) + 2\beta_2 \eta_t^2 \left(g(m^H(fb)) - \mathbb{E}_{m(fb) \sim q(\theta^*, \bar{\theta}^t, \tau)} [g(m(fb))] \right), \quad (5.6)$$

where $g(m(fb)) = \frac{\log p(m(fb)|\tau; \bar{\theta}^t)}{\partial \theta}$. The first two terms are the same as the level-1 belief update, whereas the third term grasps the message's context by comparing the selected message against the also-runs and leverages the advantage to further update the belief. Notice that θ^* is unknown to the agent, so q in the expectation dose not have an exact solution. Thus, we use $\bar{\theta}^t + \eta_t g(m^H(fb))$ as an approximation of θ^* . That is, we first calculate level-1 ToM update on the parameters of the value function, then we take an additional gradient ascent step for level-2 ToM update.

Proposal Generation The XAI system generates proposals in accord to the change of expected belief. At each step, the agent first computes a new $\bar{\theta}'_m$ for each $m \in M^H$. Next, the change of expected belief can be calculated by $\delta(\tau, \bar{\theta}) = \mathbb{E}_{m \sim p(m^H|\tau, \bar{\theta})} [\|\bar{\theta}'_m - \bar{\theta}\|_2]$ for each $\tau \in \mathcal{T}$. If $\max_{\tau \in \mathcal{T}} \delta(\tau, \bar{\theta})$ surpasses a given threshold, the robot scouts make a proposal with $\operatorname{argmax}_{\tau} \delta(\tau, \bar{\theta})$. This formulation is generic; we can also substitute in other measurement

(*e.g.*, the expected variance of the $\bar{\theta}'$) in terms of the change of expected belief to generate more diverse update.

5.2.1.3 Explanation Generation by Modeling Mental Utility

We generate explanations whenever proposals are generated to facilitate the human user to make decisions. Given inputs produced by the executor, the explainer aims to generate human-like paragraphs that not only provide sufficient information but also match the human user’s preference of language use, *i.e.*, the mental utility.

Formally, an explanation is defined by its semantic inputs and a set of syntactic rules. The former is to provide explanations regarding *what*, including the current observation o , physical state s , and belief over the value function $b(\theta)$. The latter is to provide explanations regarding *how*. The explainer model is to determine the optimal syntax that matches the human user’s mental utility. Specifically, we pre-define a set of attributed templates; each is labeled with a set of distinguishable attributes. At each step, the explainer predicts the human user’s most favorable attributes based on the satisfactory score. We propose a sequential explanation generation model capable of adopting the temporal dynamics of the human’s mental state; it defines utility functions to synthesize the most efficient and suitable explanations.

Sequential Explanation Generation At time step t , the explainer takes in a tuple $h_t = \{(m_{t-1}^E, ss_{t-1}, o_t)\}$ as input, where $m_{t-1}^E \in M^E$ is the explanation of previous round $t - 1$, $ss_{t-1} \in SS$ is user’s satisfactory score estimated by human user’s feedback, and $o_t \in O$ is the current observation. Given the sequential input history $H_t = \{h_k, k = 1, \dots, t\}$, the explainer is to generate a new explanation m_t^E that maximizes the expected score:

$$m_t^E = \operatorname{argmax}_{m^E \in M^E} \mathbb{E}_{ss \sim p(ss|H_t)} [\hat{ss}(a^E)] - \lambda_c \operatorname{cost}(m^E), \quad (5.7)$$

where $a^E \in A^E$ is an extracted attribute vector of m^E , $\text{cost}(\cdot)$ a pre-defined cost function, and λ_c a constant factor.

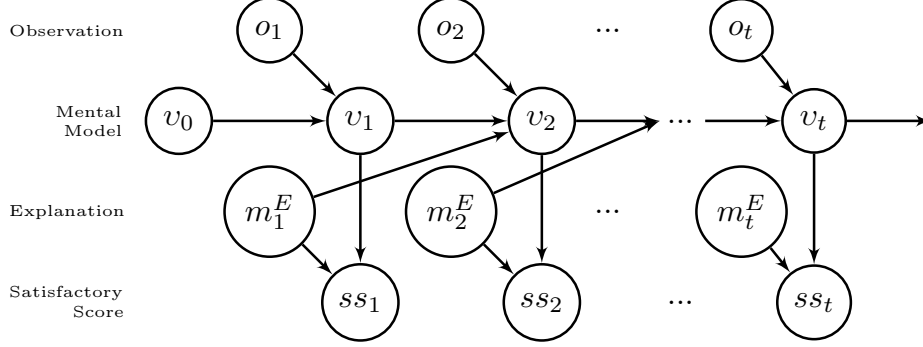


Figure 5.2: Temporal Evolution of Explanation Generation.

Algorithm 4 Explanation Generation with Hidden Mental Utilities

Input: *templates* - all explanation templates

Output: $\{m_1^E, m_2^E, \dots\}$

- 1: $t \leftarrow 1$
 - 2: **while** *not stopped* **do**
 - 3: *all_explanations* \leftarrow FillSlots(*templates*)
 - 4: Get O_t, ss_{t-1} from agent
 - 5: $m^E \leftarrow \text{None}$
 - 6: **for** m_i^E **in** *all_explanations* **do**
 - 7: $a^E \leftarrow \text{ExtractAttribute}(m_i^E)$
 - 8: Compute $\mathbb{E}[\hat{ss}(a^E)]$ according to Eq. (5.11).
 - 9: $m^E \leftarrow \text{argmax}_{\{m^E, m_i^E\}} \mathbb{E}[\hat{ss}(a^E)] - \text{cost}(m^E)$
 - 10: **end for**
 - 11: $m_t^E \leftarrow m^E, t \leftarrow t + 1$
 - 12: **end while**
-

We model the process of computing $\mathbb{E}[\hat{s}s(a^E)]$ as a HMM by introducing a mental state variable $v \in \Upsilon$, which corresponds to the human user’s mental utility of the explanation; see Fig. 5.2 for the graphical illustration of the computing process. At time step t ,

$$\begin{aligned} \mathbb{E}_{\hat{s}s \sim p(ss|H_t)}[\hat{s}s(a^E)] &= \sum_{ss \in SS} p(ss|a^E, ss_{1:t-1}, o_{1:t}, a_{1:t-1}^E)ss \\ &= \sum_{ss \in SS} \left(\sum_{v_t \in \Upsilon} p(v_t|ss_{1:t-1}, a_{1:t-1}^E, o_{1:t})p(ss|v_t, a^E) \right) ss. \end{aligned} \quad (5.8)$$

Let $\mathcal{K}(a_{t-1}^E, o_t) = p(v_t|v_{t-1}, a_{t-1}^E, o_t)$ be the transition matrix that depicts the transition probability from v_{t-1} to v_t , and $\mathcal{F}(a^E) = p(ss|v_t, a^E)$ be the score function that models the distribution of satisfactory score. We have:

$$p(v_t|ss_{1:t-1}, e_{1:t-1}, o_{1:t}) = \sum_{v_{t-1} \in \Upsilon} p(v_{t-1}|ss_{1:t-1}, a_{1:t-1}^E, o_{1:t-1})\mathcal{K}(a_{t-1}^E, o_t), \quad (5.9)$$

where $p(v_t|ss_{1:t}, a_{1:t}^E, o_{1:t}) = \alpha_t$ is computed by an iterative process:

$$\begin{aligned} p(v_t|ss_{1:t}, a_{1:t}^E, o_{1:t}) &\propto \mathcal{F}(a_t^E) \odot (\mathcal{K}(a_{t-1}^E, o_t)^T p(v_{t-1}|ss_{1:t-1}, a_{1:t-1}^E, o_{1:t-1})) \\ &= \mathcal{F}(a_t^E) \odot (\mathcal{K}(a_{t-1}^E, o_t)^T \alpha_{t-1}), \end{aligned} \quad (5.10)$$

where \odot is an element-wise product operator. Therefore, Eq. (5.8) can be written as

$$\mathbb{E}_{\hat{s}s \sim p(ss|H_t)}[\hat{s}s(a^E)] = \sum_{ss \in SS} \frac{ss}{Z} \alpha_t^T \mathcal{K}(a_{t-1}^E, o_t) \mathcal{F}(e), \quad (5.11)$$

where Z is a normalization constant of $p(ss|H_t)$; see Algorithm 4 for the computational flow.

5.2.1.4 Explanation with Ontogenetic Ritualization

Literature in evolutionary anthropology demonstrates strong evidence that early infants learn to communicate, especially in a symbolic manner, not based on imitation but rather on an individual learning process termed *ontogenetic ritualization* [MN12, Tom10, Loc80]. [TC97] argue such communicative behavior as a communicative signal that can be formed by two individuals shaping each other’s behavior in repeated instances of interaction over

time. Similar phenomena have also been observed and investigated on other primates, such as great apes [HRT13, Tom96]. For example, many individual chimpanzees come to use a stylized “arm-raise” to indicate that they are about to hit the other and thus initiate play [TC97]. In this way, a behavior that was not at first a communicative signal would become one over time. Generally, we follow [TZ02] to define the process of ontogenetic ritualization: (i) individual A performs behavior X ; (ii) individual B reacts consistently with behavior Y ; (iii) based on the initial steps of X , B anticipates A ’s performance of X , and hence performs Y ; and finally, (iv) A anticipates B ’s anticipation of X , and hence produces X in ritualized form so as to elicit Y .

We argue that the process of ontogenetic ritualization can also be formed during human-robot teaming, specifically when understanding and reacting to explanations. To achieve this goal, we set the “ritualized form” as a subset of explanation attributes A^E . As such, robot scouts can generate ritualized explanation based on their anticipation of human, *i.e.* $\mathbb{E}[\hat{s}s(a^E)]$.

5.2.2 Participants Description

Participants for this study were recruited from the online Prolific user research platform. Participants were selected based on their location (in the United States), their highest level of education (at least a bachelor’s degree), and the device they were using (no mobile users were selected). This choice was made to confine our participants to a population that is more likely to understand the nuance of the game while maintaining a broad pool of participants that is representative of the general population. A desktop/laptop computer was required to interact with the game appropriately. Information on the participants’ computer was collected (*i.e.* User-Agent). No other demographic information from the participant was collected after passing our demographic selection criteria.

After participants finished introductory material, a 7-question familiarity test was given to participants before proceeding into the game. This check was to make sure partici-

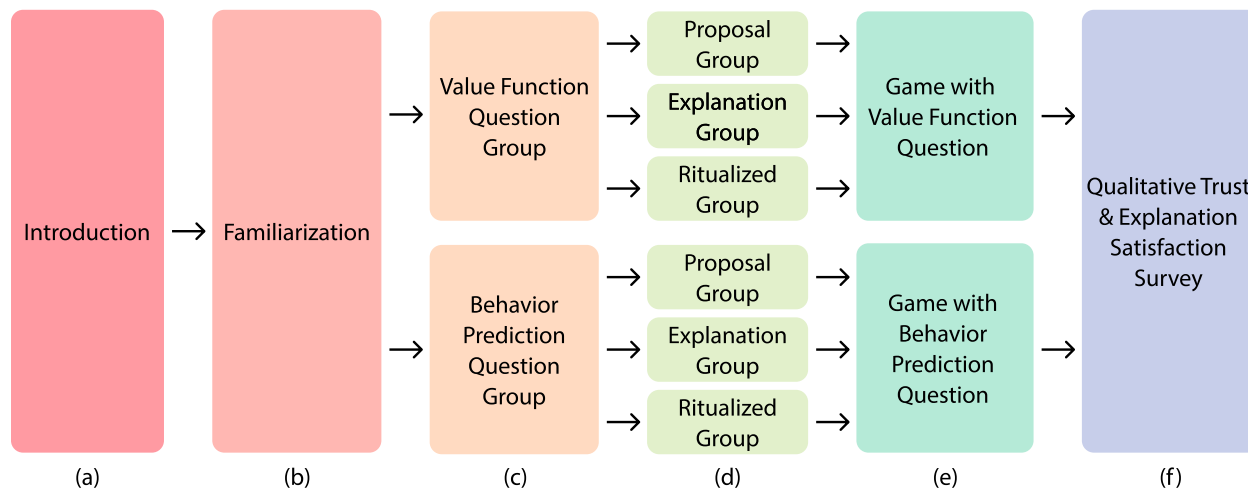


Figure 5.3: User study flow.

participants understood the rules of the game, what their objectives were, how to interpret value functions, and the distinction between explanations and proposals. Participants passed the questionnaire if they answered every question correctly. If a participant missed a question, a page was shown to explain the correct answer. Participants that missed a question had to repeat the entire questionnaire, and participants that failed to pass the questionnaire twice were removed from the study.

Participants were assigned randomly to each condition and were balanced automatically by our survey platform (Qualtrics). Compensation started at \$10 USD per participant, and our scoring system incentivizes participants to score as many points as possible. Participants received \$0.05 USD per point in the game, with a maximum total payout of \$20 USD per participant.

5.2.3 Study Design/Procedure/Measurement

The study was conducted in a between-subject design. Participants were randomized in a hierarchical group selection process: an outer hierarchy and an inner hierarchy. Fig. 5.3 illustrates the basic user study flow: (a) Participants begin with an introduction to explain

the setting and define key terms. (b) Participants are then familiarized with the game interfaces, and a questionnaire is given to verify participants understand the game. Participants that did not pass the familiarization were removed from the study. (c) Participants are randomly split into two groups: a group that is asked to infer the robot scout’s current value function and a group that is asked to predict the robot scout’s next behavior. This is done in a between-subject design. (d) Participants are further randomly split to receive different forms of explanations: proposals, explanations, and ritualized explanations. This is done in a between-subject design. (e) The participants then play the game and are asked the question assigned to their group throughout the experiment. (f) After finishing the game, participants were asked qualitative trust and explanation satisfaction questions.

The outer hierarchy was randomized to evenly split participants based on mental model questions: (i) value function and (ii) behavior prediction. The inner hierarchy was randomized to evenly assign participants based on different explanation formats: (i) a proposal group, (ii) an explanation group, and (iii) a ritualized explanation group. Among three groups, the robot scouts will follow the exact same action policy, π , and belief update process, BU . The groups differ only by the explanations forms received by the human user, λ_R , and the question about the robot scouts’ plan (current *vs* next round).

Our study includes four variables. The only independent variable is the form of the explanation a participant received: proposal, explanation, or ritualized explanation. Three dependent variables are (i) value function alignment, (ii) behavior prediction, and (iii) qualitative trust and explanation satisfaction. To ensure no confounding on order or effects from answering a question regarding value function prediction and behavior prediction, the study was designed with the above outer hierarchy using a between-subject design. Participants from all groups were asked to provide qualitative trust and explanation satisfaction at the end of the study.

5.2.4 Instruments/Materials

We first outline the flow of the experiment to give a high-level overview of the participant's experience through the game, as shown in Fig. 5.3. The figure shows our between-subject design across our two mental model questions (value function and behavior prediction) and our explanation formats (proposal, explanation, and ritualized).

The introduction phase of the experiment introduces the basic background of the game. The introduction outlines that the participant is a commander in charge of finding a path from the lower-right-hand corner of the map to the upper-left-hand corner of the map. The introduction outlines that the area may have dangerous devices, such as bombs, along the path. The participant is told they have a team of robot scouts to help explore the area, and that the scouts will provide proposals (and in explanation groups, explanations).

During familiarization, the participant is instructed that while their objective is to get to the upper-left corner, they are also instructed there are sub-goals the team would benefit from achieving. These sub-goals consist of time, area explored, number of bombs investigated, and the resources collected. Participants are then informed that these goals are specific to the team's current circumstance and the value function conveys the relative importance of these sub-goals. Participants are informed they will be scored, and this score is weighted by the value function. Finally, the robot scout proposals and explanations are described. Proposals correspond to robot scout plans, and explanations attempt to justify those plans. These various components are introduced using figures of the panels shown in Fig. 5.4. Moving from left to right, the Legend panel displays a permanent legend for the user to refer to understand different tile types. The Value Function panel shows the current value function of the user's team, is unknown to the robot scouts, and cannot be modified by the user. The central map shows the current information on the map. The Score panel shows the user's current score and the individual fluent functions that contribute to the score. The overall score is calculated as the normalized, value function-weighted sum of the individual fluent

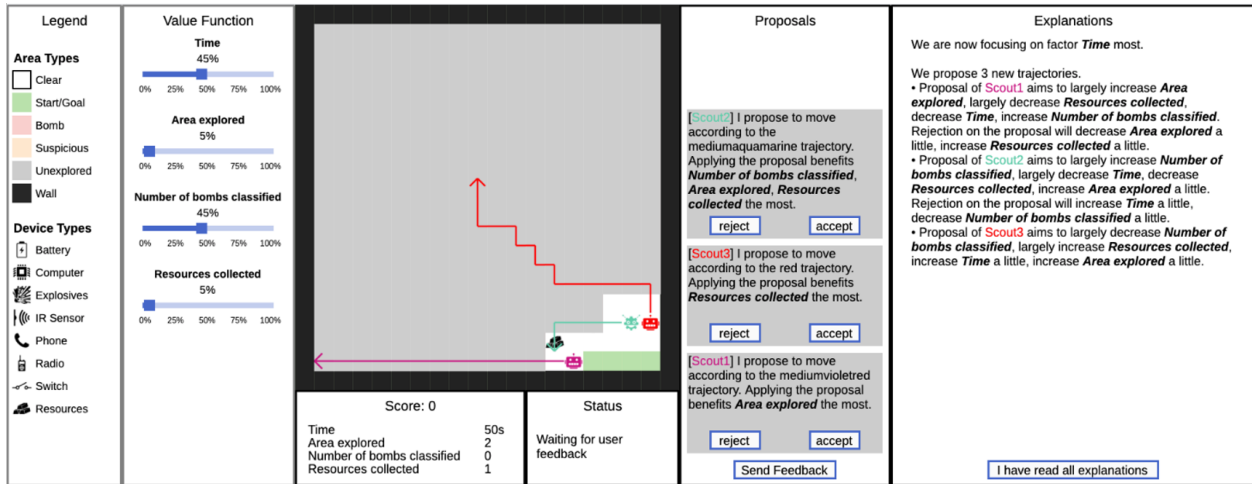


Figure 5.4: User interface of the scout exploration game.

function scores. The Status panel displays the current status of the system. The Proposal panel shows the robot scouts’ current proposals, and the user can accept/reject each. The Explanation panel shows explanations provided by the scouts.

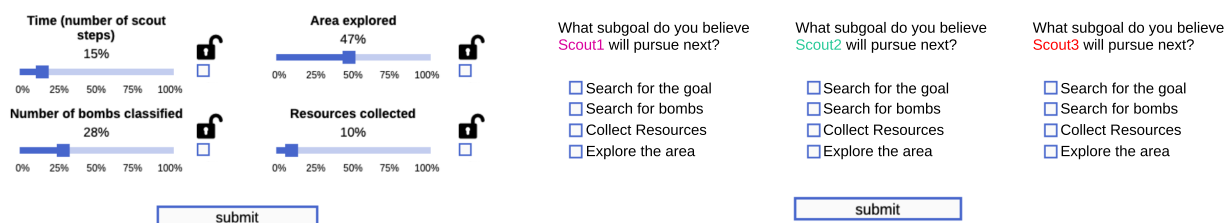
To test the participant’s understanding of the background, at the end of the familiarization phase, the participant is given several attention-check questions at the end of familiarization. For example, we asked “for each trial, you will be given a value function that describes the team’s current mission priorities” (correct answer: true). Participants who answered a question incorrectly would receive instruction as to why the correct answer is correct and would be required to repeat the attention-check. Participants who failed the attention-check twice did not further participate in the study.

We implemented this game using HaxeFlixel, a 2D Game Engine used to create JavaScript-based games. Users can access the game on web browsers. The full user interface of the game is displayed in Fig. 5.4. Throughout the study, the user monitors the progress of the team, receives explanations, and gives feedback by accepting or rejecting the proposals. The team’s performance is quantified as a score, which reflects how well the scouts can estimate the user’s value function and act accordingly. Participants are instructed to maximize the team’s score. The score is weighted by the value function to appropriately score the relative

importance of sub-goals. Each sub-goal score is computed from the environment’s reward function.

During the game, the robot scouts attempt to infer the human value function. To infer the correct value function, the robot team proposes action plans to the user and estimates the value function based on the user’s feedback. Explanations of the proposals will accompany to better clarify the motivation of the robot scouts. An example proposal is: “We can keep moving despite the suspicious area (proposal) if we want to find a path from A to B as soon as possible (explanation).” If the user accepts this proposal, the robots will increase the value of time in the value function. Otherwise, the robots will increase the value of investigating bombs and tile exploration but decrease the value of time. The game repeats in a loop, where robot scouts make proposals (and in some groups, explain), execute plans, and repropose until they find a path to the upper-left-hand corner of the map.

Our between-subject design is divided by the question type that will be asked during the experiment (value function or behavior prediction) or and by the explanation format displayed to the user (proposal, explanation, or ritualized). The question indicates what question users will be asked to infer before the robot scouts start the next round of explanation and proposing. The value function question asks participants to provide the value function *they believe* the robot scouts are using. Participants provide their rating by manipulating a set of sliders that are interdependent; the slides must always sum to 100%. This provides the relative importance of sub-goals. For the behavior prediction question, participants are asked to predict what proposal the robot scouts will make next. Note that this is a between-subject design, so participants will see one question but not the other. Fig. 5.5 illustrates examples of two question interface: (a) Users can slide the bars to set a relative importance of each sub-goal. The sub-goals must sum to 100%. As the user changes one slider, the others will automatically decrease to keep the sum at 100%. Users can lock a particular slider by checking the lock symbol to the right of the slider. (b) Users are asked to predict which sub-goal the robot scouts will pursue next. Users are asked to predict the



(a) Value function question interface

(b) Behavior prediction question interface

Figure 5.5: Example interfaces for the value function question and the behavior prediction question.

sub-goal for each scout individually; this is because proposals are generated on a per-scout basis.

For the explanation format displayed, participants in the proposal group will only see robot scout proposals (see Proposal Panel in Fig. 5.4) while participants in the explanation group will see robot scout proposals and explanations (see Explanation Panel in Fig. 5.4). The ritualized group is identical to the explanation group, except that robot scouts actively attempt to ritualize explanations based on the shared common mind between the robot scouts and the participant.

After the participants finish the game, they are directed to a post-experiment survey to evaluate qualitative trust and explanation satisfaction. Self-reported trust is evaluated using Likert-scale questions, which are designed based on Muir’s questionnaire [Mui94] and Madsen’s Human-Computer Trust Instrument [MG00]. The questionnaire intends to evaluate how the information given to the users across different groups helps them make appropriate decisions when they are asked to give feedback on the proposals. Such appropriate reliance [LS04] is supported by a correct understanding of multiple components of the system, including the planning, value function estimation, proposal generation, feedback interpretation, and/or explanation generation, which form the basis of trust. Specifically, the trust questionnaire is composed of questions that intend to evaluate the perceived reliability, technical

competence, and understand-ability of the scouts with respect to these components. We ask the users “how much would you trust the robot scouts to achieve a high score on their own, given they have the correct value function?” and “how much do you trust the robot scouts to learn the value function of another commander in another circumstance?”

Explanation satisfaction is evaluated in the aspects of *transparency*, *helpfulness* and *timeliness* via Likert-scale questions to reflect users’ feeling on how well the explanation has helped them understand these components and make correct feedback to guide the team towards plans that better suited to the scenario and value function given to the users.

5.2.5 Hypotheses

The hypotheses we are testing in this experiment are related to quantitative measures for mental model alignment and qualitative measures relating to trust and explanation satisfaction. The quantitative measures for mental model alignment include: (*H1*) value function alignment, (*H2*) behavior prediction, and (*H3*) user-machine task performance. For value function alignment (*H1*), we hypothesize that groups that have access to explanations will be more accurate inferring the current robot scout value function. For the behavior prediction (*H2*), we hypothesize that groups that have access to explanations will be more accurate in predicting what the robot scouts will do next. The user-machine task performance (*H3*) will be evaluated by the score participants receive from the game.

Our qualitative measures will assess trust by asking users whether they would trust the robot scouts to complete the task on their own, given they have the correct value function. Additionally, we will ask users whether or not they trust the scout to learn a different value function with a different commander. We hypothesize that groups that have access to richer explanations will rate the qualitative trust measures higher than those without (*H4*). Our second qualitative hypothesis (*H5*) is that groups with access to richer explanations will report higher degrees of explanation satisfaction.

5.3 Results

Here we outline how we plan on analyzing the results. Our primary measures of significance will be using a student's t -test and analysis of variance (ANOVA) using the F -test. We expect to see the following significance' for each hypothesis:

- $H1$: Significance between proposal and explanation group. Significance between proposal and ritualized explanation group. No significance between explanation and ritualized explanation group.
- $H2$: Significance between proposal and explanation group. Significance between proposal and ritualized explanation group. No significance between explanation and ritualized explanation group.
- $H3$: Significance between proposal and explanation group. Significance between proposal and ritualized explanation group. No significance between explanation and ritualized explanation group.
- $H4$: Significance between proposal and explanation group. Significance between proposal and ritualized explanation group. Significance between explanation and ritualized explanation group.
- $H5$: Significance between proposal and explanation group. Significance between proposal and ritualized explanation group. Significance between explanation and ritualized explanation group.

We believe for $H1$ and $H2$, we will observe significance between the proposal and explanation group. This is predominately due to the richer explanations providing a deeper insight into the robot scout's reasoning process, allowing better inference of the mental state of the robot scouts. The same applies between the proposal and ritualized explanation group. We do not expect to see a significance between the explanation and ritualized explanation for these hypotheses because these two forms of explanation convey similar information in different forms (ritualized being an abridged version of the full explanation based on the

common mind between the human and the robot scouts).

For $H3$ and $H4$, we believe to see significance between all groups. Between the proposal and explanation group, we believe the transparency and insight provided by the explanations will improve trust and satisfaction ratings. Furthermore, between the explanation and ritualized explanation group, we predict the ritualization will further improve trust and satisfaction ratings, as the ritualization conveys a deeper understanding of the shared common mind between the human and the robot scouts.

5.4 Summary and Conclusions

In this study, we looked at a unique XAI paradigm, namely a *active machine–active human* paradigm wherein both the machine and the human are active participants in the explanation process. This contrasts to more traditional XAI studies that use a *passive machine—active human* paradigm wherein the machine provides an explanation that a human user interprets, with no engagement from the human back to the machine. To achieve this paradigm, we adopt a communicative learning framework based on Theory of Mind (ToM) where the machine actively reasons about human users’ mental states. This communicative learning paradigm generates explanations that help build a *common mind* between the user and machine, thereby allowing the machine to perform the task better.

We constructed a Robot Scout Exploration Game, where a team of robot scouts explores a dangerous area, looking for a safe path for the commander’s team to cross the area. The team has sub-goals, such as minimizing the amount of time or investigating devices that may be bombs. The robot scouts provide information to the commander from their sensing capabilities, along with proposals on what the scouts plan to do next and explanations for those proposals. The commander can then accept or reject the proposals, thereby providing feedback to the scouts on the utility of a proposal. The robot scouts then use this feedback to estimate the commander’s intents and goals to improve future proposals and explanations.

This iterative communication process continues until the team completes the task (finding a safe path to reach the destination).

The user study presented here quantitatively assesses the degree to which different forms of explanation improve mental model understanding between the user and the machine and qualitatively assess user-machine trust and explanation satisfaction. We expect that access to richer forms of explanation will improve the mental model understanding and user-machine task performance. Additionally, we expect richer forms of explanations to foster more trust and improve explanation satisfaction. Of note, we anticipate that these scores will be the highest in a ritualized explanation group, where the machine shortens explanations as the user and machine establish a *common mind*.

This study aims to present an unexplored *active machine–active human* paradigm where both the human and the machine actively participate in the explanation process. We believe this opens a new venue for future interactive XAI studies that showcase collaborative environments between users and machines.

CHAPTER 6

Conclusion

In this dissertation, we show multimodal conversation modeling from three different perspectives – neural perception, structure learning, and communication. Specifically,

In Chapter 2, we learn the joint multimodal distribution by cooperative training of a fast thinking initializer and slow thinking solver. The initializer generates the output directly by a non-linear transformation of the input as well as a noise vector that accounts for latent variability in the output. The slow thinking solver learns an objective function in the form of a conditional energy function, so that the output can be generated by optimizing the objective function, or more rigorously by sampling from the conditional energy-based model. We demonstrate the effectiveness of the proposed method on various conditional learning tasks, *e.g.*, class-to-image generation, image-to-image translation, and image recovery, *etc.*

In Chapter 3, we explicitly formalize visual dialogue as inference in a graphical model with partially observed nodes and unknown graph structures (relations in dialogue). The given dialog entities are viewed as the observed nodes. The answer to a given question is represented by a node with missing value. We first introduce an Expectation Maximization (EM) algorithm to infer both the underlying dialogue structures and the missing node values (desired answers). Based on this, we proceed to propose a differentiable graph neural network (GNN) solution that approximates this process.

In Chapter 4, we present a grammar-based dialogue dataset, GRICE, designed to bring implicature into pragmatic reasoning in the context of conversations. Our design of GRICE also incorporates other essential aspects of modern dialogue modeling (*e.g.*, coreference).

The entire dataset is systematically generated using a hierarchical grammar model, such that each dialogue context has intricate implicatures and is temporally consistent. We further present two tasks, the implicature recovery task followed by the pragmatic reasoning task in conversation, to evaluate the model’s reasoning capability.

In Chapter 5, we present a human robot collaboration task – bomb defusing game, where the explanation is serving as the communication media with human users. We propose to model the explanation as the posterior distribution of human’s hidden mental utilities and observations. We solve the explanation generation by using forwarding algorithm of hidden markov model. We show that our model is able to generate more diverse explanations compared with expert systems.

This dissertation aims at making some progress in the line of building explainable multimodal conversation models. However, there are still a lot of missing dimensions in conversation modeling, *e.g.* logic and reasoning. Moving forward, we call for future research explainable multimodal models to build conversational agents with human-like intelligence.

REFERENCES

- [AAL15] Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C. Lawrence Zitnick, and Devi Parikh. “VQA: Visual Question Answering.” In *ICCV*, 2015.
- [ACB17] Martin Arjovsky, Soumith Chintala, and Léon Bottou. “Wasserstein Generative Adversarial Networks.” In *International Conference on Machine Learning (ICML)*, pp. 214–223, 2017.
- [AHB18] Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. “Bottom-Up and Top-Down Attention for Image Captioning and Visual Question Answering.” In *CVPR*, 2018.
- [AN04] Pieter Abbeel and Andrew Y Ng. “Apprenticeship learning via inverse reinforcement learning.” In *International Conference on Machine Learning (ICML)*, pp. 1–8, 2004.
- [AWZ20] Arjun Akula, Shuai Wang, and Song-Chun Zhu. “CoCoX: Generating conceptual and counterfactual explanations via fault-lines.” In *Proceedings of AAAI Conference on Artificial Intelligence (AAAI)*, 2020.
- [Bat00] Gregory Bateson. *Steps to an ecology of mind: Collected essays in anthropology, psychiatry, evolution, and epistemology*. University of Chicago Press, 2000.
- [BCW17] Jianmin Bao, Dong Chen, Fang Wen, Houqiang Li, and Gang Hua. “CVAE-GAN: fine-grained image generation through asymmetric training.” In *IEEE International Conference on Computer Vision (ICCV)*, pp. 2745–2754, 2017.
- [BFZ18] Yalong Bai, Jianlong Fu, Tiejun Zhao, and Tao Mei. “Deep Attention Neural Tensor Network for Visual Question Answering.” In *ECCV*, 2018.
- [BHW79] Robert H Bonczek, Clyde W Holsapple, and Andrew B Whinston. “Computer-based support of organizational decision making.” *Decision Sciences*, **10**(2):268–291, 1979.
- [BHW81] Robert H Bonczek, Clyde W Holsapple, and Andrew B Whinston. “A generalized decision support system using predicate calculus and network data base management.” *Operations Research*, **29**(2):263–281, 1981.
- [BJL17] Piotr Bojanowski, Armand Joulin, David Lopez-Paz, and Arthur Szlam. “Optimizing the latent space of generative networks.” In *International Conference on Machine Learning (ICML)*, pp. 599–608, 2017.
- [Bor09] Emma Borg. “On three theories of implicature: Default theory, relevance theory and minimalism.” *International Review of Pragmatics*, **1**(1):63–83, 2009.

- [BPL16] Peter W Battaglia, Razvan Pascanu, Matthew Lai, Danilo Rezende, and Koray Kavukcuoglu. “Interaction networks for learning about objects, relations and physics.” In *NeurIPS*, 2016.
- [BSM17] David Berthelot, Thomas Schumm, and Luke Metz. “Began: Boundary equilibrium generative adversarial networks.” *arXiv preprint arXiv:1703.10717*, 2017.
- [BZ20] Adrian Barbu and Song-Chun Zhu. *Monte Carlo Methods*. Springer, 2020.
- [CDH16] Xi Chen, Yan Duan, Rein Houthoofd, John Schulman, Ilya Sutskever, and Pieter Abbeel. “InfoGAN: Interpretable representation learning by information maximizing generative adversarial nets.” In *Advances in Neural Information Processing Systems (NIPS)*, pp. 2172–2180, 2016.
- [CJS18] Fuhai Chen, Rongrong Ji, Xiaoshuai Sun, Yongjian Wu, and Jinsong Su. “GroupCap: Group-Based Image Captioning With Structured Relevance and Diversity Constraints.” In *CVPR*, 2018.
- [CMG14] Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. “Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation.” In *EMNLP*, 2014.
- [COW16] Xiao Chu, Wanli Ouyang, Xiaogang Wang, et al. “CRF-CNN: Modeling structured information in human pose estimation.” In *NeurIPS*, 2016.
- [CSK20] Tathagata Chakraborti, Sarath Sreedharan, and Subbarao Kambhampati. “The emerging landscape of explainable automated planning & decision making.” In *Proceedings of International Joint Conference on Artificial Intelligence (IJCAI)*, 2020.
- [CWL20] Leyang Cui, Yu Wu, Shujie Liu, Yue Zhang, and Ming Zhou. “MuTual: A Dataset for Multi-Turn Dialogue Reasoning.” In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, 2020.
- [CZC17] Henry Y Chen, Ethan Zhou, and Jinho D Choi. “Robust Coreference Resolution and Entity Linking on Dialogues: Character Identification on TV Show Transcripts.” In *Proceedings of the Conference on Computational Natural Language Learning (CoNLL)*, 2017.
- [CZY18] Tianlang Chen, Zhongping Zhang, Quanzeng You, Chen Fang, Zhaowen Wang, Hailin Jin, and Jiebo Luo. ““Factual” or “Emotional”: Stylized Image Captioning with Adaptive Learning and Attention.” In *ECCV*, 2018.
- [Dav16] Wayne A Davis. “Implicature.” In *Irregular Negatives, Implicatures, and Idioms*, pp. 51–84. Springer, 2016.

- [DB13] Gerard De Melo and Mohit Bansal. “Good, great, excellent: Global inference of semantic intensities.” *Transactions of the Association for Computational Linguistics*, 1:279–290, 2013.
- [DBP17] Vincent Dumoulin, Ishmael Belghazi, Ben Poole, Alex Lamb, Martín Arjovsky, Olivier Mastropietro, and Aaron C. Courville. “Adversarially Learned Inference.” In *International Conference on Learning Representations (ICLR)*, 2017.
- [DBV16] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. “Convolutional neural networks on graphs with fast localized spectral filtering.” In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, 2016.
- [DCF15] Emily L Denton, Soumith Chintala, and Rob Fergus. “Deep Generative Image Models using a Laplacian Pyramid of Adversarial Networks.” In *Advances in Neural Information Processing Systems (NIPS)*, pp. 1486–1494, 2015.
- [DCL18] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. “Bert: Pre-training of deep bidirectional transformers for language understanding.” *arXiv preprint arXiv:1810.04805*, 2018.
- [DKB14] Laurent Dinh, David Krueger, and Yoshua Bengio. “Nice: Non-linear independent components estimation.” *arXiv preprint arXiv:1410.8516*, 2014.
- [DKG17] Abhishek Das, Satwik Kottur, Khushi Gupta, Avi Singh, Deshraj Yadav, José MF Moura, Devi Parikh, and Dhruv Batra. “Visual dialog.” In *CVPR*, 2017.
- [DKM17] Abhishek Das, Satwik Kottur, José MF Moura, Stefan Lee, and Dhruv Batra. “Learning Cooperative Visual Dialog Agents with Deep Reinforcement Learning.” In *ICCV*, 2017.
- [DMI15] David K Duvenaud, Dougal Maclaurin, Jorge Iparraguirre, Rafael Bombarell, Timothy Hirzel, Alán Aspuru-Guzik, and Ryan P Adams. “Convolutional networks on graphs for learning molecular fingerprints.” In *NeurIPS*, 2015.
- [DR95] Robert Dale and Ehud Reiter. “Computational interpretations of the Gricean maxims in the generation of referring expressions.” *Cognitive science*, 19(2):233–263, 1995.
- [DSB17] Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. “Density estimation using Real NVP.” In *International Conference on Learning Representations (ICLR)*, 2017.
- [EGL19] Mark Edmonds, Feng Gao, Hangxin Liu, Xu Xie, Siyuan Qi, Brandon Rothrock, Yixin Zhu, Ying Nian Wu, Hongjing Lu, and Song-Chun Zhu. “A tale of two explanations: Enhancing human trust by explaining robot behavior.” *Science Robotics*, 4(37), 2019.

- [EMQ20] Mark Edmonds, Xiaojian Ma, Siyuan Qi, Yixin Zhu, Hongjing Lu, and Song-Chun Zhu. “Theory-based causal transfer: Integrating instance-level induction and abstract-level structure learning.” In *Proceedings of AAAI Conference on Artificial Intelligence (AAAI)*, 2020.
- [EQZ19] Mark Edmonds, Siyuan Qi, Yixin Zhu, James Kubricht, Song-Chun Zhu, and Hongjing Lu. “Decomposing human causal learning: bottom-up associative learning and top-down schema reasoning.” In *Proceedings of the Annual Meeting of the Cognitive Science Society (CogSci)*, 2019.
- [Esc96] Victor Escandell. *Introducción a la pragmática*. Ariel Linguística. española. 6ta, 1996.
- [Fet17] Anita Fetzer. “Context.” In *The Oxford handbook of pragmatics*. Oxford University Press, 2017.
- [FG12] Michael C Frank and Noah D Goodman. “Predicting pragmatic reasoning in language games.” *Science*, **336**(6084):998–998, 2012.
- [FXW18] Hao-Shu Fang, Yuanlu Xu, Wenguan Wang, Xiaobai Liu, and Song-Chun Zhu. “Learning pose grammar to encode human body configuration for 3D pose estimation.” In *AAAI*, 2018.
- [GA19] David Gunning and David Aha. “DARPA’s explainable artificial intelligence (XAI) program.” *AI Magazine*, **40**(2):44–58, 2019.
- [GAA17] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. “Improved training of wasserstein GANs.” In *Advances in Neural Information Processing Systems (NIPS)*, pp. 5767–5777, 2017.
- [GCS20] Aditya Grover, Christopher Chute, Rui Shu, Zhangjie Cao, and Stefano Ermon. “AlignFlow: Cycle Consistent Learning from Multiple Domains via Normalizing Flows.” In *The Thirty-Fourth AAAI Conference on Artificial Intelligence (AAAI)*, pp. 4028–4035, 2020.
- [GF16] Noah D Goodman and Michael C Frank. “Pragmatic language interpretation as probabilistic inference.” *Trends in cognitive sciences*, **20**(11):818–829, 2016.
- [GGH17] C. Gan, Z. Gan, X. He, J. Gao, and L. Deng. “StyleNet: Generating Attractive Visual Captions with Styles.” In *CVPR*, 2017.
- [GGZ20] Xiaofeng Gao, Ran Gong, Yizhou Zhao, Shu Wang, Tianmin Shu, and Song-Chun Zhu. “Joint Mind Modeling for Explanation Generation in Complex Human-Robot Collaborative Tasks.” In *Proceedings of International Symposium on Robot and Human Interactive Communication (RO-MAN)*, 2020.

- [GKS17] Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. “Making the V in VQA Matter: Elevating the Role of Image Understanding in Visual Question Answering.” In *CVPR*, 2017.
- [GLZ18] Ruiqi Gao, Yang Lu, Junpei Zhou, Song-Chun Zhu, and Ying Nian Wu. “Learning generative ConvNets via multi-grid modeling and sampling.” In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9155–9164, 2018.
- [GMS05] Marco Gori, Gabriele Monfardini, and Franco Scarselli. “A new model for learning in graph domains.” In *IJCNN*, 2005.
- [GMZ15] Haoyuan Gao, Junhua Mao, Jie Zhou, Zhiheng Huang, Lei Wang, and Wei Xu. “Are you talking to a machine? Dataset and methods for multilingual image question.” In *NeurIPS*, 2015.
- [GPM14] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. “Generative adversarial nets.” In *Advances in Neural Information Processing Systems*, pp. 2672–2680, 2014.
- [Gri75] Herbert P Grice. “Logic and conversation.” In *Speech acts*, pp. 41–58. Brill, 1975.
- [GS13] Noah D Goodman and Andreas Stuhlmüller. “Knowledge and implicature: Modeling language understanding as social cognition.” *Topics in cognitive science*, 5(1):173–184, 2013.
- [GSR17] Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. “Neural Message Passing for Quantum Chemistry.” In *ICML*, 2017.
- [GWH14] Yunchao Gong, Liwei Wang, Micah Hodosh, Julia Hockenmaier, and Svetlana Lazebnik. “Improving image-sentence embeddings using large weakly annotated photo collections.” In *ECCV*, 2014.
- [HE16] Jonathan Ho and Stefano Ermon. “Generative adversarial imitation learning.” In *Advances in Neural Information Processing Systems (NIPS)*, pp. 4565–4573, 2016.
- [Hin02] Geoffrey E Hinton. “Training products of experts by minimizing contrastive divergence.” *Neural Computation*, 14(8):1771–1800, 2002.
- [Hir85] Julia Linn Bell Hirschberg. *A theory of scalar implicature*. University of Pennsylvania, 1985.

- [HLM19] Yedid Hoshen, Ke Li, and Jitendra Malik. “Non-adversarial image synthesis with generative latent nearest neighbors.” In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5811–5819, 2019.
- [HRT13] Marta Halina, Federico Rossano, and Michael Tomasello. “The ontogenetic ritualization of bonobo gestures.” *Animal Cognition*, **16**(4):653–666, 2013.
- [HRU17] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. “GANs trained by a two time-scale update rule converge to a local nash equilibrium.” In *Advances in Neural Information Processing Systems (NIPS)*, pp. 6626–6637, 2017.
- [Hua17] Yan Huang. *The Oxford handbook of pragmatics*. Oxford University Press, 2017.
- [HW04] Laurence R Horn and Gregory L Ward. *The handbook of pragmatics*. Wiley Online Library, 2004.
- [HWS16] Mikael Henaff, Jason Weston, Arthur Szlam, Antoine Bordes, and Yann LeCun. “Tracking the world state with recurrent entity networks.” *arXiv preprint arXiv:1612.03969*, 2016.
- [HZH19] Mingqi Hu, Deyu Zhou, and Yulan He. “Variational Conditional GAN for Fine-grained Controllable Image Generation.” In *Asian Conference on Machine Learning (ACML)*, pp. 109–124, 2019.
- [IS15] Sergey Ioffe and Christian Szegedy. “Batch normalization: Accelerating deep network training by reducing internal covariate shift.” In *International Conference on Machine Learning (ICML)*, pp. 448–456, 2015.
- [IZZ17] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. “Image-to-image translation with conditional adversarial networks.” In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5967–5976, 2017.
- [JHM17] Justin Johnson, Bharath Hariharan, Laurens van der Maaten, Li Fei-Fei, C Lawrence Zitnick, and Ross Girshick. “CLEVR: A diagnostic dataset for compositional language and elementary visual reasoning.” In *CVPR*, 2017.
- [JJM16] Allan Jabri, Armand Joulin, and Laurens van der Maaten. “Revisiting visual question answering baselines.” In *ECCV*, 2016.
- [JKF16] Justin Johnson, Andrej Karpathy, and Li Fei-Fei. “Densecap: Fully convolutional localization networks for dense captioning.” In *CVPR*, 2016.
- [JLS18] Unnat Jain, Svetlana Lazebnik, and Alexander G Schwing. “Two can play this game: Visual dialog with discriminative question generation and answering.” In *CVPR*, 2018.

- [JWB20] Paloma Jeretic, Alex Warstadt, Suvrat Bhooshan, and Adina Williams. “Are Natural Language Inference Models IMPPRESsive? Learning IMPLICature and PRESupposition.” In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, 2020.
- [KB15] Diederik P Kingma and Jimmy Ba. “Adam: A method for stochastic optimization.” In *International Conference on Learning Representations (ICLR)*, 2015.
- [KF15] Andrej Karpathy and Li Fei-Fei. “Deep visual-semantic alignments for generating image descriptions.” In *CVPR*, 2015.
- [KFW17] Thomas Kipf, Ethan Fetaya, Kuan-Chieh Wang, Max Welling, and Richard Zemel. “Neural relational inference for interacting systems.” In *ICML*, 2017.
- [KMP18a] Satwik Kottur, Jose M. F. Moura, Devi Parikh, Dhruv Batra, and Marcus Rohrbach. “Visual Coreference Resolution in Visual Dialog using Neural Module Networks.” In *ECCV*, 2018.
- [KMP18b] Satwik Kottur, José M. F. Moura, Devi Parikh, Dhruv Batra, and Marcus Rohrbach. “Visual Coreference Resolution in Visual Dialog using Neural Module Networks.” In *Proceedings of European Conference on Computer Vision (ECCV)*, 2018.
- [KP20] Kapa Korta and John Perry. “Pragmatics.” In Edward N. Zalta, editor, *The Stanford Encyclopedia of Philosophy*. Metaphysics Research Lab, Stanford University, spring 2020 edition, 2020.
- [Kri09] Alex Krizhevsky. “Learning multiple layers of features from tiny images.” Technical report, University of Toronto, 2009.
- [KSH12] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. “Imagenet classification with deep convolutional neural networks.” In *Advances in Neural Information Processing Systems (NIPS)*, pp. 1097–1105, 2012.
- [KW14] Diederik P. Kingma and Max Welling. “Auto-Encoding Variational Bayes.” In *International Conference on Learning Representations (ICLR)*, 2014.
- [KW17] Thomas N Kipf and Max Welling. “Semi-supervised classification with graph convolutional networks.” In *International Conference on Learning Representations (ICLR)*, 2017.
- [LBB98] Yann LeCun, Léon Bottou, Yoshua Bengio, Patrick Haffner, et al. “Gradient-based learning applied to document recognition.” *Proceedings of the IEEE*, **86**(11):2278–2324, 1998.

- [LDL18] Weiyang Liu, Bo Dai, Xingguo Li, Zhen Liu, James Rehg, and Le Song. “Towards black-box iterative machine teaching.” In *Proceedings of International Conference on Machine Learning (ICML)*, 2018.
- [LDM12] Hector Levesque, Ernest Davis, and Leora Morgenstern. “The Winograd schema challenge.” In *International Conference on the Principles of Knowledge Representation and Reasoning*, 2012.
- [Lev85] Stephen C Levinson. *Pragmatics*. Cambridge Univ. Press, 1985.
- [Lev95] Stephen C Levinson. “Interactional biases in human thinking.” In *Social intelligence and interaction*, pp. 221–260. Cambridge University Press, 1995.
- [LHH20] Qing Li, Siyuan Huang, Yining Hong, and Song-Chun Zhu. “A Competence-aware Curriculum for Visual Concepts Learning via Question Answering.” *Proceedings of European Conference on Computer Vision (ECCV)*, 2020.
- [Liu08] Jun S Liu. *Monte Carlo strategies in scientific computing*. Springer, 2008.
- [LKY17] Jiasen Lu, Anitha Kannan, Jianwei Yang, Devi Parikh, and Dhruv Batra. “Best of both worlds: Transferring knowledge from discriminative learning to a generative visual dialog model.” In *NeurIPS*, 2017.
- [LLG19] Rui Liu, Yu Liu, Xinyu Gong, Xiaogang Wang, and Hongsheng Li. “Conditional adversarial generative flow for controllable image synthesis.” In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7992–8001, 2019.
- [LLL15] Ziwei Liu, Xiaoxiao Li, Ping Luo, Chen-Change Loy, and Xiaoou Tang. “Semantic image segmentation via deep parsing network.” In *ICCV*, 2015.
- [LMB14] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. “Microsoft coco: Common objects in context.” In *ECCV*, 2014.
- [Loc80] A. Lock. *The guided reinvention of language*. Academic Pr, 1980.
- [LPC18] Ruotian Luo, Brian Price, Scott Cohen, and Gregory Shakhnarovich. “Discriminability Objective for Training Descriptive Captions.” In *CVPR*, 2018.
- [LPS15] Ryan Lowe, Nissan Pow, Iulian Vlad Serban, and Joelle Pineau. “The Ubuntu Dialogue Corpus: A Large Dataset for Research in Unstructured Multi-Turn Dialogue Systems.” In *Annual Meeting of the Special Interest Group on Discourse and Dialogue*, 2015.
- [LS04] John D Lee and Katrina A See. “Trust in automation: Designing for appropriate reliance.” *Human factors*, **46**(1):50–80, 2004.

- [LSR15] Guosheng Lin, Chunhua Shen, Ian Reid, and Anton van den Hengel. “Deeply learning the messages in message passing inference.” In *NeurIPS*, 2015.
- [LSV16] Guosheng Lin, Chunhua Shen, Anton Van Den Hengel, and Ian Reid. “Efficient piecewise training of deep structured models for semantic segmentation.” In *CVPR*, 2016.
- [LTB16] Yujia Li, Daniel Tarlow, Marc Brockschmidt, and Richard Zemel. “Gated graph sequence neural networks.” In *ICML*, 2016.
- [LTH17] Christian Ledig, Lucas Theis, Ferenc Huszar, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, and Wenzhe Shi. “Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network.” In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4681–4690, 2017.
- [LXP17] Jiasen Lu, Caiming Xiong, Devi Parikh, and Richard Socher. “Knowing When to Look: Adaptive Attention via a Visual Sentinel for Image Captioning.” In *CVPR*, 2017.
- [LYB16] Jiasen Lu, Jianwei Yang, Dhruv Batra, and Devi Parikh. “Hierarchical question-image co-attention for visual question answering.” In *NeurIPS*, 2016.
- [LZS18] Hangxin Liu, Yaofang Zhang, Wenwen Si, Xu Xie, Yixin Zhu, and Song-Chun Zhu. “Interactive robot knowledge patching using augmented reality.” In *Proceedings of International Conference on Robotics and Automation (ICRA)*, 2018.
- [MBM16] Federico Monti, Davide Boscaini, Jonathan Masci, Emanuele Rodolà, Jan Svoboda, and Michael M Bronstein. “Geometric deep learning on graphs and manifolds using mixture model CNNs.” *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.
- [MDS18] Mateusz Malinowski, Carl Doersch, Adam Santoro, and Peter Battaglia. “Learning Visual Question Answering by Bootstrapping Hard Attention.” In *ECCV*, 2018.
- [MF14] Mateusz Malinowski and Mario Fritz. “A multi-world approach to question answering about real-world scenes based on uncertain input.” In *NeurIPS*, 2014.
- [MG00] Maria Madsen and Shirley Gregor. “Measuring human-computer trust.” In *Australasian Conference on Information Systems*, 2000.
- [MGK18] Jiayuan Mao, Chuang Gan, Pushmeet Kohli, Joshua B Tenenbaum, and Jiajun Wu. “The neuro-symbolic concept learner: Interpreting scenes, words, and sentences from natural supervision.” In *International Conference on Learning Representations (ICLR)*, 2018.

- [MLX17] Xudong Mao, Qing Li, Haoran Xie, Raymond YK Lau, Zhen Wang, and Stephen Paul Smolley. “Least squares generative adversarial networks.” In *IEEE International Conference on Computer Vision (ICCV)*, pp. 2794–2802, 2017.
- [MMP10] Marie-Catherine de Marneffe, Christopher D Manning, and Christopher Potts. “Was it good? it was provocative. learning the meaning of scalar adjectives.” In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, 2010.
- [MN12] Paula Marentette and Elena Nicoladis. “Does ontogenetic ritualization explain early communicative gestures in human infants.” *Developments in primate gesture research*, **6**:33, 2012.
- [MO14] Mehdi Mirza and Simon Osindero. “Conditional generative adversarial nets.” *arXiv preprint arXiv:1411.1784*, 2014.
- [MSD18] Daniela Massiceti, N. Siddharth, Puneet K. Dokania, and Philip H.S. Torr. “FlipDial: A Generative Model for Two-Way Visual Dialogue.” In *CVPR*, 2018.
- [MSK19] Seungwhan Moon, Pararth Shah, Anuj Kumar, and Rajen Subba. “Opendialkg: Explainable conversational reasoning with attention-based walks over knowledge graphs.” In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, 2019.
- [Mui94] Bonnie M Muir. “Trust in automation: Part I. Theoretical issues in the study of trust and human intervention in automated systems.” *Ergonomics*, **37**(11):1905–1922, 1994.
- [MXH18] Alexander Mathews, Lexing Xie, and Xuming He. “SemStyle: Learning to Generate Stylised Image Captions Using Unaligned Text.” In *CVPR*, 2018.
- [MXY15] Junhua Mao, Wei Xu, Yi Yang, Jiang Wang, Zhiheng Huang, and Alan Yuille. “Deep captioning with multimodal recurrent neural networks (m-rnn).” In *ICML*, 2015.
- [NAK16] Mathias Niepert, Mohamed Ahmed, and Konstantin Kutzkov. “Learning convolutional neural networks for graphs.” In *Proceedings of International Conference on Machine Learning (ICML)*, 2016.
- [NBG18] Aida Nematzadeh, Kaylee Burns, Erin Grant, Alison Gopnik, and Tom Griffiths. “Evaluating Theory of Mind in Question Answering.” In *Proceedings of the conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2018.

- [NC36] Isaac Newton and John Colson. *The Method of Fluxions and Infinite Series; with Its Application to the Geometry of Curve-lines*. Henry Woodfall; and sold by John Nourse, 1736.
- [Nea11] Radford M Neal. “MCMC using Hamiltonian dynamics.” *Handbook of Markov Chain Monte Carlo*, **2**, 2011.
- [Nes03] Yurii Nesterov. *Introductory lectures on convex optimization: A basic course*, volume 87. Springer Science & Business Media, 2003.
- [NHH20] Erik Nijkamp, Mitch Hill, Tian Han, Song-Chun Zhu, and Ying Nian Wu. “On the Anatomy of MCMC-based Maximum Likelihood Learning of Energy-Based Models.” In *The Thirty-Fourth AAAI Conference on Artificial Intelligence (AAAI)*, pp. 5272–5280, 2020.
- [NHZ19] Erik Nijkamp, Mitch Hill, Song-Chun Zhu, and Ying Nian Wu. “Learning Non-Convergent Non-Persistent Short-Run MCMC Toward Energy-Based Model.” In *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 5233–5243, 2019.
- [NKK11] Jiquan Ngiam, Aditya Khosla, Mingyu Kim, Juhan Nam, Honglak Lee, and Andrew Y Ng. “Multimodal deep learning.” In *International Conference on Machine Learning (ICML)*, pp. 689–696, 2011.
- [NLS18] Medhini Narasimhan, Svetlana Lazebnik, and Alexander Schwing. “Out of the box: Reasoning with graph convolution nets for factual visual question answering.” In *NeurIPS*, 2018.
- [NWC11] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. “Reading digits in natural images with unsupervised feature learning.” In *NIPS workshop on deep learning and unsupervised feature learning*, 2011.
- [ODM18] Georg Ostrovski, Will Dabney, and Rémi Munos. “Autoregressive quantile networks for generative modeling.” In *International Conference on Machine Learning (ICML)*, pp. 3933–3942, 2018.
- [OKB11] Vicente Ordonez, Girish Kulkarni, and Tamara L Berg. “Im2text: Describing images using 1 million captioned photographs.” In *NeurIPS*, 2011.
- [OKK16] Aäron van den Oord, Nal Kalchbrenner, and Koray Kavukcuoglu. “Pixel recurrent neural networks.” In *International Conference on Machine Learning (ICML)*, pp. 1747–1756, 2016.

- [OOS17] Augustus Odena, Christopher Olah, and Jonathon Shlens. “Conditional image synthesis with auxiliary classifier GANs.” In *International Conference on Machine Learning (ICML)*, pp. 2642–2651, 2017.
- [Pea84] Judea Pearl. *Intelligent search strategies for computer problem solving*. Addison Wesley, 1984.
- [PKD16] Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A Efros. “Context encoders: Feature learning by inpainting.” In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2536–2544, 2016.
- [PKK17] Cesc Chunseong Park, Byeongchang Kim, and Gunhee Kim. “Attend to You: Personalized Image Captioning with Context Sequence Memory Networks.” In *CVPR*, 2017.
- [PMS16] Emilio Parisotto, Abdel-rahman Mohamed, Rishabh Singh, Lihong Li, Dengyong Zhou, and Pushmeet Kohli. “Neuro-symbolic program synthesis.” In *International Conference on Learning Representations (ICLR)*, 2016.
- [QJZ20] Siyuan Qi, Baoxiong Jia, and Song-Chun Zhu. “Generalized earley parser: Bridging symbolic grammars and sequence data for future prediction.” *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2020.
- [QLZ20] Shuwen Qiu, Hangxin Liu, Zeyu Zhang, Yixin Zhu, and Song-Chun Zhu. “Human-Robot Interaction in a Shared Augmented Reality Workspace.” In *Proceedings of International Conference on Intelligent Robots and Systems (IROS)*, 2020.
- [QWJ18] Siyuan Qi, Wenguan Wang, Baoxiong Jia, Jianbing Shen, and Song-Chun Zhu. “Learning Human-Object Interactions by Graph Parsing Neural Networks.” In *ECCV*, 2018.
- [RAY16] Scott Reed, Zeynep Akata, Xinchun Yan, Lajanugen Logeswaran, Bernt Schiele, and Honglak Lee. “Generative adversarial text to image synthesis.” In *International Conference on Machine Learning (ICML)*, volume 48, pp. 1060–1069, 2016.
- [RCM19] Siva Reddy, Danqi Chen, and Christopher D Manning. “Coqa: A conversational question answering challenge.” *Transactions of the Association for Computational Linguistics*, 7:249–266, 2019.
- [RFB15] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. “U-net: Convolutional networks for biomedical image segmentation.” In *International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, pp. 234–241, 2015.

- [RKZ15] Mengye Ren, Ryan Kiros, and Richard Zemel. “Exploring models and data for image question answering.” In *NeurIPS*, 2015.
- [RMC16] Alec Radford, Luke Metz, and Soumith Chintala. “Unsupervised representation learning with deep convolutional generative adversarial networks.” In *International Conference on Learning Representations (ICLR)*, 2016.
- [RNS18] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. “Improving language understanding by generative pre-training.”, 2018.
- [Rus03] Bertrand Russell. *The Principles of Mathematics*. Cambridge University Press, 1903.
- [Sax06] Rebecca Saxe. “Uniquely human social cognition.” *Current opinion in neurobiology*, **16**(2):235–239, 2006.
- [SBB20] Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. “Winogrande: An adversarial winograd schema challenge at scale.” In *Proceedings of AAAI Conference on Artificial Intelligence (AAAI)*, 2020.
- [SCH17] Robyn Speer, Joshua Chin, and Catherine Havasi. “Conceptnet 5.5: An open multilingual graph of general knowledge.” In *Proceedings of AAAI Conference on Artificial Intelligence (AAAI)*, 2017.
- [SGT09] Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. “The graph neural network model.” *IEEE TNNLS*, **20**(1):61–80, 2009.
- [SGZ16] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. “Improved techniques for training GANs.” In *Advances in Neural Information Processing Systems (NIPS)*, pp. 2234–2242, 2016.
- [SHK14] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. “Dropout: a simple way to prevent neural networks from overfitting.” *Journal of Machine Learning Research (JMLR)*, **15**(1):1929–1958, 2014.
- [SHL18] Heung-Yeung Shum, Xiao-dong He, and Di Li. “From Eliza to XiaoIce: challenges and opportunities with social chatbots.” *Frontiers of Information Technology & Electronic Engineering*, **19**(1):10–26, 2018.
- [SK17] Martin Simonovsky and Nikos Komodakis. “Dynamic Edge-Conditioned Filters in Convolutional Neural Networks on Graphs.” *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

- [SLH17] Paul Hongsuck Seo, Andreas Lehrmann, Bohyung Han, and Leonid Sigal. “Visual reference resolution using attention memory for visual dialog.” In *NeurIPS*, 2017.
- [SLY15] Kihyuk Sohn, Honglak Lee, and Xinchun Yan. “Learning structured output representation using deep conditional generative models.” In *Advances in Neural Information Processing Systems (NIPS)*, pp. 3483–3491, 2015.
- [SRB17] Adam Santoro, David Raposo, David G Barrett, Mateusz Malinowski, Razvan Pascanu, Peter Battaglia, and Timothy Lillicrap. “A simple neural network module for relational reasoning.” In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, 2017.
- [SRC19] Maarten Sap, Hannah Rashkin, Derek Chen, Ronan Le Bras, and Yejin Choi. “Social IQa: Commonsense Reasoning about Social Interactions.” In *Proceedings of the conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2019.
- [SS12] Nitish Srivastava and Ruslan R Salakhutdinov. “Multimodal learning with deep Boltzmann machines.” In *Advances in Neural Information Processing Systems (NIPS)*, pp. 2222–2230, 2012.
- [SSF16] Sainbayar Sukhbaatar, Arthur Szlam, and Rob Fergus. “Learning multiagent communication with backpropagation.” In *NEURIPS*, 2016.
- [SSH16] Kevin J Shih, Saurabh Singh, and Derek Hoiem. “Where to look: Focus regions for visual question answering.” In *CVPR*, 2016.
- [SV10] David Silver and Joel Veness. “Monte-Carlo planning in large POMDPs.” In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, 2010.
- [SVI16] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. “Rethinking the inception architecture for computer vision.” In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2818–2826, 2016.
- [SW86] Dan Sperber and Deirdre Wilson. *Relevance: Communication and cognition*. Harvard University Press Cambridge, MA, 1986.
- [SYC19] Kai Sun, Dian Yu, Jianshu Chen, Dong Yu, Yejin Choi, and Claire Cardie. “DREAM: A Challenge Data Set and Models for Dialogue-Based Reading Comprehension.” *Transactions of the Association for Computational Linguistics*, 7:217–231, 2019.

- [SZ15] Karen Simonyan and Andrew Zisserman. “Very deep convolutional networks for large-scale image recognition.” In *ICLR*, 2015.
- [SZZ20] Stephanie Stacy, Qingyi Zhao, Minglu Zhao, Max Kleiman-Weiner, and Tao Gao. “Intuitive signaling through an “Imagined We”.” In *Proceedings of the Annual Meeting of the Cognitive Science Society (CogSci)*, 2020.
- [TAH18] Damien Teney, Peter Anderson, Xiaodong He, and Anton van den Hengel. “Tips and Tricks for Visual Question Answering: Learnings From the 2017 Challenge.” In *CVPR*, 2018.
- [TC97] Michael Tomasello and Josep Call. *Primate cognition*. Oxford University Press, 1997.
- [THL19] Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. “CommonsenseQA: A Question Answering Challenge Targeting Commonsense Knowledge.” In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, 2019.
- [TLH17] Damien Teney, Lingqiao Liu, and Anton van den Hengel. “Graph-Structured Representations for Visual Question Answering.” In *CVPR*, 2017.
- [Tom96] Michael Tomasello. “Do apes ape.” *Social learning in animals: The roots of culture*, pp. 319–346, 1996.
- [Tom10] Michael Tomasello. *Origins of human communication*. MIT press, 2010.
- [TPZ13] Kewei Tu, Maria Pavlovskaja, and Song-Chun Zhu. “Unsupervised structure learning of stochastic and-or grammars.” In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, 2013.
- [TS13] Radim Tyleček and Radim Šára. “Spatial pattern templates for recognition of objects with regular structure.” In *German Conference on Pattern Recognition (GCPR)*, pp. 364–374, 2013.
- [TSZ20] Ning Tang, Stephanie Stacy, Minglu Zhao, Gabriel Marquez, and Tao Gao. “Bootstrapping an imagined We for cooperation.” In *Proceedings of the Annual Meeting of the Cognitive Science Society (CogSci)*, 2020.
- [TZ02] Michael Tomasello and Klaus Zuberbühler. *Primate vocal and gestural communication*. MIT Press, 2002.
- [Vaa00] Aad W Van der Vaart. *Asymptotic statistics*, volume 3. Cambridge university press, 2000.
- [VH10] Tony Veale and Yanfen Hao. “Detecting ironic intent in creative comparisons.” In *European Conference on Artificial Intelligence (ECAI)*, 2010.

- [Voo99] Ellen M Voorhees et al. “The TREC-8 question answering track report.” In *Trec*, 1999.
- [VSC17] Harm de Vries, Florian Strub, Sarath Chandar, Olivier Pietquin, Hugo Larochelle, and Aaron Courville. “GuessWhat?! Visual Object Discovery Through Multi-Modal Dialogue.” In *CVPR*, 2017.
- [VTB15] Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. “Show and tell: A neural image caption generator.” In *CVPR*, 2015.
- [WBC15] Jason Weston, Antoine Bordes, Sumit Chopra, Alexander M Rush, Bart van Merriënboer, Armand Joulin, and Tomas Mikolov. “Towards ai-complete question answering: A set of prerequisite toy tasks.” *arXiv preprint arXiv:1502.05698*, 2015.
- [WCB14] Jason Weston, Sumit Chopra, and Antoine Bordes. “Memory networks.” *arXiv preprint arXiv:1410.3916*, 2014.
- [WF01] Yair Weiss and William T Freeman. “On the optimality of solutions of the max-product belief-propagation algorithm in arbitrary graphs.” *IEEE Transactions on Information Theory*, 2001.
- [Wit53] Ludwig Wittgenstein. *Philosophical investigations. Philosophische Untersuchungen*. Macmillan, 1953.
- [Wit69] Ludwig Wittgenstein. *The blue and brown books*, volume 958. Blackwell Oxford, 1969.
- [WL16] Dilin Wang and Qiang Liu. “Learning to draw samples: With application to amortized MLE for generative adversarial learning.” *arXiv preprint arXiv:1611.01722*, 2016.
- [WLZ18a] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. “High-resolution image synthesis and semantic manipulation with conditional GANs.” In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8798–8807, 2018.
- [WLZ18b] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Nikolai Yakovenko, Andrew Tao, Jan Kautz, and Bryan Catanzaro. “Video-to-Video Synthesis.” In *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 1152–1164, 2018.
- [WT09] Xiaogang Wang and Xiaoou Tang. “Face photo-sketch synthesis and recognition.” *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, **31**(11):1955–1967, 2009.

- [WWS18a] Qi Wu, Peng Wang, Chunhua Shen, Ian Reid, and Anton van den Hengel. “Are You Talking to Me? Reasoned Visual Dialog Generation Through Adversarial Learning.” In *CVPR*, 2018.
- [WWS18b] Qi Wu, Peng Wang, Chunhua Shen, Ian Reid, and Anton Van Den Hengel. “Are you talking to me? reasoned visual dialog generation through adversarial learning.” In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.
- [WWX17] Yu Wu, Wei Wu, Chen Xing, Ming Zhou, and Zhoujun Li. “Sequential Matching Network: A New Architecture for Multi-turn Response Selection in Retrieval-Based Chatbots.” In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, 2017.
- [WXS18] Wenguan Wang, Yuanlu Xu, Jianbing Shen, and Song-Chun Zhu. “Attentive fashion grammar network for fashion landmark detection and clothing category classification.” In *CVPR*, 2018.
- [XBK15] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. “Show, attend and tell: Neural image caption generation with visual attention.” In *ICML*, 2015.
- [XLG18a] Jianwen Xie, Yang Lu, Ruiqi Gao, and Ying Nian Wu. “Cooperative learning of energy-based model and latent variable model via MCMC teaching.” In *The Thirty-Second AAAI Conference on Artificial Intelligence (AAAI)*, pp. 4292–4301, 2018.
- [XLG18b] Jianwen Xie, Yang Lu, Ruiqi Gao, Song-Chun Zhu, and Ying Nian Wu. “Cooperative training of descriptor and generator networks.” *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, **42**(1):27–45, 2018.
- [XLZ16] Jianwen Xie, Yang Lu, Song-Chun Zhu, and Ying Nian Wu. “A theory of generative ConvNet.” In *International Conference on Machine Learning (ICML)*, pp. 2635–2644, 2016.
- [XRV17] Han Xiao, Kashif Rasul, and Roland Vollgraf. “Fashion-MNIST: a novel image dataset for benchmarking machine learning algorithms.” *arXiv preprint arXiv:1708.07747*, 2017.
- [XT15] Saining Xie and Zhuowen Tu. “Holistically-nested edge detection.” In *IEEE International Conference on Computer Vision (ICCV)*, pp. 1395–1403, 2015.
- [XYA19] Zhisheng Xiao, Qing Yan, and Yali Amit. “Generative Latent Flow.” *arXiv preprint arXiv:1905.10485*, 2019.

- [XYH05] Eric P. Xing, Rong Yan, and Alexander G. Hauptmann. “Mining associated text and images with dual-wing harmoniums.” In *Conference in Uncertainty in Artificial Intelligence (UAI)*, pp. 633–641, 2005.
- [XZF21] Jianwen Xie, Zilong Zheng, Xiaolin Fang, Song-Chun Zhu, and Ying Nian Wu. “Learning Cycle-Consistent Cooperative Networks via Alternating MCMC Teaching for Unsupervised Cross-Domain Translation.” In *The Thirty-Fifth AAAI Conference on Artificial Intelligence (AAAI)*, 2021.
- [XZG18] Jianwen Xie, Zilong Zheng, Ruiqi Gao, Wenguan Wang, Song-Chun Zhu, and Ying Nian Wu. “Learning descriptor networks for 3D shape synthesis and analysis.” In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8629–8638, 2018.
- [XZG20] Jianwen Xie, Zilong Zheng, Ruiqi Gao, Wenguan Wang, Song-Chun Zhu, and Ying Nian Wu. “Generative VoxelNet: Learning Energy-Based Models for 3D Shape Synthesis and Analysis.” *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2020.
- [XZW17] Jianwen Xie, Song-Chun Zhu, and Ying Nian Wu. “Synthesizing Dynamic Patterns by Spatial-Temporal Generative ConvNet.” In *CVPR*, 2017.
- [XZW19] Jianwen Xie, Song-Chun Zhu, and Ying-Nian Wu. “Learning Energy-based Spatial-Temporal Generative ConvNets for Dynamic Patterns.” *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2019.
- [YCC18] Tom Young, Erik Cambria, Iti Chaturvedi, Hao Zhou, Subham Biswas, and Minlie Huang. “Augmenting end-to-end dialogue systems with commonsense knowledge.” In *Proceedings of AAAI Conference on Artificial Intelligence (AAAI)*, 2018.
- [YHG16] Zichao Yang, Xiaodong He, Jianfeng Gao, Li Deng, and Alex Smola. “Stacked attention networks for image question answering.” In *CVPR*, 2016.
- [YLF20] Tao Yuan, Hangxin Liu, Lifeng Fan, Zilong Zheng, Tao Gao, Yixin Zhu, and Song-Chun Zhu. “Joint inference of states, robot knowledge, and human (false-) beliefs.” In *Proceedings of International Conference on Robotics and Automation (ICRA)*, 2020.
- [You99] Laurent Younes. “On the convergence of Markovian stochastic algorithms with rapidly decreasing ergodicity rates.” *Stochastics: An International Journal of Probability and Stochastic Processes*, **65**(3-4):177–228, 1999.
- [YWG18] Kexin Yi, Jiajun Wu, Chuang Gan, Antonio Torralba, Pushmeet Kohli, and Josh Tenenbaum. “Neural-symbolic vqa: Disentangling reasoning from vision

- and language understanding.” In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, 2018.
- [ZDU18] Saizheng Zhang, Emily Dinan, Jack Urbanek, Arthur Szlam, Douwe Kiela, and Jason Weston. “Personalizing Dialogue Agents: I have a dog, do you have pets too?” In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, 2018.
- [ZGB16] Yuke Zhu, Oliver Groth, Michael Bernstein, and Li Fei-Fei. “Visual7w: Grounded question answering in images.” In *CVPR*, 2016.
- [ZJR15] Shuai Zheng, Sadeep Jayasumana, Bernardino Romera-Paredes, Vibhav Vineet, Zhizhong Su, Dalong Du, Chang Huang, and Philip HS Torr. “Conditional random fields as recurrent neural networks.” In *ICCV*, 2015.
- [ZM07] Song-Chun Zhu and David Mumford. *A stochastic grammar of images*. Now Publishers Inc, 2007.
- [ZMB08] Brian D Ziebart, Andrew L Maas, J Andrew Bagnell, and Anind K Dey. “Maximum Entropy Inverse Reinforcement Learning.” In *Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence*, pp. 1433–1438, 2008.
- [ZML17] Junbo Zhao, Michael Mathieu, and Yann LeCun. “Energy-based generative adversarial network.” In *International Conference on Learning Representations (ICLR)*, 2017.
- [ZPI17] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. “Unpaired image-to-image translation using cycle-consistent adversarial networks.” In *IEEE International Conference on Computer Vision (ICCV)*, pp. 2223–2232, 2017.
- [ZRH20] Quanshi Zhang, Jie Ren, Ge Huang, Ruiming Cao, Ying Nian Wu, and Song-Chun Zhu. “Mining Interpretable AOG Representations from Convolutional Networks via Active Question Answering.” *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2020.
- [ZWQ19] Zilong Zheng, Wenguan Wang, Siyuan Qi, and Song-Chun Zhu. “Reasoning visual dialogs with structural and partial observations.” In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.
- [ZWS18] Bohan Zhuang, Qi Wu, Chunhua Shen, Ian D Reid, and Anton van den Hengel. “Parallel attention: A unified framework for visual object discovery through dialogs and queries.” In *CVPR*, 2018.
- [ZWW20] Quanshi Zhang, Xin Wang, Ying Nian Wu, Huilin Zhou, and Song-Chun Zhu. “Interpretable CNNs for Object Classification.” *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2020.

- [ZWZ18] Quanshi Zhang, Ying Nian Wu, and Song-Chun Zhu. “Interpretable convolutional neural networks.” In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.
- [ZXL17] Han Zhang, Tao Xu, Hongsheng Li, Shaoting Zhang, Xiaogang Wang, Xiao lei Huang, and Dimitris N Metaxas. “StackGAN: Text to photo-realistic image synthesis with stacked generative adversarial networks.” In *IEEE International Conference on Computer Vision (ICCV)*, pp. 5907–5915, 2017.
- [ZZ18] Quanshi Zhang and Song-Chun Zhu. “Visual interpretability for deep learning: a survey.” *Frontiers of Information Technology & Electronic Engineering*, **19**(1):27–39, 2018.
- [ZZP17] Jun-Yan Zhu, Richard Zhang, Deepak Pathak, Trevor Darrell, Alexei A Efros, Oliver Wang, and Eli Shechtman. “Toward multimodal image-to-image translation.” In *Advances in Neural Information Processing Systems (NIPS)*, pp. 465–476, 2017.
- [ZZZ20] Zhenliang Zhang, Yixin Zhu, and Song-Chun Zhu. “Graph-based Hierarchical Knowledge Representation for Robot Task Transfer from Virtual to Physical World.” In *Proceedings of International Conference on Intelligent Robots and Systems (IROS)*, 2020.