University of California
Santa Barbara

# Achieving Human-like Chatbots from Reasoning and Optimization Perspectives

A dissertation submitted in partial satisfaction
of the requirements for the degree

Doctor of Philosophy
in
Computer Science

by

Yi-Lin Tuan

Committee in charge:

Professor William Yang Wang, Chair
Professor Lise Getoor
Professor Xifeng Yan

September 2024

The Dissertation of Yi-Lin Tuan is approved.

Professor Lise Getoor

Professor Xifeng Yan

Professor William Yang Wang, Committee Chair

August 2024

Achieving Human-like Chatbots
from Reasoning and Optimization Perspectives

To my family

# Acknowledgements

First and foremost, I would like to thank my advisor and chair of my committee, Professor William Wang, for his invaluable support and open-mindedness. I also could not have undertaken this journey without my defense committee, Professor Lise Getoor and Professor Xifeng Yan, who generously provided knowledge and expertise.

Additionally, this endeavor would not have been possible without the generous support from my advisors at National Taiwan University before PhD study, Professor Lin-shan Lee, Professor Hung-yi Lee, and Professor Yun-Nung Chen, who guided me into the machine learning research journey.

I am also grateful to the teams when doing internships in the industry, especially my mentors and managers, Ahmed-El Kishky, Paco Guzmán, Sajjad Beygi, Maryam Fazal-Zarandi, Wenhan Xiong, Yashar Mehdad, Dan Bikel, for their experience sharing, editing help, and precious feedback.

Thanks should also go to my co-workers and friends, Connor Pryor, Alon Albalak, Sharon Levy, Wenda Xu, Kai Nakamura, Wenhu Chen, Micheal Saxon, Wei Wei, Yujie Lu, Professor Lei Li, Varun Embar, Adithya Renduchintala, Vishrav Chaudhary, Professor Lucia Specia, Che-ping Tsai, Jinzhi Zhang, Qiaozi Gao, Alessandra Cervone, Feng-Guang Su, Aliyah R Hsu, Pegah Jandaghi, Deepak Ramachandran, Professor Jay Pujara, Lucas Relic, Bowen Zhang, Professor Michael Beyeler, Xilun Chen, Eric Michael Smith, Louis Martin, Soumya Batra, Asli Celikyilmaz, Professor Sui Tang, Professor Shiyu Chang, Professor Kenneth Huang, Professor Michael Yip, Chunting Zhou, Chunyang Wu, Anchit Gupta, Tsu-Jui Fu, Wanrong Zhu, and many more people I'm incredibly grateful to but unable to mention individually for their project discussion, editing help, and moral support.

Lastly, I would like to extend my deepest gratitude to my family for their unwavering

emotional support throughout this journey. Their encouragement, love, and belief in me have been a constant source of strength, especially during the most challenging moments. Specifically, I am lucky to have Zih-Yun always standing by my side for the past nine years. Without her accompany for all the ups and downs, I could not go through the journey.

## Curriculum Vitæ

Yi-Lin Tuan

### Education

| | |
|---|---|
| 2019-2024 | Ph.D. in Computer Science, University of California, Santa Barbara |
| 2013-2017 | B.S. in Electrical Engineering, National Taiwan University |

### Work Experience

| | |
|---|---|
| 2023 | Research Intern, Meta AI |
| 2022 | Research Intern, Meta AI |
| 2021 | Applied Scientist Intern, Amazon Alexa AI |
| 2020 | Research Intern, Facebook AI |
| 2017-2019 | Full-time Research Assistant, National Taiwan University |

### Publications

* indicates equal contribution.
P: pre-print or under submission. C: conference. W: workshop. J: journal.

[**P6**] Yi-Lin Tuan and William Yang Wang. *"A Gradient Analysis Framework for Rewarding Good and Penalizing Bad Examples in Language Models"*. Under Submission, 2024.

[**P5**] Yi-Lin Tuan, Xilun Chen, Eric Michael Smith, Louis Martin, Soumya Batra, Asli Celikyilmaz, William Yang Wang, Daniel M. Bikel. *"Towards Safety and Helpfulness Balanced Responses via Controllable Large Language Models"*. Preprint, 2024.

[**P4**] Yi-Lin Tuan, Zih-Yun Chiu, William Yang Wang. *"Dynamic Latent Separation for Deep Learning"*. Preprint, 2024.

[**C12**] Zih-Yun Chiu*, Yi-Lin Tuan*, William Yang Wang, Michael Yip. *"Flexible Attention-Based Multi-Policy Fusion for Efficient Deep Reinforcement Learning"*. **NeurIPS**, 2023.

[**C11**] Yi-Lin Tuan, Alon Albalak, Wenda Xu, Michael Saxon, Connor Pryor, Lise Getoor, William Yang Wang. *"CausalDialogue: Modeling Utterance-level Causality in Conversations"*. **ACL** Findings, 2023.

[**C10**] Alon Albalak, Yi-Lin Tuan, Pegah Jandaghi, Connor Pryor, Luke Yoffe, Deepak Ramachandran, Lise Getoor, Jay Pujara, William Yang Wang. *"FETA: A Benchmark for Few-Sample Task Transfer in Open-Domain Dialogue"*. **EMNLP**, 2022.

[**C9**] Wenda Xu, Yi-Lin Tuan, Yujie Lu, Michael Saxon, Lei Li, William Yang Wang. *"Not All Errors are Equal: Learning Text Generation Metrics using Stratified Error Synthesis"*. **EMNLP** Findings, 2022.

[**C8**] Kai Nakamura, Sharon Levy, Yi-Lin Tuan, Wenhu Chen, William Yang Wang. *"HybriDialogue: An Information-Seeking Dialogue Dataset Grounded on Tabular and Textual Data"*. **ACL** Findings, 2022.

[**C7**] Yi-Lin Tuan, Sajjad Beygi, Maryam Fazel-Zarandi, Qiaozi Gao, Alessandra Cervone, and William Yang Wang. *"Towards Large-Scale Interpretable Knowledge Graph Reasoning for Dialogue Systems"*. **ACL** Findings, 2022.

[**W1**] Alon Albalak, Varun Embar, Yi-Lin Tuan, Lise Getoor, William Yang Wang. *"D-REX: Dialogue Relation Extraction with Explanations"*. **ACL** NLP4ConvAI workshop, 2022.

[**C6**] Lucas Relic, Bowen Zhang, Yi-Lin Tuan, Michael Beyeler. *"Deep Learning Based Perceptual Stimulus Encoder for Bionic Vision"*. **ACM AHs**, 2022.

[**C5**] Yi-Lin Tuan, Connor Pryor, Wenhu Chen, Lise Getoor, and William Yang Wang. *"Local Explanation of Dialogue Response Generation"*. **NeurIPS**, 2021.

[**C4**] Yi-Lin Tuan, Ahmed El-Kishky, Adi Renduchintala, Vishrav Chaudhary, Francisco Guzman, and Lucia Specia. *"Quality Estimation without Human-labeled Data"*. **EACL**, 2021.

[**P3**] Zih-Yun Chiu, Yi-Lin Tuan, Hung-yi Lee, Li-Chen Fu. *"Parallelized Reverse Curriculum Generation"*. Preprint, 2021.

[**P2**] Yi-Lin Tuan, Wei Wei, and William Yang Wang. *"Knowledge Injection into Dialogue Generation via Language Models"*. Preprint, 2020.

[**C3**] Yi-Lin Tuan, Yun-Nung Chen, and Hung-yi Lee. *"DyKgChat: Benchmarking Dialogue Generation Grounding on Dynamic Knowledge Graphs"*. **EMNLP**, 2019.

[**C2**] Feng-Guang Su*, Aliyah R Hsu*, Yi-Lin Tuan, and Hung-Yi Lee. *"Personalized Dialogue Response Generation Learned from Monologues"*. **Interspeech**, 2019.

[**J1**] Yi-Lin Tuan and Hung-yi Lee. *"Improving conditional sequence generative adversarial networks by stepwise evaluation"*. **IEEE/ACM Transactions on Audio, Speech, and Language Processing**, 2019.

[**P1**] Yi-Lin Tuan*, Jinzhi Zhang*, Yujia Li, and Hung-yi Lee. *"Proximal Policy Optimization and its Dynamic Version for Sequence Generation"*. Preprint, 2018.

[**C1**] Che-Ping Tsai*, Yi-Lin Tuan*, and Lin-shan Lee. *"Transcribing lyrics from commercial song audio: the first step towards singing content processing"*. **ICASSP**, 2018.

# Abstract

Achieving Human-like Chatbots
from Reasoning and Optimization Perspectives

by

Yi-Lin Tuan

Human-like chatbots – machines that can act as humans to chat about any topic – need to listen, understand, reason, respond, and interactively learn to optimize the whole process. Since requiring to conduct these complex tasks, the advancement of human-like chatbots often marks the evolution of artificial intelligence. Recent developments in machine learning for artificial intelligence, such as recurrent neural networks, transformers, and large language models (LLMs), have been progressively taken as the backbone models for chatbots. Among them, the latest LLMs have shown impressive abilities to interact with users in chatting-like scenarios with proper utterances. However, LLMs have yet to reflect human-like attributes, their reasoning processes are intransparent, the inner work of optimization remains a black box, and they require significant scaling of model and data sizes. These issues prevent further development of more efficient, effective, and explainable human-like chatbots. In this dissertation, I address these issues from three aspects: (1) Unveiling reasoning process from post-hoc and prevention views, (2) optimization methods to improve human-like attributes, and (3) optimization techniques with reasoning interpretation.

This dissertation contributes to algorithms, frameworks, and paradigms that reveal the underlying reasoning process of human-like chatbots and optimize chatbots toward human-like attributes. First, I develop a method to explain any black-box language model behaviors. This approach unveils the relationship of input and output segments from the statistical view of the model. Besides the theoretical desired properties, this approach also shows generalizability and human readability through empirical evaluation and human study. Second, I present a framework to actively disclose the reasoning process before text generation. This framework can be inserted into any model type and provides the reasoning path as a sequence of traversed knowledge graph triples. Through experiments, the framework shows its scalability to large-scale knowledge graphs and its efficacy in keeping or improving performance while providing interpretation. Third, I propose a loss function to promote response quality, agility, and steerability. I derive this loss function from modeling conversation generation in the view of causality. The proposed loss function shows its generalizability, efficacy, and efficiency across various models and data types via empirical results and advanced gradient analyses. Thereafter, I explore advanced reinforcement and representation learning algorithms, which are two critical directions in machine learning and have shown benefits in chatbot training. I introduced our efforts to allow reinforcement learning to efficiently use existing knowledge, thus promoting learning speed and results. Finally, I introduce the new concept of modeling data examples as atoms, using physical principles to discretize data examples within a continuous space. These developed approaches are optimization methods that also equip a model with an interpretable reasoning process. Experiments show their generalizability in broad domains, from vision synthesis to robotics control, and point out an expectation of their future in helping chatbot learning. Together, this dissertation provides top-down design ideas and bottom-up fundamental theory for human-like chatbots and exhibits future possibilities to unlock a chatbot's ability in advance.

# Contents

# Chapter 1

# Introduction

Chatbots that humans can chat with everything, ask anything, and get help from can potentially improve our daily lives [1, 2, 3, 4]. For example, chatbots that role-play with humans can engage us in either learning or entertainment; chatbots as customer service agents can facilitate the process; chatbots as search engines can make information more accessible. Acquiring these utilities needs chatbots to master listening, comprehending, reasoning, and responding through supervision and interactions with the world. Because of the requirement to carry out this sequence of complex tasks, including understanding and generating various forms of natural language, the development of chatbots has been used to measure the advancement of machine learning techniques [5, 6, 7, 8] and closeness to artificial general intelligence [9, 10].

In recent years, studies in machine learning and natural language processing have facilitated chatbot developments. The efforts include algorithm and model designs to (1) scale model and data sizes to increase overall performance [11, 12, 13, 14, 15, 16, 17, 2, 18], (2) enable specific applications, uch as conversational assistants that help arrange schedules, give suggestions [19], or search the internet [20], and (3) integrate desired chatbot attributes, such as personality [21, 22, 23], talking style [24], knowledge from the back-end [25, 26, 27], or information retrieved from the internet [28]. While enormous research has demonstrated success, human-like chatbots still face several challenges. In this dissertation, I address two key challenges: reasoning and optimization. I examine current approaches and develop new methods to tackle these issues.

## 1.1 Challenges

**Reasoning.** Despite the improved naturalness and logic in text generated by LLMs when tested on selected benchmarks, these models suffer from limited transparency and robustness, potentially harming users. For instance, the black box characteristic of their reasoning process raises concerns about copyright [29] and takes over control of products from users, preventing users from fully trusting the responses [30]. Studies are exploring methods to explain other natural language tasks [31, 32] to shed light on the underlying reasoning process of such black box models. However, human-like chatbots present unique challenges that hinder the direct usage of existing methods, including the unbounded possible responses and the unclear relationship among utterances in a conversation.

**Optimization.** Existing LLMs research often centers around addressing the enormous consumptions of energy [33] by reducing model sizes [34, 35] and making superhuman chatbots to serve as specialized assistants [19, 36]. These research lines shift away from the goal of human-like chatbots. Achieving human-like chatbots requires models to have human-like attributes, such as personalities, speaking styles, and engaging in a conversation akin to an actual human. These requirements are based on one fundamental constraint: Humans can continue a conversation with many proper and possible responses [37, 38, 39]. This constraint poses a challenge: How can we optimize a model without unique ground-truth labels but still mimicking human behaviors? Moreover, while recent studies have made progress in injecting human-like attributes into chatbots, including response diversity [40, 41], personality consistency [21, 42, 24], and engagement [38, 26, 43], the efficacy of the optimization methods is yet satisfied and needs advance.

## 1.2 Overview

In this dissertation, we ask: How can we understand the models' reasoning process underneath, prevent potential harm to users and their environments, and improve training efficiency and efficacy? By exposing underlying reasons, we can gain real users' trust in a chatbot. By improving optimization methods, we can train the human-like chatbot more efficiently and effectively. Finally, we explore optimization methods that can give models inherent reasoning ability and expect to take future steps to deploy them on human-like chatbots.

**Part I: Interpreting Reasoning Process in LMs.** I discuss the methods to reveal the reasoning process of human-like chatbots from both retrospective and precautionary perspectives. In Chapter 2, I introduce a method to extend post-hoc and model-agnostic explanation approaches to language model generation, which needs to consider the properties of multiple classes, multiple time steps, and implicit relationships between input and output in conversations [44]. This method, called Local Explanation for Response Generation (LERG), is mathematically proven to have the desired properties to faithfully represent a model's prediction through the extracted explanations. The experiments demonstrate that LERG's explanation shows significantly higher necessity and sufficiency to a model's responses and gains higher readability from real user studies than other alternatives. In Chapter 3, I look into increasing the scalability and interpretability of the reasoning process in knowledge graph-grounded dialogue generation tasks [45]. The proposed framework – differentiable knowledge graph dialogue model (DiffKG) – can actively exhibit its underlying reasoning process during response generation. Not only does it show the ability to unveil reasoning processes and tackle small to large-scale knowledge graphs without subgraph sampling beforehand, but DiffKG also demonstrates on-par or improved entity F1 compared to the state-of-the-art in the generated responses.

**Part II: Optimization Method Advancement.** I introduce a set of optimization methods for human-like chatbots with a shared purpose to be generally applicable for overall performance improvement and desired properties injection into the responses. In Chapter 4, I look at a chatbot's ability to agilely reply to humans with different content, even if the input conversations are similar [46]. I propose a loss function called exponential maximum average treatment effect (ExMATE), which views the chatbot response as the effect and the difference of nuanced input conversation as the cause. We prepare a dataset, CausalDialogue, that mixes expert-written scripts and crowd-sources expansion, thus embracing the characteristics of diverse branches and colliders, high-quality conversations, and language abundance. Tested on CausalDialogue, ExMATE shows superior response agility compared to the widely used maximum likelihood estimation method while maintaining high fluency. In Chapter 5, I further investigate the underlying theory foundation of ExMATE. I propose a new gradient analysis paradigm for language model generation [47] that considers the impacts from literal similarity, softmax layer, and the multiple classes per time step. Applying this gradient analysis paradigm, I discover the fundamental differences among optimization methods that reward good examples while penalizing bad ones, such as ExMATE, unlikelihood training, and direct preference optimization (DPO). With empirical verification, ExMATE shows overall superior results than MLE and unlikelihood training and enhances DPO performance while being more effective than DPO in broader scenarios. In Chapter 6, Noticing that helpfulness and harmlessness can occasionally conflict in conversations, I investigate methods to revert optimized LLMs, leverage their knowledge, and control their levels of helpfulness and safety [48]. The proposed framework unlocks the steerability of an LLM by first generating data from the LLM, distilling input data with multiple extreme cases, and optimizing using ExMATE. Compared to reinforcement learning from human feedback, MLE, and training-free approaches such as reranking multiple generations, ExMATE shows the best controllability in both helpfulness and safety attributes. This work also demonstrates the possibility of unsealing the knowledge of safety-prioritized LLMs, thus making LLMs more flexible for different applications.

**Part III: Optimization Methods with Reasons.** I explore enhancing optimization methods and equipping the optimized models with interpretability. The surveyed optimization methods have shown their potential to improve language models. Chapter 7 introduces a knowledge-grounded reinforcement learning (KGRL) paradigm to allow machines to have human-like learning behaviors [49]. We propose a knowledge-inclusive attention network (KIAN) that enables the machine to efficiently leverage external guidance and reuse learned knowledge. Our experiments demonstrate that KIAN in KGRL addresses the issues of sample efficiency, generalizability, and knowledge use. Meanwhile, KIAN provides its reasoning process when operating in an environment. In Chapter 8, I propose an auxiliary loss function to enhance the representation space for a model. This method, modeling each data example as an atom, activates the end-to-end optimization to dynamically separate the latent codes of data samples to diversify the output space further [50]. Empirical results show that this approach enhances the output separation for better classification and higher generation diversity. Simultaneously, atom modeling assists in explaining the importance of input features to data semantic meaning and dissimilarity. The explored methods in Chapters 7 and 8 show the potential to advance grounded, efficient, and interpretable chatbot learning in the future.

In Chapter 9, as the conclusion, I summarize our research on developing human-like chatbots, focusing on methods to understand model reasoning capabilities and optimization techniques to improve training efficiency and effectiveness. Ultimately, I discuss the potential future directions to further advance human-like chatbots from both the reasoning and optimization perspectives.

# Part I

# Understanding the Reasoning Process of Chatbots

# Chapter 2

# Explaining a Trained Language Model

To first reveal the underlying reasoning process of arbitrary language models, in this chapter, we study the model-agnostic explanations of a representative text generation task – dialogue response generation. However, in comparison to the interpretation of classification models, the explanation of language models is important yet has seen little attention. Dialog response generation is challenging with its open-ended sentences and multiple acceptable responses. To overcome the challenges, we propose a new method, local explanation of response generation (LERG), that regards the explanations as the mutual interaction of segments in input and output sentences. LERG views the sequence prediction as uncertainty estimation of a human response and then creates explanations by perturbing the input and calculating the certainty change over the human response. We show that LERG adheres to desired properties of explanation for text generation, including unbiased approximation, consistency, and cause identification. Empirically, LERG consistently enhances classifier-focused explanation methods on proposed automatic- and human- evaluation metrics for this new task by 4.4-12.8%. Analysis also demonstrates that LERG can extract both explicit and implicit relations between input and output segments.

## 2.1 Introduction

As we use machine learning models in daily tasks, such as medical applications [51, 52, 53], speech processing [54, 55], etc., being able to trust the predictions being made has become increasingly important. To understand the underlying reasoning process of complex machine learning models a sub-field of explainable artificial intelligence (XAI) [56, 57, 58] called local explanations, has seen promising results [31]. Local explanation methods [59, 60] often approximate an underlying black box model by fitting an interpretable proxy, such as a linear model or tree, around the neighborhood of individual predictions. These methods have the advantage of being model-agnostic and locally interpretable.

Traditionally, off-the-shelf local explanation frameworks, such as the Shapley value in game theory [61] and the learning-based Local Interpretable Model-agnostic Explanation (LIME) [31] have been shown to work well on classification tasks with a small number of classes. In particular, there has been work on image classification [31], sentiment analysis [62], and evidence selection for question answering [63]. However, to the best of our knowledge, there has been less work studying explanations over models with sequential output and large class sizes at each time step. An attempt by [32] aims at explaining machine translation by aligning the sentences in source and target languages. Nonetheless, unlike translation, where it is possible to find almost all word alignments of the input and output sentences, many text generation tasks are not alignment-based. We further explore explanations over sequences that contain implicit and indirect relations between the input and output utterances.

In this chapter, we study explanations over a set of representative conditional text generation models – dialogue response generation models [6, 15]. These models typically aim to produce an engaging and informative [38, 64] response to an input message. The open-ended sentences and multiple acceptable responses in dialogues pose two major challenges: (1) an exponentially large output space and (2) the implicit relations between the input and output texts. For example, the open-ended prompt "How are you today?" could lead to multiple responses depending on the users' emotion, situation, social skills, expressions, etc. A simple answer such as "Good. Thank you for asking." does not have an explicit alignment to words in the input prompt. Even though this alignment does not exist, it is clear that "good" is the key response to "how are you". To find such crucial corresponding parts in a dialogue, we propose to extract explanations that can answer the question: *"Which parts of the response are influenced the most by parts of the prompt?"*

To obtain such explanations, we introduce *LERG*, a novel yet simple method that extracts the ranked importance scores of every input-output segment pair from a dialogue response generation model. We view this sequence prediction as the uncertainty estimation of the response and find a linear proxy that simulates the certainty caused from one input segment to an output segment. We further derive two optimization variations of LERG. The first is learning-based [31], while the other derives an optimal similar to Shapley value [61]. To theoretically verify LERG, we propose that an ideal explanation of text generation should adhere to three properties: unbiased approximation, intra-response consistency, and causal cause identification.

To verify if the explanations are both faithful (the explanation is fully dependent on the model being explained) [56] and interpretable (the explanation is understandable by humans) [65], we conduct comprehensive automatic evaluations and user study. For automatic evaluations, we measure the *necessity*, perplexity change when removing salient input segments, and *sufficiency*, perplexity of only salient segments remaining, of the extracted explanation to the generation model. In our user study, we present annotators with only the most salient parts in an input and ask them to select the most appropriate response from a set of candidates. Empirically, our proposed method consistently outperforms baselines on both automatic metrics and human evaluation.

## 2.2 Background: Local Explanation

Local explanation methods aim to explain predictions of an arbitrary model by interpreting the neighborhood of individual predictions [31]. It can be viewed as training a proxy that adds the contributions of input features to a model's predictions [60]. More formally, given an example with input features $x = \{x_i\}_{i=1}^{M}$, the corresponding prediction $y$ with probability $f(x) = P_\theta(Y = y|x)$ (the classifier is parameterized by $\theta$), we denote the contribution from each input feature $x_i$ as $\phi_i \in \mathbb{R}$ and denote the concatenation of all contributions as $\boldsymbol{\phi} = [\phi_1, ..., \phi_M]^T \in \mathbb{R}^M$. Two popular local explanation methods are the learning-based Local Interpretable Model-agnostic Explanations (LIME) [31] and the game theory-based Shapley value [61].

**LIME** interprets a complex classifier $f$ based on locally approximating a linear classifier around a given prediction $f(x)$. The optimization of the explanation model that LIME uses adheres to:

$$\xi(x) = \arg\min_{\varphi} [L(f, \varphi, \pi_x) + \Omega(\varphi)] \,, \tag{2.1}$$

where we sample a perturbed input $\tilde{x}$ from $\pi_x(\tilde{x}) = exp(-D(x, \tilde{x})^2/\sigma^2)$ taking $D(x, \tilde{x})$ as a distance function and $\sigma$ as the width. $\Omega$ is the model complexity of the proxy $\varphi$. The objective of $\xi(x)$ is to find the simplest $\varphi$ that can approximate the behavior of $f$ around $x$. When using a linear classifier $\boldsymbol{\phi}$ as the $\varphi$ to minimize $\Omega(\varphi)$ [31], we can formulate the objective function as:

$$\boldsymbol{\phi} = \arg\min_{\boldsymbol{\phi}} E_{\tilde{x}\sim\pi_x}(P_\theta(Y = y|\tilde{x}) - \boldsymbol{\phi}^T \mathbf{z})^2 \,, \tag{2.2}$$

where $\mathbf{z} \in \{0, 1\}^M$ is a simplified feature vector of $\tilde{x}$ by a mapping function $h$ such that $\mathbf{z} = h(x, \tilde{x}) = \{\mathbb{1}(x_i \in \tilde{x})\}_{i=1}^M$. Equation 2.2 minimizes the classification error in the neighborhood of $x$ sampled from $\pi_x$. Therefore, using LIME, we can find an interpretable linear model that approximates any complex classifier's behavior around an example $x$.

**Shapley value** takes the input features $x = \{x_i\}_{i=1}^M$ as $M$ independent players who cooperate to achieve a benefit in a game [61]. The Shapley value computes how much each player $x_i$ contributes to the total received benefit:

$$\varphi_i(x) = \sum_{\tilde{x}\subseteq x\backslash\{x_i\}} \frac{|\tilde{x}|!(|x| - |\tilde{x}| - 1)!}{|x|!} [P_\theta(Y = y|\tilde{x} \cup \{x_i\}) - P_\theta(Y = y|\tilde{x})] \,. \tag{2.3}$$

To reduce the computational cost, instead of computing all combinations, we can find surrogates $\phi_i$ proportional to $\varphi_i$ and rewrite the above equation as an expectation over $x$ sampled from $P(\tilde{x})$:

$$\phi_i = \frac{|x|}{|x| - 1}\varphi_i = E_{\tilde{x}\sim P(\tilde{x})}[P_\theta(Y = y|\tilde{x} \cup \{x_i\}) - P_\theta(Y = y|\tilde{x})], \forall i \,, \tag{2.4}$$

where $P(\tilde{x}) = \frac{1}{(|x|-1)\binom{|x|-1}{|\tilde{x}|}}$ is the perturb function.[1] We can also transform the above formulation into argmin:

$$\phi_i = \arg\min_{\phi_i} E_{\tilde{x}\sim P(\tilde{x})}([P_\theta(Y = y|\tilde{x} \cup \{x_i\}) - P_\theta(Y = y|\tilde{x})] - \phi_i)^2 \,. \tag{2.5}$$

[1] $\sum_{\tilde{x}\subseteq x\backslash\{x_i\}} P(\tilde{x}) = \frac{1}{(|x|-1)} \sum_{\tilde{x}\subseteq x\backslash\{x_i\}} 1/\binom{|x|-1}{|\tilde{x}|} = \frac{1}{(|x|-1)} \sum_{|\tilde{x}|} \binom{|x|-1}{|\tilde{x}|}/\binom{|x|-1}{|\tilde{x}|} = \frac{(|x|-1)}{(|x|-1)} = 1$. This affirms that the $P(\tilde{x})$ is a valid probability mass function.

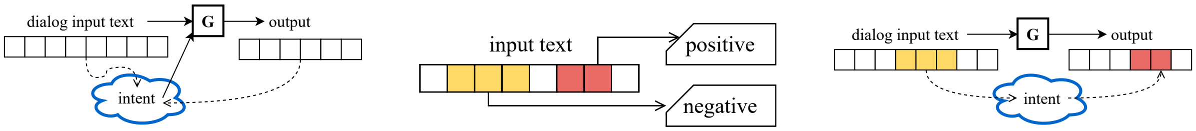


Figure 2.1: The motivation of local explanation for dialogue response generation: (Left) Controllable dialogue models, (Middle) Explanation of classifier, and (Right) = (Left)+(Middle) Our concept is to identify the most salient pair of segments in the input and output, which represents a certain intent of the model's response.

## 2.3 Local Explanation for Conversational Language Models

We aim to explain a model's response prediction to a dialogue history one at a time and call it the *local explanation of dialogue response generation*. We focus on the local explanation for a more fine-grained understanding of the model's behavior.

### 2.3.1 Task Definition

As depicted in Figure 2.1, we draw inspiration from the notions of controllable dialogue generation models (Figure 2.1 (Left)) and local explanation in sentiment analysis (Figure 2.1 (Middle)). The first one uses a concept in predefined classes as a cause to the response; the latter finds the features that correspond to positive or negative sentiment. We propose to find parts within the input and output texts that are related by an underlying intent (Figure 2.1 (Right)).

We first define the notations for dialogue response generation, which aims to predict a response $y = y_1y_2...y_N$ given an input message $x = x_1x_2...x_M$. $x_i$ is the $i$-th token in sentence $x$ with length $M$ and $y_j$ is the $j$-th token in sentence $y$ with length $N$. To solve this task, a typical sequence-to-sequence model $f$ parameterized by $\theta$ produces a sequence of probability masses $< P_\theta(y_1|x), P_\theta(y_2|x, y_1), ..., P_\theta(y_N|x, y_{<N}) >$ [6]. The probability of $y$ given $x$ can then be computed as the product of the sequence $P_\theta(y|x) =$

$P_\theta(y_1|x)P_\theta(y_2|x, y_1)...P_\theta(y_N|x, y_{<N})$.

To explain the prediction, we then define a new explanation model $\Phi \in \mathbb{R}^{M \times N}$ where each column $\Phi_j \in \mathbb{R}^M$ linearly approximates single sequential prediction at the $j$-th time step in text generation. To learn the optimal $\Phi$, we sample perturbed inputs $\tilde{x}$ from a distribution centered on the original inputs $x$ through a probability density function $\tilde{x} = \pi(x)$. Finally, we optimize $\Phi$ by ensuring $u(\Phi_j^T z) \approx g(\tilde{x})$ whenever $z$ is a simplified embedding of $\tilde{x}$ by a mapping function $z = h(x, \tilde{x})$, where we define $g$ as the gain function of the target generative model $f$, $u$ as a transform function of $\Phi$ and $z$ and $L$ as the loss function. Note that $z$ can be a vector or a matrix and $g(\cdot)$, $u(\cdot)$ can return a scalar or a vector depending on the used method. Therefore, we unify the local explanations (LIME and Shapley value) under dialogue response generation as:

**Definition 1: A Unified Formulation of Local Explanation for Dialogue Response Generation**

$$\Phi_j = \arg\min_{\Phi_j} L(g(y_j|\tilde{x}, y_{<j}), u(\Phi_j^T h(\tilde{x}))), \text{ for } j = 1, 2, ..., N \,. \tag{2.6}$$

However, direct adaptation of LIME and Shapley value to dialogue response generation fails to consider the complexity of text generation and the diversity of generated examples. We develop disciplines to alleviate these problems.

### 2.3.2 Method

Our proposed method is designed to (1) address the exponential output space and diverse responses built within the dialogue response generation task and (2) compare the importance of segments within both input and output text.

First, considering the exponential output space and diverse responses, recent work often generates responses using sampling, such as the dominant beam search with top-k sampling [66]. The generated response is therefore only a sample from the estimated probability mass distribution over the output space. Further, the samples drawn from the distribution will inherently have built-in errors that accumulate along generation steps [37]. To avoid these errors we instead explain the estimated probability of the ground truth human responses. In this way, we are considering that the dialogue response generation model is estimating the certainty to predict the human response by $P_\theta(y|x)$. Meanwhile, given the nature of the collected dialogue dataset, we observe only one response per sentence, and thus the mapping is deterministic. We denote the data distribution by $P$ and the probability of observing a response $y$ given input $x$ in the dataset by $P(y|x)$. Since the mapping of $x$ and $y$ is deterministic in the dataset, we assume $P(y|x) = 1$.

Second, if we directly apply prior explanation methods of classifiers on sequential generative models, it turns into a One-vs-Rest classification situation for every generation step. This can cause an unfair comparison among generation steps. For example, the impact from a perturbed input on $y_j$ could end up being the largest just because the absolute certainty $P_\theta(y_j|x, y_{<j})$ was large. However, the impact from a perturbed input on each part in the output should be *how much the certainty has changed after perturbation* and *how much the change is compared to other parts.*

Therefore we propose to find explanation in an input-response pair $(x, y)$ by comparing the interactions between segments in $(x, y)$. To identify the most salient interaction pair $(x_i, y_j)$ (the $i$-th segment in $x$ and the $j$-th segment in $y$), we anticipate that a perturbation $\tilde{x}$ impacts the $j$-th part most in $y$ if it causes

$$D(P_\theta(y_j|\tilde{x}, y_{<j})||P_\theta(y_j|x, y_{<j})) > D(P_\theta(y_{j'}|\tilde{x}, y_{<j'})||P_\theta(y_{j'}|x, y_{<j'})), \forall j' \neq j \, , \tag{2.7}$$

where $D$ represents a distance function measuring the difference between two probability masses. After finding the different part $x_i$ in $x$ and $\tilde{x}$, we then define an existing salient interaction in $(x, y)$ is $(x_i, y_j)$.

In this work, we replace the distance function $D$ in Equation 2.7 with Kullback–Leibler divergence ($D_{KL}$) [67]. However, since we reduce the complexity by considering $P_\theta(y|x)$ as the certainty estimation of $y$, we are limited to obtaining only one point in the distribution. We transfer the equation by modeling the estimated joint probability by $\theta$ of $x$ and $y$. We reconsider the joint distributions as $P_\theta(\tilde{x}, y_{\leq j})$ such that $\sum_{\tilde{x},y} P_\theta(\tilde{x}, y_{\leq j}) = 1$ and $q(\tilde{x}, y) = P_{\theta,\pi_{inv}}(\tilde{x}, y_{\leq j}) = P_\theta(x, y)$ such that $\sum_{\tilde{x},y} q(\tilde{x}, y) = \sum_{\tilde{x},y} P_\theta(x, y_{\leq j}) = \sum_{\tilde{x},y} P_{\theta,\pi_{inv}}(\tilde{x}, y_{\leq j}) = 1$ with $\pi_{inv}$ being the inverse function of $\pi$. Therefore,

$$D(P_\theta(\tilde{x}, y_{\leq j})||P_\theta(x, y_{\leq j})) = D_{KL}(P_\theta(\tilde{x}, y_{\leq j})||q(\tilde{x}, y_{\leq j})) = \sum_{y_j} \sum_{\tilde{x}} P_\theta(\tilde{x}, y_{\leq j}) \log \frac{P_\theta(\tilde{x}, y_{\leq j})}{P_\theta(x, y_{\leq j})} . \tag{2.8}$$

Moreover, since we are estimating the certainty of a response $y$ drawn from data distribution, we know that the random variables $\tilde{x}$ is independently drawn from the perturbation model $\pi$. Their independent conditional probabilities are $P(y|x) = 1$ and $\pi(\tilde{x}|x)$. We approximate the multiplier $P_\theta(\tilde{x}, y_{\leq j}) \approx P(\tilde{x}, y_{\leq j}|x) = P(\tilde{x}|x)P(y|x) = \pi(\tilde{x}|x)$. The divergence can be simplified to

$$D(P_\theta(\tilde{x}, y_{\leq j})||P_\theta(x, y_{\leq j})) \approx \sum_{y_j} \sum_{\tilde{x}} \pi(\tilde{x}|x) \log \frac{P_\theta(\tilde{x}, y_{\leq j})}{P_\theta(x, y_{\leq j})} = E_{\tilde{x}\sim\pi(\cdot|x)} \log \frac{P_\theta(\tilde{x}, y_{\leq j})}{P_\theta(x, y_{\leq j})} . \tag{2.9}$$

To meet the inequality for all $j$ and $j' \neq j$, we estimate each value $\Phi_j^T \mathbf{z}$ in the explanation model $\Phi$ being proportional to the divergence term, where $\mathbf{z} = h(x, \tilde{x}) = \{\mathbb{1}(x_i \in \tilde{x})\}_{i=1}^{M}$. It turns out to be re-estimating the distinct of the chosen segment $y_j$ by normalizing over its original predicted probability.

$$\Phi_j^T \mathbf{z} \propto E_{\tilde{x} \subseteq x \backslash \{x_i\}} D(P_\theta(\tilde{x}, y_{\leq j}) || P_\theta(x, y_{\leq j})) \approx E_{\tilde{x}, \tilde{x} \subseteq x \backslash \{x_i\}} \log \frac{P_\theta(\tilde{x}, y_{\leq j})}{P_\theta(x, y_{\leq j})} . \tag{2.10}$$

We propose two variations to optimize $\Phi$ following the unified formulation defined in Equation 2.6.

First, since logarithm is strictly increasing, so to get the same order of $\Phi_{ij}$, we can drop off the logarithmic term in Equation 2.10. After reducing the non-linear factor, we use mean square error as the loss function. With the gain function $g = \frac{P_\theta(\tilde{x}, y_{\leq j})}{P_\theta(x, y_{\leq j})}$, the optimization equation becomes

$$\Phi_j = \arg\min_{\Phi_j} E_{P(\tilde{x})} (\frac{P_\theta(\tilde{x}, y_{\leq j})}{P_\theta(x, y_{\leq j})} - \Phi_j^T \mathbf{z})^2, \forall j . \tag{2.11}$$

We call this variation as LERG_L in Algorithm 1, since this optimization is similar to LIME but differs by the gain function being a ratio.

To derive the second variation, we suppose an optimized $\Phi$ exists and is denoted by $\Phi^*$, we can write that for every $\tilde{x}$ and its correspondent $\mathbf{z} = h(x, \tilde{x})$,

$$\Phi_j^* \mathbf{z} = \log \frac{P_\theta(\tilde{x}, y_{\leq j})}{P_\theta(x, y_{\leq j})} . \tag{2.12}$$

We can then find the formal representation of $\Phi_{ij}^*$ by

$$\begin{aligned} \Phi_{ij}^* &= \Phi_j^* \mathbf{1} - \Phi_j^* \mathbf{1}_{i=0} \\ &= \Phi_j^* (\mathbf{z} + e_i) - \Phi_j^* \mathbf{z}, \forall \tilde{x} \in x \backslash \{x_i\} \text{ and } \mathbf{z} = h(x, \tilde{x}) \\ &= E_{\tilde{x} \in x \backslash \{x_i\}} [\Phi_j^* (\mathbf{z} + e_i) - \Phi_j^* \mathbf{z}] \\ &= E_{\tilde{x} \in x \backslash \{x_i\}} [\log P_\theta(y_j | \tilde{x} \cup \{x_i\}, y_{<j}) - \log P_\theta(y_j | \tilde{x}, y_{<j})] \end{aligned} \tag{2.13}$$

**Algorithm 1:** Local Explanation of Response Generation

```
Input: input message $x = x_1x_2...x_M$, ground-truth response $y = y_1y_2...y_N$
Input: a response generation model $\theta$ to be explained
Input: a local explanation model parameterized by $\Phi$
// 1st variation – LERG_L
for each iteration do
    sample a batch of $\tilde{x}$ perturbed from $\pi(x)$
    map $\tilde{x}$ to $z = \{0, 1\}_1^M$
    compute gold probability $P_\theta(y_j|x, y_{<j})$
    compute perturbed probability $P_\theta(y_j|\tilde{x}, y_{<j})$
    optimize $\Phi$ to minimize loss function
        $L = \sum_j \sum_{\tilde{x}} (\frac{P_\theta(y_j|\tilde{x}, y_{<j})}{P_\theta(y_j|x, y_{<j})} - \Phi_j^T \mathbf{z})^2$
// 2nd variation - LERG_S
for each $i$ do
    sample a batch of $\tilde{x}$ perturbed from $\pi(x \backslash \{x_i\})$
    $\Phi_{ij} = \frac{1}{m} \sum_{\tilde{x}} \log P_\theta(y_j|\tilde{x} \cup \{x_i\}, y_{<j}) - \log P_\theta(y_j|\tilde{x}, y_{<j})$, for $\forall j$
return $\Phi_{ij}$, for $\forall i, j$
```

We call this variation as LERG_S in Algorithm 1, since this optimization is similar to Shapley value but differs by the gain function being the difference of logarithm. To further reduce computations, we use Monte Carlo sampling with $m$ examples as a sampling version of Shapley value [68].

### 2.3.3 Properties

We propose that an explanation of dialogue response generation should adhere to three properties to prove itself faithful to the generative model and understandable to humans.

**Property 1: unbiased approximation** *To ensure the explanation model $\Phi$ explains the benefits of picking the sentence $y$, the summation of all elements in $\Phi$ should approximate the difference between the certainty of $y$ given $x$ and without $x$ (the language* *modeling of y).*

$$\sum_j \sum_i \Phi_{ij} \approx \log P(y|x) - \log P(y)\,. \tag{2.14}$$

**Property 2: consistency** *To ensure the explanation model* $\Phi$ *consistently explains different generation steps* $j$*, given a distance function if* $\forall j', \forall \tilde{x} \in x \backslash \{x_i\}$

$$D(P_\theta(y_j|\tilde{x}, y_{<j}), P_\theta(y_j|\tilde{x} \cup \{x_i\}, y_{<j})) > D(P_\theta(y_{j'}|\tilde{x}, y_{<j'}), P_\theta(y_{j'}|\tilde{x} \cup \{x_i\}, y_{<j'}))\,, \tag{2.15}$$

then $\Phi_{ij} > \Phi_{ij'}$.

**Property 3: cause identification** *To ensure that the explanation model sorts different input features by their importance to the results, if*

$$g(y_j|\tilde{x} \cup \{x_i\}) > g(y_j|\tilde{x} \cup \{x_i'\}), \forall \tilde{x} \in x \backslash \{x_i, x_i'\}\,, \tag{2.16}$$

then $\Phi_{ij} > \Phi_{i'j}$

Meanwhile Shapley value follows Properties 2 and 3, while LIME follows Property 3 when an optimized solution exists. These properties also demonstrate that our method approximates the text generation process while sorting out the important segments in both the input and output texts. This could be the reason to serve as explanations to any sequential generative model.

## 2.4 Necessity and Sufficiency of Explanation

Explanation is notoriously hard to evaluate even for digits and sentiment classification which are generally more intuitive than *explaining response generation.* For digit classification (MNIST), explanations often mark the key curves in figures that can identify digit numbers. For sentiment analysis, explanations often mark the positive and negative words in text. Unlike them, we focus on identifying the key parts in both input messages and their responses. Our move requires an explanation include the interactions of the input and output features.

To evaluate the defined explanation, we quantify the necessity and sufficiency of explanations towards a model's uncertainty of a response. We evaluate these aspects by answering the following questions.

- **necessity:** How is the model influenced after removing explanations?
- **sufficiency:** How does the model perform when only the explanations are given?

Furthermore, we conduct a user study to judge human understandings of the explanations to gauge how trustworthy the dialog agents are.

### 2.4.1 Dataset, Models, Methods

We evaluate our method over chit-chat dialogues for their more complex and realistic conversations. We specifically select and study a popular conversational dataset called DailyDialog [69] because its dialogues are based on daily topics and have less uninformative responses.Due to the large variation of topics, open-ended nature of conversations and informative responses within this dataset, explaining dialogue response generation models trained on DailyDialog is challenging but accessible.

We fine-tune a GPT-based language model [12, 70] and a DialoGPT [15] on DailyDialog by minimizing the following loss function:

$$L = -\sum_{m}\sum_{j} \log P_\theta(y_j|x, y_{<j})\,, \tag{2.17}$$

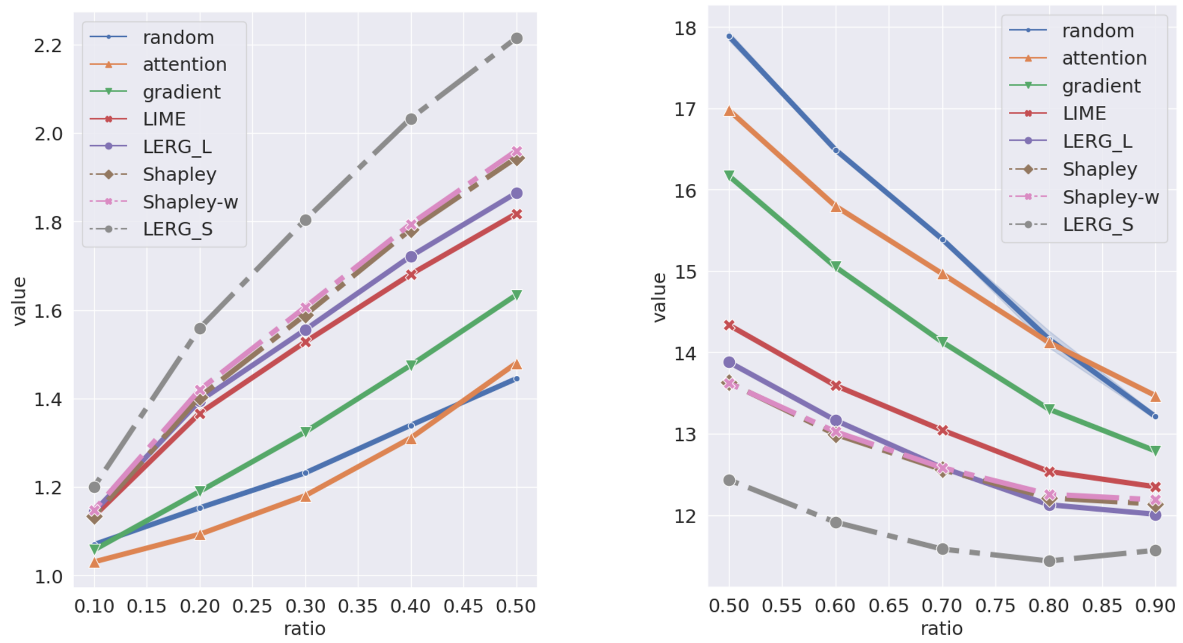


Figure 2.2: Results of GPT fine-tuned on DailyDialog: (Left) $PPLC_R$ (Right) $PPL_A$.

where $\theta$ is the model's parameter. We train until the loss converges on both models and achieve fairly low test perplexities compared to [69]: 12.35 and 11.83 respectively. The low perplexities demonstrate that the models are more likely to be rationale and therefore, evaluating explanations over these models will be more meaningful and interpretable.

We compare our explanations LERG_L and LERG_S with attention [71], gradient [72], LIME [31] and Shapley value [73]. We use sample mean for Shapley value to avoid massive computations (Shapley for short), and drop the weights in Shapley value (Shapley-w for short) due to the intuition that not all permutations should exist in natural language [74, 75]. Our comparison is fair since all methods requiring permutation samples utilize the same amount of samples.

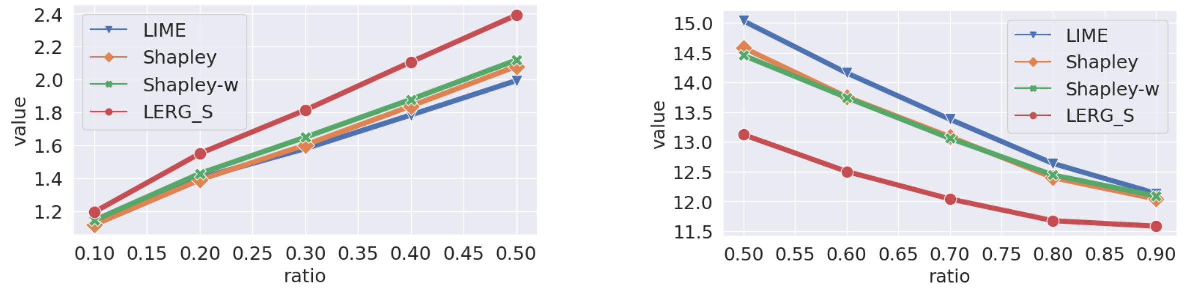


Figure 2.3: Results of DialoGPT fine-tuned on DailyDialog: (Left) $PPLC_R$ (Right) $PPL_A$.

### 2.4.2 Necessity: How is the model influenced after removing explanations?

Assessing the correctness of estimated important feature relevance requires labeled features for each model and example pair, which is rarely accessible. Inspired by [56, 76] who removes the estimated salient features and observe how the performance changes, we introduce the notion *necessity* that extends their idea. We quantify the necessity of the estimated salient input features to the uncertainty estimation of response generation by *perplexity change of removal* ($PPLC_R$), defined as:

$$PPLC_R := exp^{\frac{1}{m}[-\sum_j \log P_\theta(y_j|x_R,y_{<j})+\sum_j \log P_\theta(y_j|x,y_{<j})]}, \tag{2.18}$$

where $x_R$ is the remaining sequence after removing top-k% salient input features.

As shown in Figure 2.4.1 and Figure 2.4.1[2], removing larger number of input features consistently causes the monotonically increasing $PPLC_R$. Therefore, to reduce the factor that the $PPLC_R$ is caused by, the removal ratio, we compare all methods with an additional baseline that *randomly* removes features. LERG_S and LERG_L both outperform their counterparts Shapley-w and LIME by 12.8% and 2.2% respectively. We further

[2] We did a z-test and a t-test [77] with the null hypothesis between LERG_L and LIME (and LERG_S and Shapley). For both settings the p-value was less than 0.001, indicating that the proposed methods significantly outperform the baselines.

observe that Shapley-w outperforms the LERG_L. We hypothesize that this is because LERG_L and LIME do not reach an optimal state.

### 2.4.3 Sufficiency: How does the model perform when only the explanations are given?

Even though necessity can test whether the selected features are crucial to the model's prediction, it lacks to validate how possible the explanation itself can determine a response. A complete explanation is able to recover model's prediction without the original input. We name this notion as *sufficiency* testing and formalize the idea as:

$$PPL_A := exp^{-\frac{1}{m}\sum_j \log P_\theta(y_j|x_A,y_{<j})}, \tag{2.19}$$

where $x_A$ is the sequential concatenation of the top-k% salient input features.

As shown in Figure 2.4.1 and Figure 2.4.1, removing larger number of input features gets the $PPL_A$ closer to the perplexity of using all input features 12.35 and 11.83. We again adopt a random baseline to compare. LERG_S and LERG_L again outperform their counterparts Shapley-w and LIME by 5.1% and 3.4% respectively. Furthermore, we found that LERG_S is able to go lower than the original 12.35 and 11.83 perplexities. This result indicates that LERG_S is able to identify the most relevant features while avoiding features that cause more uncertainty during prediction.

## 2.5 User Study

To ensure the explanation is easy-to-understand by non machine learning experts and gives users insights into the model, we resort to user study to answer the question: "If an explanation can be understood by users to respond?"

Table 2.1: Confidence (1-5) with 1 denotes *not confident* and 5 denotes *highly confident*.

| Method | Acc | Conf |
|---|---|---|
| Random | 36.15 | 3.00 |
| Attention | 34.75 | 2.81 |
| Gradient | 42.52 | 2.97 |
| LIME | 46.37 | 3.26 |
| LERG_L | 47.97 | 3.24 |
| Shapley-w | 53.65 | 3.20 |
| LERG_S | **56.03** | **3.35** |

We ask human judges to compare explanation methods. Instead of asking judges to annotate their explanation for each dialogue, to increase their agreements we present only the explanations (Top 20% features) and ask them to choose from four response candidates, where one is the ground-truth, two are randomly sampled from other dialogues, and the last one is randomly sampled from other turns in the same dialogue. Therefore the questionnaire requires human to interpret the explanations but not guess a response that has word overlap with the explanation. The higher accuracy indicates the higher quality of explanations. To conduct more valid human evaluation, we randomly sample 200 conversations with sufficiently long input prompt (length$\geq$ 10). This way it filters out possibly non-explainable dialogues that can cause ambiguities to annotators and make human evaluation less reliable.

We employ three workers on Amazon Mechanical Turk [78] [3] for each method of each conversation, resulting in total 600 annotations. Besides the multiple choice questions, we also ask judges to claim their confidences of their choices. The results are listed in Table 2.1. We observe that LERG_L performs slightly better than LIME in accuracy while maintaining similar annotator's confidence. LERG_S significantly outperforms Shapley-w in both accuracy and annotators' confidence. Moreover, these results indicates that when presenting users with only 20% of the tokens they are able to achieve 56% accuracy

[3] `https://www.mturk.com`

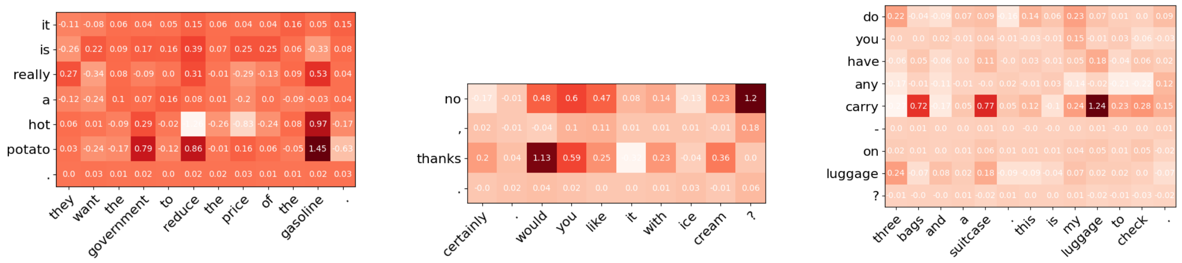


Figure 2.4: Two major categories of local explanation except word alignment and one typical error. The horizontal text is the input prompt and the vertical text is the response. (Left) Implication: find the ”hot potato” might indicate ”gasoline”. (Middle) Sociability: find ”No” for the ”question mark” and ”thanks” for the ”would like”, the polite way to say ”want”. (Right) Error analysis: related but not the best.

while a random selection is around 25%.

We further analyzed the extracted explanation for each dialogue. We found that these fine-grained level explanations can be split into three major categories: implication / meaning, sociability, and one-to-one word mapping. As shown in Figure 2.5, the “hot potato” in response implies the phenomenon of “reduce the price of gasoline”. On the other hand, Figure 2.5 demonstrates that a response with sociability can sense the politeness and responds with “thanks”. We ignore word-to-word mapping here since it is intuitive and can already be successfully detected by attention models. Figure 2.5 shows a typical error that our explanation methods can produce. As depicted, the word “carry” is related to “bags”,“suitcases”, and “luggage”. Nonetheless a complete explanation should cluster “carry-on luggages”. The error of explanations can result from (1) the target model or (2) the explanation method. When taking the first view, in future work, we might use explanations as an evaluation method for dialogue generation models where the correct evaluation metrics are still in debates. When taking the second view, we need to understand that these methods are *trying* to explain the model and are not absolutely correct. Hence, we should carefully analyze the explanations and use them as reference and should not fully rely on them.

# Chapter 3

# Training a Model to Have Interpretable Reasoning Process

Besides explaining a language model after it was trained as the Chapter 2, we can build a generation model that reveals its own reasoning process while responding to user's utterances. By improving the reasoning capability, this revealing can also soothe the issue of limited user experience that users interacting with the assistants need to phrase their requests in a specific manner to elicit an appropriate response. In this chapter, we propose a newly designed transformer model, **d**ialogue di**ff**erentiable **k**nowledge **g**raph model (DiffKG), that is equipped with a human interpretable, large-scale knowledge graph reasoning capability. We investigate the reasoning abilities of DiffKG on both task-oriented and domain-specific chit-chat dialogues. Empirical results show that this method can effectively and efficiently incorporate a knowledge graph into a dialogue system with fully-interpretable reasoning paths.

## 3.1 Introduction

Nowadays, dialogue systems are ubiquitous in customer service and voice-based assistants. One of the main uses of this technology is supporting humans in accomplishing tasks that might require accessing and navigating large knowledge bases (e.g., movies search). A dialogue system architecture is typically composed of a natural language understanding (NLU) module, a dialogue management (DM) module, and a natural language generation (NLG) module [79, 8]. First, the NLU component extracts a meaning representation from the user utterance based on which the DM generates the next system action by reasoning over the meaning representation and communicating with external applications if necessary. For example, the DM may retrieve information from external knowledge graphs (KG) to answer the user's query based on the dialogue history. This process requires the DM to convert the output of NLU to a query to be issued to the backend. Given the difficulty of this step, which is often domain-dependent, the DM component might require the design of hand-crafted rules. However, such rules are usually not scalable to different applications. They could require considerable effort to cover all possible cases/dialogue flows, leading to expensive costs to design new applications. Moreover, in several cases, users interacting with such assistants are forced to formulate specific queries in order to accomplish their objective, which might break user engagement.

To alleviate the problem of having to design expensive hand-crafted rules and breaking user experience, recent works have explored the possibility of building end-to-end dialogue systems [80] and all-in-one response generation models [81]. Among them, since graph is one of the main structure to store knowledge, recent research [26, 82, 83, 27, 84] has proposed methods to generate natural language responses according to both the dialogue history and external knowledge graph. Despite these innovative and inspiring methods, there are some shortcomings. For instance, these methods are either not fully-interpretable or limited to small-scale knowledge graphs.

In this chapter, we propose a novel dialogue differentiable knowledge graph model (DiffKG). The DiffKG is a single transformer model that directly (1) generates a sequence of relations to perform multi-hop reasoning on a reified KG representation proposed by [85], and then (2) generates responses using the retrieved entities. To the best of our knowledge, this is the first dialogue model that can directly walk on a large-scale KG with flexibility and interpretability. DiffKG allows having flexible entity values in the KG and handling novel entity values with an arbitrarily defined number of tokens. The reasoning path of DiffKG consists of the predicted relations, thus allowing for transparency.

We run extensive experiments to test DiffKG performance on KG-grounded dialogues. We select Stanford Multi-domain Dialogues (SMD) [25] and propose a new dataset, SMD-Reasoning, to simulate scenarios requiring multiple reasoning types and select the OpenDialKG [83] to simulate scenarios requiring large-scale KG reasoning without preprocessing. We then compare DiffKG with state-of-the-art models on SMD and OpenDialKG and an additional baseline that flattens KGs into a textual form from which transformers can learn. Empirically, our experiments show that DiffKG can effectively be trained on large-scale KGs and demonstrate its robustness with modified triplets in a KG. From the perspective of computation, DiffKG leads to relatively low extra time and memory usage compared to transformer models not using any KG information.

In summary, our contributions are: 1) We propose DiffKG, a novel method that can effectively and flexibly incorporate large-scale KG; 2) We demonstrate that DiffKG is a model-agnostic method and can be applied to different model architectures; 3) We show that DiffKG is an interpretable method with low add-on latency at inference time.

## 3.2 Background: Knowledge Graph Grounded Response Generation

### 3.2.1 Knowledge Graph

We assume that the knowledge of the system can be represented by a knowledge graph (KG) $\mathcal{G} = \{\mathcal{E}, \mathcal{R}\}$, where $\mathcal{E}$ denotes the entities and $\mathcal{R}$ denotes the relations. The knowledge graph $\mathcal{G}$ contains multiple triples describing the connections among entities and relations. We denote the $k$-th triple of this graph as $(e_k^h, r_k, e_k^t)$ , where $e_k^h$, $r_k$, $e_k^t$ are respectively the head entity, relation, and tail entity. The total numbers of triples, entities, and relations are denoted as $N_\mathcal{T}$, $N_\mathcal{E}$, $N_\mathcal{R}$, respectively.[1]

### 3.2.2 Grounded Response Generation

If we define the dialogue history as a sequence of tokens that occurred during the user and system interactions, then a flattened dialogue history can be written as:

$$\mathbf{x} = (x_1, x_2, ..., x_m, ..., x_M) \tag{3.1}$$

where $x_m$ is the $m$-th token in the dialogue history with $M$ tokens. In an end-to-end dialogue system, we assume a dialogue system parameterized by $\theta$ exists that can predict a probability distribution of responses $P_\theta(\cdot|\mathbf{x}, \mathcal{G})$. The generated responses are sampled from this probability distribution.

[1] An example of the triples in $\mathcal{G}$ is a triple $e_k^h =$ gas station, $r_k =$ IsTypeOf, and $e_k^t =$ Chevron. That is, “gas station is the type of Chevron” to this system.

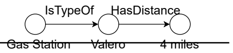
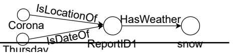
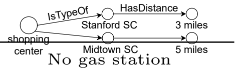
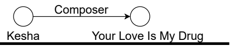
| Reasoning Type | | | Example | Related Info. in KG |
|---|---|---|---|---|
| Semantic Form | KG reasoning | | U: I need unleaded gas.<br>R: inform Valero, 4 miles |  |
| | Logical Reasoning | True/False | U: Is it going to snow this week at Corona?<br>R: Yes |  IsLocationOf<br>Corona<br>IsDateOf<br>Thursday<br>HasWeather<br>ReportID1<br>snow |
| | | Selection | U: give me the direction to the nearest shopping mall.<br>R: inform Stanford Shopping Center, 3 miles |  IsTypeOf<br>HasDistance<br>Stanford SC<br>3 miles<br>shopping center<br>Midtown SC<br>5 miles |
| | | Extraction | U: What gas stations are here?<br>R: include poi_type gas station | No gas station in the available KG |
| NL Form | KG reasoning | | U: Have you listen to any of the singer Kesha's song?<br>R: I do enjoy in her music, especially "Your Love Is My Drug" |  |

Table 3.1: Example of different reasoning types and output formats (semantic and natural language forms) in a dialogue system with the related information in the accessible KGs.

### 3.2.3 Problem Statement

We focus on understanding the ability of language models in performing reasoning during a conversation. We consider two tasks that are usually required in dialogue scenarios and call them semantic form and natural language (NL) form in Table 3.1. First, given a dialogue history and a user's query, the task of semantic form is to predict the next system action, corresponding to the output of the DM module, based on the available knowledge. In this case, we assume the expected output is the essential knowledge for an NLG module. We argue that this task could help better evaluate if the response is correct or not and which type of reasoning can be more successfully handled. Second, given a dialogue history and a user's query, the task of the NL form could be to directly output the response given by the system. This setting with annotated reasoning path can shed light on understanding if the model can learn to support chit-chat and reasoning at the same time.

Moreover, we aim to understand models' reasoning capability both in the form of logical reasoning and over the provided knowledge. As illustrated in Table 3.1, by KG reasoning, we refer to the ability of the model to retrieve information from an arbitrary scaled KG in multiple hops. Meanwhile, we refer to logical reasoning as the ability of the model to conduct operations such as evaluating whether a statement is true or false,

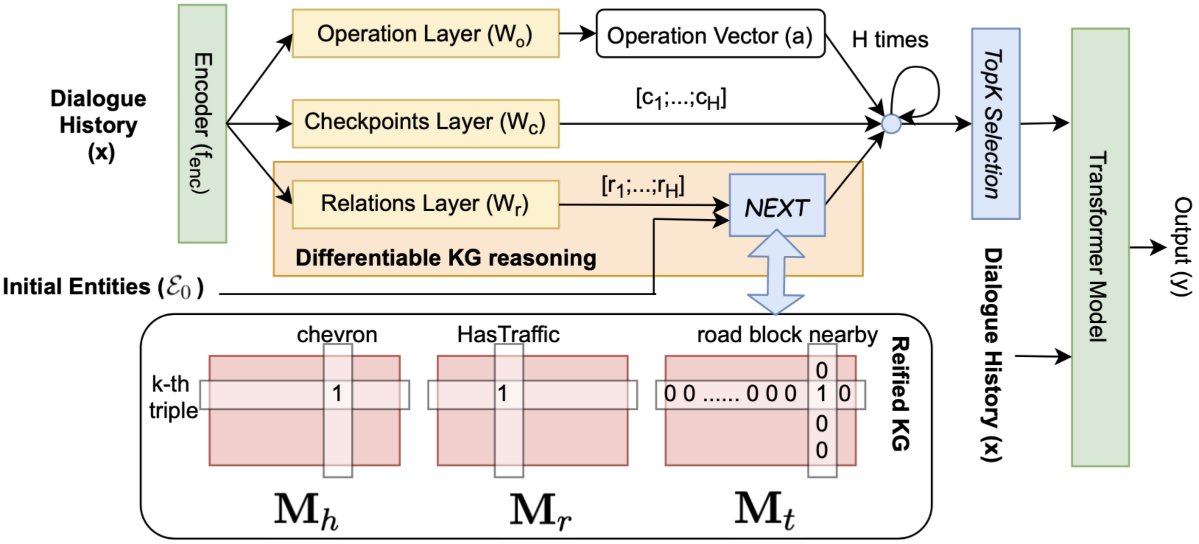


Figure 3.1: The illustration of proposed DiffKG, which leverages a pretrained transformer model (T5 or GPT2) and the Reified KG. The model generates the response depending on the predicted relation sequence $[r_1; ...; r_H]$, thus being fully interpretable in terms of the used reasoning path.

selecting min/max from a list of alternatives, and extracting constraints.

We formulate the task that we focus on as follows: given the dialogue history $\mathbf{x}$ and currently accessible KG $\mathcal{G}$, can we extend a transformer model to predict a correct response $y$ in either semantic or NL form? As illustrated in Table 3.1, this task not only requires the model to accurately retrieve information from the KG, but also needs to do further logical operations on the information. To solve this task, a model should also be able to effectively integrate the dialogue history $\mathbf{x}$ with the KG $\mathcal{G}$.

## 3.3 Enhanced Transformer: Dialogue Differentiable Knowledge Graph Model

Figure 3.1 illustrates our proposed architecture which contains four main parts: a dialogue history encoder, a differentiable KG reasoning module, a learnable logical operations module, and a response decoder (the transformer model). Note that we experiment with two types of transformers: a causal language model GPT2 [86] and an encoder-decoder model T5 [14]. For GPT2, we reuse the same encoder that is used at the beginning of the process, i.e., $f_{enc}$ in Figure 1, as the final transformer that generates the response token by token. For T5, we reuse the same encoder as the encoder of the final transformer with a separate decoder that generates the response. Therefore, this method contains a single transformer model. In following sections we present each module in detail.

### 3.3.1 Dialogue History Encoder

We use encoder model to project $\mathbf{x}$ and obtain the dialogue history embedding through $\tilde{\mathbf{x}} = f_{enc}(\mathbf{x}) \in \mathbb{R}^d$, where $d$ is the hidden size of the encoder. The embedding $\tilde{\mathbf{x}}$ is first fed into an operation layer with parameters $\mathbf{W}_o \in \mathbb{R}^{d \times d}$. The operation layer predicts the operation vector $\mathbf{a} = \mathbf{W}_o^T \tilde{\mathbf{x}} \in \mathbb{R}^d$. At the same time, the embedding $\tilde{\mathbf{x}}$ is also fed into a relation layer with parameters $\mathbf{W}_r \in \mathbb{R}^{d \times N_{\mathcal{R}} H}$. The relation layer predicts the concatenation of a sequence of relations $\mathbf{r} = \{\mathbf{r}_h | 1 \leq h \leq H\}$, where $\mathbf{r}_h \in \mathbb{R}^{N_{\mathcal{R}}}$ is the relation to be used at the $h$-th hop in the programmed walking block and $H$ is the maximum number of hops. The embedding $\tilde{\mathbf{x}}$ is also fed into a checkpoints layer with parameters $\mathbf{W}_c \in \mathbb{R}^{d \times 2H}$. This layer produces the concatenation of a sequence of walk-or-check vectors $\mathbf{c} = \{c_h | 1 \leq h \leq H\}$, where $c_h \in \mathbb{R}^2$ is the walk-or-check vector at the $h$-th hop to determine the weights of the programmed walking module and the operation vector.

$$\begin{aligned}
\tilde{\mathbf{x}} &= f_{enc}(\mathbf{x})\,, \\
\mathbf{a} &= \mathbf{W}_o^T\tilde{\mathbf{x}}\,, \\
\mathbf{r} &= \mathbf{W}_r^T\tilde{\mathbf{x}}\,, \\
\mathbf{c} &= \mathtt{softmax}(\mathbf{W}_c^T\tilde{\mathbf{x}})\,.
\end{aligned} \tag{3.2}$$

### Differential Knowledge Graph Reasoning

To ensure that our model can scale to larger KGs, we adopt the reified KG representation proposed by [85]. The reified KG represents the graph $\mathcal{G}$ using three sparse matrices: head matrix $\mathbf{M}_h \in \mathbb{R}^{N_\mathcal{T} \times N_\mathcal{E}}$, relation matrix $\mathbf{M}_r \in \mathbb{R}^{N_\mathcal{T} \times N_\mathcal{R}}$, and tail matrix $\mathbf{M}_t \in \mathbb{R}^{N_\mathcal{T} \times N_\mathcal{E}}$. An entry $(i, e)$ in $\mathbf{M}_h$ or $\mathbf{M}_t$ with value 1 indicates that the $i$-th triple in the KG has entity $e$ as the head or the tail; an entry $(i, r)$ in $\mathbf{M}_r$ with value 1 indicates that the $i$-th triple in the knowledge graph has the relation $r$. Since often in practical settings most entries in the three matrices are zero, saving them into sparse matrices can significantly reduce memory consumption [85].

After predicting the relation sequence $\mathbf{r}$, we start the graph traversal from a given set of initial entities $\mathcal{E}_0 \subseteq \mathcal{E}$. We first map the initial entities into a vector $\mathbf{e}_1 = [\mathbb{1}(e \in \mathcal{E}_0), \forall e \in \mathcal{E}]$. That is, each entry of $\mathbf{e}_1 \in \mathbb{R}^{N_\mathcal{E}}$ has value 1 if that entity is in the initial entities list $\mathcal{E}_0$, otherwise, the entry is zero. We then predict the next (temporary) entity vector $\mathbf{e}_2$ by conducting a `Next` module:

$$\mathbf{e}_{h+1}^r = \mathtt{Next}(\mathbf{e}_h, \mathbf{r}_h)\,, \tag{3.3}$$

where

$$\mathtt{Next}(\mathbf{e}_h, \mathbf{r}_h) = \frac{\mathbf{M}_t^T(\mathbf{M}_h\mathbf{e}_h \odot \mathbf{M}_r\mathbf{r}_h)}{||\mathbf{M}_t^T(\mathbf{M}_h\mathbf{e}_h \odot \mathbf{M}_r\mathbf{r}_h)||_2 + \epsilon}\,, \tag{3.4}$$

Here $\epsilon$ is an arbitrary small number to offset the denominator and prevent division by zero. We introduce the normalized `Next` to solve the issue with the method proposed by [85] for knowledge graph completion defined as $\texttt{Follow}(\mathbf{e}_h, \mathbf{r}_h) = \mathbf{M}_t^T(\mathbf{M}_h\mathbf{e}_h \odot \mathbf{M}_r\mathbf{r}_h)$; since in a dialogue model, we can seldom predict the relation vectors that perfectly match the entity vectors. That is, if directly using the `Follow` module in [85], the $||\mathbf{e}_h||_2$ will not be one and will vanish as the hop number $h$ increases. Specifically, note that in our proposed module, the predicted relations $\mathbf{r}_h$ are independent of the traversed entities $\mathbf{e}_h$. For instance, finding the "distance" of "the nearby gas station" is independent of whether the nearby gas station is "Chevron" or "Shell".

To allow the model to dynamically select the number of reasoning hops, we add a relation type "ToSelf" into $\mathcal{R}$ and connect each entity to itself by "ToSelf". More specifically, the KG will contain triples $(e_k^h, r_k, e_k^t)$ for all $e_k^h = e_k^t \in \mathcal{E}$ and $r_k = \texttt{ToSelf}$.

### 3.3.2 Entity Embeddings

At each hop, we further conduct the operation vectors $\mathbf{a}$ on the entities weighted by the entity vector $\mathbf{e}_h$. First, we tokenize each entity and represent it by the concatenation of its token embeddings. This step allows (1) representing entities with longer texts such as phrases and sentences, and (2) eliminating the effort to retrain entity embeddings whenever new entity values are added. The entity embeddings can then be represented as a tensor $\mathbf{E} \in \mathbb{R}^{N_{\mathcal{E}} \times d \times m}$, where $m$ is the maximum number of tokens of entities[2].

### 3.3.3 Learnable Logical Operation and Checkpoints

We compute the transformed entity embeddings by element-wise multiplication of the entity embeddings $\mathbf{E}$ with the entity vector $\mathbf{e}_h$ at the $h$-th hop. Next, the dot product

[2] In our experiments, we compute the maximum length of all entities and pad shorter entities to the length of $m$.

of the operation vectors and the transformed entity embeddings is passed to a softmax layer as the entity vector at the next hop:

$$\mathbf{e}^{a}_{h+1} = \texttt{softmax}\left(\mathbf{a}(\mathbf{E} \odot \mathbf{e}_h)\right) \ , \tag{3.5}$$

Further, at the $h$-th hop we use the walk-or-check vector $\mathbf{c}_h$ to combine the `Next` and operation modules above. The combined entity vector is given by:

$$\begin{aligned} \mathbf{e}_{h+1} &= \mathbf{c}_h^T \begin{bmatrix} \mathbf{e}^{r}_{h+1} \\ \mathbf{e}^{a}_{h+1} \end{bmatrix} \\ &= \mathbf{c}_h^T \begin{bmatrix} \texttt{Next}(\mathbf{e}_h, \mathbf{r}_h) \\ \texttt{softmax}\left(\mathbf{a}(\mathbf{E} \odot \mathbf{e}_h)\right) \end{bmatrix} \ , \end{aligned} \tag{3.6}$$

### 3.3.4 Response Decoder

After $H$ hops reasoning is done, the entities with top-$k$ values in the entity vector $\mathbf{e}_H$ are selected, indicating that they have the highest probability to be retrieved from the graph. These entities are converted into their embeddings in $E$ and multiplied by their values in $\mathbf{e}_H$. These entity embeddings are then concatenated with the dialogue history $\mathbf{x}$. The concatenated vectors are fed as the input into the transformer model to predict the response token by token. The predicted probability distribution over the output space can be written as $P(\cdot|\mathbf{x}, \mathbf{M}_h, \mathbf{M}_r, \mathbf{M}_t)$. Since all components are differentiable, all modules can be trained end-to-end with the dialogue history $\mathbf{x}$ and the reified KG representation $\{\mathbf{M}_h, \mathbf{M}_r, \mathbf{M}_t\}$ using the cross-entropy loss with the ground-truth output $y$ as the labels.

$$L = \sum_{(\mathbf{x},y)} - \log P(y|\mathbf{x}, \mathbf{M}_h, \mathbf{M}_r, \mathbf{M}_t) \, . \tag{3.7}$$

During the inference time, the reasoning modules (relation layer, operation layer, and checkpoints layer) work exactly the same as the training stage, the only difference is that the response decoder is fed with predicted tokens in previous time steps (inference stage) instead of the ground-truth output (training stage).

## 3.4 Experiments

### 3.4.1 Datasets

We evaluate our proposed approach on three datasets. Among them, we use Stanford Multi-domain Dialogues (SMD) [25] and OpenDialKG [83] to test the methods generalizability on different dialogue types (task-oriented / chit-chat) and scales of structured knowledge (pairwise database / universal KG). To further analyze the reasoning ability, we propose a new dataset [will release], SMD-Reasoning, by modifying the output of SMD dataset from natural language responses to actions paired with their reasoning types.

**Stanford Multi-domain Dialogues (SMD)** The SMD dataset [25] is composed of two-speaker conversations, where a driver talks with the car assistant to tackle tasks in three domains: scheduling, navigation, and weather forecasting. Each dialogue focuses on one domain and is paired with a database having the related information. We convert the original database into two formats: (1) the natural language descriptions (NLD) and (2) the KG. The NLD form allows us to investigate the ability of the model to interpret unstructured knowledge, while the KG form could be a more extensible structured knowledge compared to tables.

**OpenDialKG** OpenDialKG dataset [83] is composed of two-speakers recommendation and chit-chat style conversations. Each turn in a dialogue is annotated with the reasoning path on the provided KG, which is filtered from Freebase [87]. The resulting KG has 1,190,658 triples, 100,813 entities and 1,358 relations. We randomly split 70/15/15% for train/valid/test sets as described in [83, 88] since they do not release their splits.

**SMD-Reasoning** To make SMD dataset suitable for more precise evaluation of reasoning abilities, we manually label and convert it to the SMD-Reasoning dataset. We first remove the natural language part from the original responses and only leave the action word (e.g., inform) along with the information being conveyed. We divide the dataset into three main reasoning types: informing items, selecting min/max, and evaluating true/false. To validate if the models can identify whether the needed knowledge is in the database, we add a new reasoning type for extracting constraints, by removing the needed knowledge from the database and changing the output to "include [knowledge description]" as shown in Table 3.1.

### 3.4.2 Evaluation Metrics

We use different evaluation methods for the three datasets. For SMD, we follow prior work [84] and use BLEU [89], and Entity F1 scores on each domain. For OpenDialKG, we follow the descriptions in prior works [83, 88] to evaluate the path@k scores, i.e., if the ground-truth path is ranked top-k in the predicted paths probabilities. Moreover, since our method not only can predict the reasoning path as prior works but also can predict the response, we also use the BLEU score to get the approximated evaluation of the response quality compared to ground-truth. Note that prior work has discussed that BLEU scores may not match human intuition [90], but we use them here as an approximated evaluation for reference.

For SMD-Reasoning, the output is more deterministic and does not include diverse sentence structures. Therefore, we compute the F1 score and the *exact match (EM)* score of prediction and the ground-truth. The EM score is calculated by removing the order of the prediction since the labels of SMD-Reasoning dataset follow the order of knowledge description appearing in the original ground-truth responses and may not have the same order as generated outputs. The EM score can be written as:

$$\texttt{EM} = \frac{1}{T} \sum \mathbb{1}(\texttt{sort}(\hat{y}) = \texttt{sort}(y)) \,. \tag{3.8}$$

where $\hat{y}$ is inferred from the model using argmax sampling and $T$ is total number of examples.

### 3.4.3 Implementation Details

Since the proposed method is model-agnostic, we implement it on GPT2 [86] and T5 [14]. Specifically for the T5 model, we use the unifiedQA-T5 model [91] which is pretrained on question answering tasks that also need to do reasoning. However, we empirically find that T5 generally has better performance than GPT2, thus using T5 model in most experiments.

### 3.4.4 Baselines

We compare our proposed DiffKG model with the state-of-the-art models on OpenDialKG reported in [83, 88] and the state-of-the-art graph-based model on SMD [84, 92] with their reported baselines including sequence-to-sequence models with and without attention (S2S and S2S+Attn) [93], pointer to unknown (Ptr-Unk) [94], GraphLSTM [95], BERT [96], Mem2Seq [97] and GLMP [98]. We follow their metrics and train our model on their preprocessed data for fair comparisons. To further analyze the reasoning ability,

| Model | BLEU | Entity F1 | | | |
|---|---|---|---|---|---|
| | | All | Sche. | Wea. | Nav. |
| S2S | 8.4 | 10.3 | 9.7 | 14.1 | 7.0 |
| S2S+Attn | 9.3 | 19.9 | 23.4 | 25.6 | 10.8 |
| Ptr-Unk | 8.3 | 22.7 | 26.9 | 26.7 | 14.9 |
| GraphLSTM | 10.3 | 50.8 | 69.9 | 46.6 | 43.2 |
| BERT | 9.13 | 49.6 | 57.4 | 47.5 | 46.8 |
| Mem2Seq | 12.6 | 33.4 | 49.3 | 32.8 | 20.0 |
| GLMP | 12.2 | 55.1 | 67.3 | 54.1 | 48.4 |
| GraphDialog | 13.7 | 57.4 | 71.9 | 59.7 | 48.6 |
| COMET-graph | 14.4 | 56.7 | 71.6 | 48.7 | 50.4 |
| T5-DiffKG | 16.04 | 56.2 | 67.2 | 61.5 | 46.7 |

Table 3.2: The results on SMD dataset. S2S, S2S+Attn, Ptr-Unk, GraphLSTM, BERT, Mem2Seq, GLMP, GraphDialog are reported from [84] and COMET-graph from [92]. Our DiffKG achieves the highest BLEU and comparable F1 scores with baselines.

we propose two more baselines based on different ways of leveraging pretrained language models. (1) **NoInfo** model does not take any format of knowledge as the input, aiming to test the performance of a fine-tuned vanilla transformer model on each dataset. (2) **FlatInfo** model constructs the input by concatenating the dialogue history with the NLD form of knowledge, allowing us to investigate the ability of the model to interpret unstructured knowledge.

### 3.4.5 Results

The results on SMD and OpenDialKG are shown in Table 3.2 and Table 3.3. On SMD dataset, we observe that DiffKG outperforms the baselines on BLEU by 17.1% (relative change of 16.04 and 13.7) and achieves comparable entity F1 scores with GLMP and GraphDialog. DiffKG might not improve the entity F1 scores because that prior works group the text inside an entity together (e.g., "road block nearby" becomes a single word "road_block_nearby" in vocabularies). In contrast, we use a universal tokenizer so as to

| Model | path@1 | path@5 | path@10 | BLEU |
|---|---|---|---|---|
| Seq2Seq | 3.1 | 29.7 | 44.1 | - |
| Tri-LSTM | 3.2 | 22.6 | 36.3 | - |
| EXT-ED | 1.9 | 9.0 | 13.3 | - |
| DialKG | 13.2 | 35.3 | 47.9 | - |
| Seq2Path | 14.92 | 31.1 | 38.68 | - |
| AttnFlow | 17.37 | 30.68 | 39.48 | - |
| AttnIO-AS | 23.72 | 43.57 | 52.17 | - |
| T5-NoInfo | - | - | - | 14.51 |
| T5-DiffKG | 26.80 | 54.33 | 61.75 | 15.37 |

Table 3.3: The results on OpenDialKG dataset. The four baselines from Seq2Seq to DialKG Walker are reported from [83] and the other three baselines from Seq2Path to AttnIO-AS are reported from [88]. Our DiffKG achieves the highest path@k scores and is the only one that can simultaneously generate responses.

| **Test KG** | **Method** | **EM** | **F1** |
|---|---|---|---|
| Fixed | GPT2-NoInfo | 10.71 | 43.78 |
| | GPT2-FlatInfo | 14.08 | 47.57 |
| | GPT2-DiffKG | 16.39 | 51.06 |
| | T5-NoInfo | 10.50 | 44.27 |
| | T5-FlatInfo | 28.99 | 66.15 |
| | T5-DiffKG | 27.52 | 63.93 |
| Shuffled | T5-FlatInfo | 17.02 | 54.51 |
| | T5-DiffKG | 27.52 | 64.00 |

Table 3.4: The results on SMD-Reasoning dataset.

prevent heavy preprocessing and specialized vocabularies. This means that DiffKG can perform similarly with state-of-the-art to retrieve knowledge without a tokenizer specified for each dataset. On OpenDialKG dataset, we observe that DiffKG outperforms the baselines in terms of path@k scores and can simultaneously outperform T5 in terms of Entity F1 and BLEU. These demonstrate that DiffKG can retrieve accurate paths for reasoning and effectively incorporate reasoning into response generation.

We also investigate the results of SMD-Reasoning dataset as shown in Table 3.4. We find that DiffKG improves NoInfo by 16.6% and 44.4% F1 scores respectively on

| Method | Domains | | | | | | Reasoning Types | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Schedule | | Navigation | | Weather | | Inform | | Selection | | Extraction | | True/False | |
| | EM | F1 | EM | F1 | EM | F1 | EM | F1 | EM | F1 | EM | F1 | EM | F1 |
| GPT2-NoInfo | 3.49 | 45.7 | 4.63 | 41.6 | 27.5 | 46.8 | 5.03 | 45.2 | 1.45 | 47.4 | 3.06 | 24.0 | 68.6 | 68.6 |
| GPT2-DiffKG | 9.30 | 53.0 | 9.65 | 47.6 | 34.4 | 56.5 | 8.04 | 50.8 | 0.00 | 48.5 | 31.6 | 53.5 | 56.9 | 56.9 |
| T5-NoInfo | 0.00 | 44.6 | 4.63 | 40.9 | 29.0 | 50.7 | 3.02 | 44.9 | 8.70 | 49.1 | 1.02 | 25.2 | 70.6 | 70.6 |
| T5-DiffKG | 20.9 | 63.8 | 19.3 | 61.9 | 48.1 | 68.1 | 18.1 | 61.7 | 11.6 | 62.4 | 50.0 | 73.5 | 70.6 | 70.6 |

Table 3.5: Detailed Evaluation Results of SMD-Reasoning dataset

| | |
|---|---|
| SMD-Reasoning | User: check the date and time of my doctor's appointment<br>(Reasoning Path: Doctor Appointment $\xrightarrow{\text{HasDate, HasTime, ToSelf}}$ Tuesday, 11am, doctor appointment)<br>DiffKG: inform 11 am tuesday doctor appointment |
| | User: Car I need to get to a gas station, please show me the nearest one<br>Assistant: There is Valero 7 miles away with moderate traffic on our way<br>User: Alright, where is it located?<br>(Reasoning Path: Gas Station $\xrightarrow{\text{IsTypeOf}}$ Valero $\xrightarrow{\text{HasAddress, ToSelf}}$ 200 Alester Ave, Valero)<br>DiffKG: inform 200 Alester Ave Valero |
| OpenDialKG | Speaker A: Do you have any info on Toni Kroos?<br>(Reasoning Path: Toni Kroos $\xrightarrow{\sim\text{Player Statistics}}$ Germany national football team)<br>DiffKG: Toni Kroos is German footballer who plays for the Germany national football team. |

Table 3.6: Generated examples and the reasoning path.

GPT2 and T5 models. This demonstrates that DiffKG can utilize knowledge effectively to improve the generation without access to information. In contrast, although FlatInfo gives similar performances as DiffKG on the SMD-Reasoning dataset, it cannot be run on OpenDialKG due to computational costs. More specifically, FlatInfo requires the knowledge graph to be transformed into sentences, which will result in at least a million tokens as the model inputs for OpenDialKG (since the number of triples is a million without designed subgraph sampling), which is not a practical number.

### 3.4.6 Quantitative Analysis

To test the robustness of the methods towards accurately locating information, we shuffle the information order. This evaluation is to simulate the cases that extra information is arbitrarily added when deploying a dialogue system. Specifically, the order of the knowledge context for FlatInfo and the order of knowledge triples are changed during inference time. As shown in the last two rows in Table 3.4, the performance of FlatInfo drops while DiffKG remains about the same. This indicates that the slight superior performance of FlatInfo with the original order can come from the blackbox tricks to group the nearby knowledge in the inputs. When this implicit trick is broken down, the DiffKG shows much better robustness and performance.

To investigate the difficulty of each domain and reasoning type, we divide the results accordingly in Table 3.5. As presented in the domains part, the models achieve the highest EM and F1 on the weather domain. We conjecture the reason is that the weather domain includes more reasoning types (weather:4, navigate:3, schedule:2), thus reflecting more balanced reasoning ability. In the reasoning types part, we observe that true/false is less well coped by DiffKG; however, DiffKG improves the extraction. This shows that DiffKG can effectively check the existence of required knowledge and then query the database.

### 3.4.7 Qualitative Analysis

We visualize the generated examples and the symbolic reasoning path by DiffKG on SMD-Reasoning and OpenDialKG datasets in Table 3.6. The examples show that DiffKG can capture some naturally occurring phenomena in this dataset: (1) the KG reasoning path can be 1 to multiple hops; (2) the reasoning will diffuse to multiple paths (e.g., DiffKG simultaneously applies “HasDate”,“HasTime”,“ToSelf” to “Doctor Appointment”). Along with analyses in previous subsections, we observe that DiffKG can extract interpretable reasoning paths and generate corresponding outputs using reasonable computational costs.

However, even though DiffKG can maintain or improve performance while doing interpretable reasoning on any scaled KG, errors might happen in some cases, such as when it is not clear what information is required in the response and when incomplete entities are retrieved.

# Part II

# Optimizing Conversational Models

# Chapter 4

# Optimizing from the Causal-Effect Perspective

As discussed in the thesis introduction, despite their widespread adoption and the overall performance boost from LLMs, neural conversation models have yet to exhibit natural chat capabilities with humans. In this Chapter, we examine user utterances as *causes* and generated responses as *effects*, recognizing that changes in a cause should produce a different effect. To further explore this concept, we have compiled and expanded upon a new dataset called **CausalDialogue** through crowd-sourcing. This dataset includes multiple cause-effect pairs within a directed acyclic graph (DAG) structure. Our analysis reveals that traditional loss functions struggle to effectively incorporate the DAG structure, leading us to propose a causality-enhanced method called Exponential Maximum Average Treatment Effect (ExMATE) to enhance the impact of causality at the utterance level in training neural conversation models. To evaluate the needs of considering causality in dialogue generation, we built a comprehensive benchmark on CausalDialogue dataset using different models, inference, and training methods. Through experiments, we find that a causality-inspired loss like ExMATE can improve the diversity and agility of conventional loss function and there is still room for improvement to reach human-level quality on this new dataset.

## 4.1 Introduction

Over time, broadly-defined dialogue models have become increasingly prevalent in society and been integrated in a range of domains from speech assistants and customer service systems to entertainment products, such as video games, where the non-playable characters (NPCs) engage in conversation with players. A core goal of training chatbots is enabling them to interact with humans naturally [6, 99]. This includes, but is not limited to: considering both the machine and addressee's personalities [21], diversifying responses to be less generic (e.g., the same response "I don't know." is often produced in a traditional setting for different dialogues) [40], grounding on external knowledge to be informative [26], and tailoring responses specific to nuanced differences in conversation.

To the best of our knowledge, no recent studies have prioritized the ability to tailor responses for minor differences in conversations. This problem is currently implicitly approached by training models with larger scale or cleaner conversation data [15, 100, 3] or involving human-in-the-loop [7, 101]. However, the effectiveness of these methods is unclear, the online rewarding scheme can be expensive, and a suitable testbed for evaluating the solution to this problem has not yet been identified.

To this end, we propose a benchmark to foster research in tailoring responses for nuanced differences in conversations by answering the question "*if all prior turns are the same, but the last turns in two conversations are semantically different, how should future turns differ?*" We call this concept *Agility* and model it as the *utterance-level causes and effects* in dialogue response generation, where the causes are the slightly different prior turns and the effects are the resulting future turns.

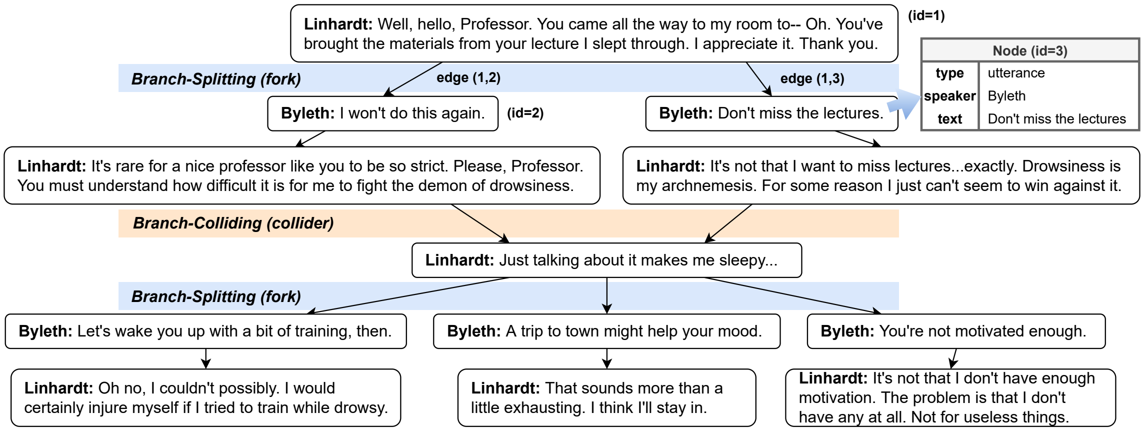

Linhardt: Well, hello, Professor. You came all the way to my room to-- Oh. You've brought the materials from your lecture I slept through. I appreciate it. Thank you.
(id=1)
Node (id=3)
type utterance
speaker Byleth
text Don't miss the lectures
Branch-Splitting (fork)
edge (1,2)
edge (1,3)
Byleth: I won't do this again.
(id=2)
Byleth: Don't miss the lectures.
Linhardt: It's rare for a nice professor like you to be so strict. Please, Professor. You must understand how difficult it is for me to fight the demon of drowsiness.
Linhardt: It's not that I want to miss lectures...exactly. Drowsiness is my archnemesis. For some reason I just can't seem to win against it.
Branch-Colliding (collider)
Linhardt: Just talking about it makes me sleepy...
Branch-Splitting (fork)
Byleth: Let's wake you up with a bit of training, then.
Byleth: A trip to town might help your mood.
Byleth: You're not motivated enough.
Linhardt: Oh no, I couldn't possibly. I would certainly injure myself if I tried to train while drowsy.
Linhardt: That sounds more than a little exhausting. I think I'll stay in.
Linhardt: It's not that I don't have enough motivation. The problem is that I don't have any at all. Not for useless things.

Figure 4.1: A dialogue DAG example in the new dataset CausalDialogue. As the conversation progress, each utterance can be continued with multiple responses (branch-splitting; fork); meanwhile, the same root dialogue with different middle turns can be continued by the same response (branch-colliding; collider).

We introduce **CausalDialogue**, a dataset seeded by expert-written dialogues containing branching dialogue paths, which we further expand in terms of scale and linguistic abundance with crowd-sourcing. Each conversation is represented as a directed acyclic graph (DAG) for ease of storage and causal analysis [102] as shown in Figure 4.1. As conversations progress, each utterance can elicit multiple responses, resulting in a split of the conversation (branch-splitting). Alternatively, multiple conversations that share a common starting point may sometimes lead to the same response, even if the middle exchanges differ (branch-colliding). Due to the DAG structure of CausalDialogue, it is ideal for aiding research on response generation that requires abundant *IF*-bases, for instance, causal inference and offline reinforcement learning, which may improve the response generation quality for nuanced differences in conversation.

To provide a benchmark for future work on the CausalDialogue dataset, we conduct experiments with various setups. We include both decoder-only and encoder-decoder transformer models pretrained on either common or dialogue-specific corpora, various inference methods, conventional training losses, and a newly proposed loss, Exponen-

| | **CausalDialogue** | **TV Series** | **MultiTalk** | **DailyDialog** | **PersonaChat** | **LIGHT** |
|---|---|---|---|---|---|---|
| Branches | ✓(DAG) | ✗ | ✓(Tree) | ✗ | ✗ | ✗ |
| Profiles | ✓ | ✓ | ✗ | ✗ | ✓ | ✓ |
| Situated | ✓ | ✓ | ✓ | ✗ | ✗ | ✓ |
| Expert involved | ✓ | ✓ | ✗ | ✓ | ✗ | ✗ |

Table 4.1: Compared to current widely used datasets, CausalDialogue contains the utterance-level graph structure and meanwhile has the features of diverse speaker profiles, descriptive situations, and high quality scripts written by experts. The referring dialogue generation datasets are: TV series [27], MultiTalk [103], DailyDialog [104], PersonaChat [23], LIGHT [105].

tial Maximum Average Treatment Effect (ExMATE), inspired by Average Treatment Effect [106, 107], which is a method commonly used to approximate the causal effect of a treatment and its outcome. In this benchmark, we show that existing methods are not sufficient in tackling the agility issue, and a simple causality-inspired loss demonstrates improvement.

## 4.2 Background: Chit-Chat Dialogue Datasets

To boost the research of dialogue models, the community has collected conversations from multiple sources. There are data with dialogues within scripts written by experts for movies [108, 109, 110], TV shows [111, 27, 112, 113], and education purposes [104, 114]. There is also work that collectively ensemble these dialogue datasets for multiple purposes [115]. Moreover, for abundant diversity and real-life scenarios, [116, 117, 118, 119] collected datasets based on the publicly available data from social media and forums. Additionally, previous work has explored the idea of collecting data through crowd-sourcing with added constraints to improve its quality or expand label types, including more task-oriented [25, 19, 120] and the ones lean toward open-domain. For example, [23] constructed a dataset with workers imitating a given personal profile. [121] built a dataset by explicitly asking workers to show their empathy during a conversation. [105, 122, 123] created datasets with the assistance of game structures, so the purpose of the dialogue is to complete a mission or collaborate with other agents. Finally, recent work by [103] collected branches of dialogues for 120 self-written prompts to create dialogue trees. Compared to previous studies, our dataset is a fusion of the scripts written by experts and responses created by crowd-sourcers with manual correction, granting it high quality, linguistic abundance, and extensive metadata. Additionally, our dataset includes both branch-splitting and branch-colliding instances, which has led us to classify dialogues as directed acyclic graphs (DAGs) instead of just sequences or trees.

## 4.3 CausalDialogue Benchmark

In this section, we introduce **CausalDialogue**, a novel dataset that includes chit-chat conversations in a Conversational Directed Acyclic Graph (DAG) data structure. This structure allows for the natural inclusion of various dialogue flows, such as forks (branch-splitting) and colliders (branch-colliding) [102]. Our goal is to offer researchers a valuable resource for studying the complexities of human conversation and advancing the understanding of causal inference in dialogue.

To create CausalDialogue, we sourced expert-written dialogues from a role-playing game (Section 4.3.1) and expanded upon them with Amazon Mechanical Turk (MTurk)[1] and manual correction (Section 4.3.2). By using our fused collection method, the dataset contains high-quality, engaging conversations with abundant linguistic usage that imitates daily life.

[1] `https://www.mturk.com`

| **Data Partition** | **Ori.-2S** | **Multi** | **Expan.** | **Total** |
|---|---|---|---|---|
| # Dialogues† | 794 | 1528 | 623 | 2322 |
| # Branches | 1633 | 1298 | 2378 | 4866 |
| # Utterances | 33247 | 13858 | 15728 | 46109 |
| # Speakers | 41 | 47 | 39 | 51 |
| Avg. utts/dial. | 17.0 | 51.4 | 5.6 | 26.8 |
| Avg. words/utt. | 18.4 | 17.8 | 11.8 | 16.5 |
| Avg. utts/spk. | 801.6 | 268.8 | 402.8 | 878.4 |

Table 4.2: The statistics of CausalDialogue dataset, where the columns Ori.-2S and Multi are the crawled and cleaned original scripts and the column Expansion is from crowd-sourcing. In total, there are 3457/741/715 dialogues for train/validation/test sets. † indicates that for Expansion set, 623 is the number of initial dialogues that are parts of the 794 Ori.-2S dialogues, so the total number of dialogues is 2322.

### 4.3.1 Data Collection

CausalDialogue is derived from the English scripts of the popular role-playing game (RPG) *Fire Emblem: Three Houses*, which we sourced from the fandom wikipedia[2] under the GNU Free Documentation License(GFDL)[3]. This RPG is well-known for its diverse, story-driven conversations, which mix the interactions of approximately 40 main characters. In this game, players have the ability to shape the narrative by making choices that lead to different dialogue branches.

Table 4.2 lists the statistics of the two main types of the crawled data, which are already divided in the raw scripts. We name the first conversation type Ori.-2S, which are mostly dialogues between two speakers, and generally include conversations about interpersonal relationships. We name the second conversation type Multi, which are dialogues between two or more speakers, and usually describe the current status of the story line. In the following sections, we will introduce the DAG structure to better describe the dataset, as well as how we obtained additional examples from crowd-sourcing to create the Expansion to these expert-written scripts.

[2] https://fireemblem.fandom.com/

[3] https://fireemblem.fandom.com/wiki/Fire_Emblem_Wiki:Copyrights

**Dialogue DAGs.** Conventional linear dialog data structures can be challenging to create when dealing with *forks* and *colliders*, as they can lead to ambiguity in the form of duplicated utterances and split responses. To address this issue, we propose using a conversational DAG to maintain the fidelity of the dialog. We convert each textual conversation into a DAG, as demonstrated in Figure 4.1. Formally, each node is a dictionary containing the text type (utterance/scene information), text, speaker, and its own id in the dialogue. A directed edge $(i, j)$ then indicates that a node with id $j$ is a possible response to the node with id $i$. Saving dialogues as DAGs may introduce some complexity, but it also offers numerous benefits. For example, it reduces the memory required to save each dialogue branch independently, enables a natural visualization of the multiple possible dialogues flows, and fosters the survey of causality on dialogue utterances.

**Speaker Profiles.** Prior work has shown the relationship between personality and language uses in conversations [124]. To ensure consistent personality, as well as to diversify linguistic features across speakers, we leverage the speaker profiles during the data collection process. The resulting CausalDialogue dataset comprises 41 main speakers who have been thoughtfully crafted by the game's developers. These speakers possess diverse backgrounds, perspectives, and interests, and their characteristics are both human-like and distinct. These speaker profiles are simplified for collecting the Expansion partition to reduce workers' cognitive load. Compared with the speaker profiles in CausalDialogue, previous works have provided limited information (e.g. "I have a dog.") [23, 105], or have a significantly smaller number of speakers [111, 27]

### 4.3.2 Data Expansion

In order to increase the breadth and scope of our dataset, we propose utilizing a crowdsourcing approach to add more diverse and current language as shown in Figure 4.2.

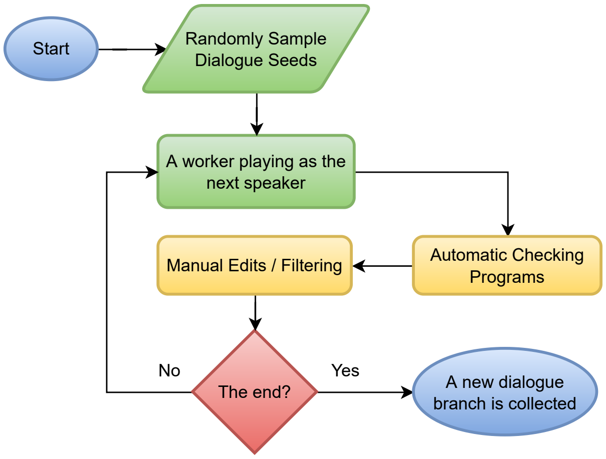
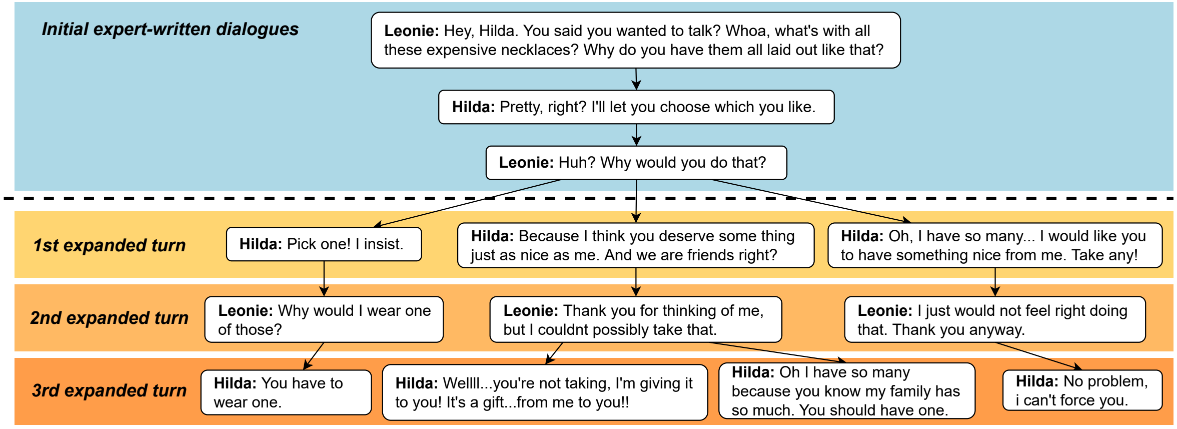

Start
Randomly Sample Dialogue Seeds
A worker playing as the next speaker
Automatic Checking Programs
Manual Edits / Filtering
The end?
No
Yes
A new dialogue branch is collected

Figure 4.2: The flowchart of our strategy for data expansion with crowd-sourcing.



**Leonie:** Hey, Hilda. You said you wanted to talk? Whoa, what's with all these expensive necklaces? Why do you have them all laid out like that?

**Hilda:** Pretty, right? I'll let you choose which you like.

**Leonie:** Huh? Why would you do that?

**Hilda:** Pick one! I insist.

**Hilda:** Because I think you deserve some thing just as nice as me. And we are friends right?

**Hilda:** Oh, I have so many... I would like you to have something nice from me. Take any!

**Leonie:** Why would I wear one of those?

**Leonie:** Thank you for thinking of me, but I couldnt possibly take that.

**Leonie:** I just would not feel right doing that. Thank you anyway.

**Hilda:** You have to wear one.

**Hilda:** Wellll...you're not taking, I'm giving it to you! It's a gift...from me to you!!

**Hilda:** Oh I have so many because you know my family has so much. You should have one.

**Hilda:** No problem, i can't force you.

Figure 4.3: A dialogue example of the Expansion partition in CausalDialogue.

**Initial Dialogue Selection.** We first randomly select 1,200 partial dialogues from the Ori.-2S partition, which is of higher quality after our manual inspection. This can result in more stable quality when crowd-sourcing responses.

**Expansion Collection.** Each initial dialogue along with the continuing speaker profile is presented to 3 workers on MTurk to write the next utterance. A new branch of continued dialogue will then be presented to another 1-2 workers playing as another speaker to gather another round of responses. We repeated this process three times and collect a total of about 13,000 written utterances. Table 4.2 lists the detailed statistics of the expanded data in the column Expansion. Note that the statistics of Expansion in Table 4.2 include the initial dialogues. Figure 4.3 shows an DAG representation of an expanded example.

**Quality Control.** We adopt three strategies to control for dialogue quality. First, we asked the workers on MTurk to annotate if they regard a dialogue as already completed or having too specific of details to continue. The purpose of the first stage of quality control is to identify conversations which cannot be continued, either because the conversation has already concluded or because the workers are lacking enough information about the world to continue the conversation. Second, we used an off-the-shelf model[4] to label potential ethical issues inside the collected utterances for reference in the next step. Finally, we invited real players of the game and machine learning researchers to manually check all the utterances by their fluency, coherence, and ethics as well as referring to the labels from the previous two steps to ensure the final Expansion partition is of high quality.

### 4.3.3 Task Definition

In this work we consider a conversation among two or more speakers. At each time step $t$, a speaker $s_t$ takes their turn with an utterance $u_t$. The goal, as in conventional response generation, is to train a model parameterized by $\theta$ that can predict a plausible next utterance given the speakers and utterances in prior turns as:

$$u_{t+1} \sim P_\theta(\cdot | s_1 u_1, s_2 u_2, ..., s_t u_t, s_{t+1}) \,. \tag{4.1}$$

[4] `https://github.com/unitaryai/detoxify`

Distinct from prior conversation datasets, CausalDialogue has many dialogue branches. If we consider each branch as an independent conversation (flatten the branches), many conversations will have large overlaps and thus bias the dataset. We consider this point and extract triples $(DH, x, y)$ from CausalDialogue. To simplify notations for following sections, we denote $s_t u_t$ as $x$, $s_{t+1}u_{t+1}$ as $y$ and $DH$ is the dialogue history $s_1u_1, s_2u_2, ..., s_{t-1}u_{t-1}$. The key idea is that for a $DH$, we will not extract duplicated pairs $(x, y)$, but $x$ or $y$ itself can be shared.

The CausalDialogue response generation task is therefore defined as finding a possible turn-taking speaker and their response given the dialogue history $DH$ with an utterance cause $x$.

$$y \sim P_\theta(\cdot|DH, x)\,. \tag{4.2}$$

The sequences $x = x_1x_2...x_i...x_{|x|}$ and $y = y_1y_2...y_j...y_{|y|}$, where $x_i$ and $y_j$ are tokens, and $|x|$ and $|y|$ are the length of the sequences $x$ and $y$ respectively.

### 4.3.4 Agility

While the above task definition resembles the standard dialogue generation setting with the exception of speaker prediction and conversation overlaps, our primary interest lies in tailoring responses to minor differences in conversation history. We refer to this concept as *Agility*, where a minor difference in conversations can be a shared $DH$ with different continuation $x$.

To quantify the idea of agility, we propose a new metric with the following idea: If the predicted next utterance $y$ and the previous turn $x$ has causal-effect relationship (i.e., $x_1 \rightarrow y_1$ and $x_2 \rightarrow y_2$), we anticipate that it is less likely that $y_2$ is caused by $x_1$. The newly proposed metric, named confidence causal-effect (CCE) is formally defined as:

$$\begin{aligned} CCE =& E_{(x,y)\in D,(x,y')\notin D,(x',y')\in D} \\ & \left[PPL_\theta(y'|DH,x) - PPL_\theta(y|DH,x)\right], \end{aligned} \tag{4.3}$$

where PPL refers to perplexity. Note that CCE is not a metric that stands by itself and needs to refer to PPL at the same time. That is, given a similar PPL score, a model with higher CCE score is better. Additionally, it is important to acknowledge that the concept of agility has been indirectly incorporated into conventional dialogue generation models and evaluation metrics, but it has not been specifically examined in isolation. Our newly introduced dataset and CCE metric can be seen as an initial step towards addressing this aspect.

## 4.4 Methods: MLE and ExMATE

In this section, we describe how conventional generative models can be used and propose a simple yet effective approach to model causal effect.

### 4.4.1 Maximize Likelihood Estimation

An often used method to train a conditional sequence generation model is minimizing the negative log likelihood [6, 81]. The loss function is as following:

$$\begin{aligned} & L_{MLE} = \\ & \underset{(DH,x,y)\sim P_D}{E} \sum_{j=1}^{|y|} -\log P_\theta(y_j|DH,x,y_{1\ldots j-1}), \end{aligned} \tag{4.4}$$

where $P_D$ represents the data distribution. Since the duplication of dialogue history is already taken in to account in our task definition (Section 4.3.3), this MLE method can be seen as the recently proposed dialogue tree model [103]. However, this function only models a part of the cause-effect relationship between the condition and the output sequence. This neglect may lead to a more vague predicted probability distribution of the output, thus generating less agile responses.

### 4.4.2 Maximize Average Treatment Effect

To explicitly model the causal effect in a conversation, we propose the Exponential Maximum Average Treatment Effect (ExMATE), taking into account the treatment effect in causal inference [102]. The treatment effect, denoted by $\delta$, is defined as the difference between the outcome under treatment $I = 1$, represented by $\mathcal{O}^{I=1}$, and the outcome under treatment $I = 0$, represented by $\mathcal{O}^{I=0}$. This measures the variation in outcomes when an event $I$ is present or absent. A higher value of $\delta$ indicates that the event $I$ is more likely to be a true cause of the outcome. Conversely, a small value of $\delta$ suggests that the event $I$ is unlikely to be a cause of the outcome and may only be correlated. We aim to utilize this characteristic in dialogue generation modeling to ensure that a preceding utterance can be considered the genuine cause of the predicted response.

We consider the *fork-like* DAGs (as shown in Figure 4.4) existing in a dataset such as Figure 4.1 and Figure 4.3. Without loss of generality, in a binary case, this type of DAG involves two triples that share the same $DH$ and can be simplified as having nodes $DH$, $X_1$, $X_2$, $Y_1$, and $Y_2$. Here we use ($X_1$, $Y_1$) and ($X_2$, $Y_2$) to denote two possibilities of ($x$,$y$) after $DH$. We take $I = 1$ as choosing the branch $X_1$, and $I = 0$ as choosing an alternative branch $X_2$. Therefore, a traditional definition of the treatment effect $\delta_i = |\mathcal{O}_i^{I=1} - \mathcal{O}_i^{I=0}|$ for the $i$-th example in this type of DAG can be rewritten as:

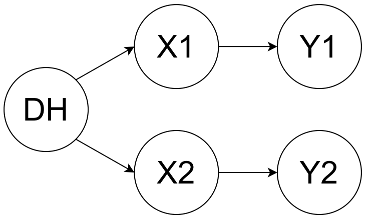


Figure 4.4: The graphical model of fork-like DAG considered in our proposed ExMATE loss.

$$\delta_i \triangleq \underset{\substack{X_1 \sim P_D(\cdot|DH_i),\\ X_2 \sim P_D(\cdot|DH_i),\\ X_1 \neq X_2}}{E} |\mathcal{O}_i^{X_1} - \mathcal{O}_i^{X_2}|\,, \tag{4.5}$$

where $\mathcal{O}_i^{X_1}$ or $\mathcal{O}_i^{X_2}$ is the outcome of an oracle given $X_1$ or $X_2$ as the input.

Since the outcome of a dialogue model is hard to be mathematically described only by an input $X$, we instead utilize the uncertainty of predicting the pair $(x, y)$ by a model $\theta$. We abuse the notation $\mathcal{O}_i$ here and redefine it as,

$$\mathcal{O}_{i,Y_1}^{X_1} \triangleq P_\theta(Y_1|DH, X_1)\,. \tag{4.6}$$

After formulating a dialogue generation problem as utterance-level causal analysis as above, we apply the Average Treatment Effects (ATE) [106] to conversational DAGs, which is defined as

$$\begin{aligned} ATE &\triangleq E_i[\delta_i] = E_i[\delta_{i,Y_1} + \delta_{i,Y_2}] \\ &= E_i[\mathcal{O}_{i,Y_1}^{X_1} - \mathcal{O}_{i,Y_1}^{X_2} + \mathcal{O}_{i,Y_2}^{X_2} - \mathcal{O}_{i,Y_2}^{X_1}]\,. \end{aligned} \tag{4.7}$$

Recall that our goal is to strengthen the cause-effect relationship of each pair, $(X_1,Y_1)$ and $(X_2,Y_2)$ in the binary case. This can be taken as maximizing the defined ATE in Equation 4.7 with respect to the model parameters $\theta$.

Therefore, we substitute the $\mathcal{O}_{i,Y}^{X}$ term in Equation 4.7 with its definition stated in

Equation 4.6 and derive:

$$\begin{aligned}
&\arg\max_{\theta} ATE = \\
&\arg\max_{\theta} \underset{(X_i,Y_i)\sim P_D(\cdot|DH)}{E} P_\theta(Y_i|DH,X_i) \\
&\quad - \underset{\substack{X_i\sim P_D(\cdot|DH),Y_j\sim P_D(\cdot|DH)\\(DH,X_i,Y_j)\notin D}}{E} P_\theta(Y_j|DH,X_i)\,.
\end{aligned} \tag{4.8}$$

To stabilize the training, we modify it with logarithmic and exponential terms and call it the ExMATE loss function. Formally, it is written as:

$$\begin{aligned}
&L_{ExMATE} = \\
&\underset{\substack{(DH,x,y)\sim P_D,\\ x_c\sim P_D(\cdot|DH),\\ (DH,x_c,y)\notin D}}{E} - \log P_\theta(y|DH,x) + \exp(\log P_\theta(y|DH,x_c))\,.
\end{aligned} \tag{4.9}$$

The intuition for this change is that without $\exp(\cdot)$, the gradient of the second term will dominate the loss function, since $\log(u)$ has much larger gradient for $u$ close to 0 than $u$ close to 1 and an $\exp(\cdot)$ term can linearize it.

Overall, the idea of ExMATE is to maximize the response generation model's causal effects given a specific $X_i$ (or $(DH, x)$) as the current cause. At the end, we found that this ATE-inspired approach turns out to be a combination of MLE and a subtraction of specific negative samples. This formulation shares a similar concept with negative sampling and contrastive learning [125, 126], but has different example selection scheme and is not applied on the embedding space. With this method, we are interested in the research question: *Will a model trained on the CausalDialogue dataset be affected when using a causality-inspired loss?*

| Model | Loss | Inference | | Fluency | | | Diversity | | Agility | Identity |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | PPL (↓) | BLEU1 (↑) | 2 (↑) | 4 (↑) | Dist1 | Dist2 | CCE (↑) | Acc (↑) |
| Human Written Responses | | | 1.2 | 48.9 | 34.0 | 25.9 | 1.70 | 11.1 | Inf | 100.0 |
| DG | MLE | Greedy Search | 18.9 | 11.2 | 4.47 | 0.84 | 0.73 | 3.42 | 2.33 | 32.51 |
| DG | MLE | Softmax (T=0.5) | 18.9 | 17.0 | 6.43 | 1.17 | 1.12 | 9.09 | 2.33 | 30.97 |
| DG | MLE | TopK (K=10) | 18.9 | 15.7 | 5.34 | 0.81 | 1.37 | 13.57 | 2.33 | 27.65 |
| DG | ExMATE | Greedy Search | 19.0 | 10.7 | 4.26 | 1.05 | 0.79 | 3.65 | 2.68 | 32.18 |
| DG | ExMATE | Softmax (T=0.5) | 19.0 | 15.5 | 5.70 | 1.06 | 1.25 | 9.71 | 2.68 | 31.18 |
| DG | ExMATE | TopK (K=10) | 19.0 | 13.5 | 4.47 | 0.67 | 1.52 | 14.44 | 2.68 | 28.16 |
| T5 | MLE | Greedy Search | 15.4 | 5.80 | 2.52 | 0.58 | 1.11 | 4.37 | 1.39 | 75.64 |
| T5 | MLE | Softmax (T=0.5) | 15.4 | 12.7 | 5.06 | 0.97 | 1.77 | 10.91 | 1.39 | 74.66 |
| T5 | MLE | TopK (K=10) | 15.4 | 14.1 | 5.09 | 0.82 | 2.07 | 15.49 | 1.39 | 72.79 |
| T5 | ExMATE | Greedy Search | 15.4 | 5.66 | 2.46 | 0.55 | 1.10 | 4.06 | 1.50 | 75.76 |
| T5 | ExMATE | Softmax (T=0.5) | 15.4 | 12.6 | 5.02 | 1.00 | 1.72 | 10.73 | 1.50 | 74.80 |
| T5 | ExMATE | TopK (K=10) | 15.4 | 14.1 | 5.06 | 0.80 | 2.06 | 15.67 | 1.50 | 72.83 |

Table 4.3: The test results on CausalDialogue of different fine-tuned backbone models (DialoGPT (DG) and T5), inference methods (Greedy Search, Softmax, TopK), and loss functions (MLE and ExMATE). Using ExMATE loss enhances the agility aspect of dialogue generation models without compromising their fluency ratings.

## 4.5 Experiments: Response Fluency vs. Agility

We provide a preliminary benchmark for CausalDialogue with often used methods and a naive causality-inspired loss. We fine-tuned two types of pretrained language models based on transformers [11]: decoder-only architecture, DialoGPT [15] and encoder-decoder architecture, T5 [14], by the conventional MLE loss and the proposed ExMATE loss, and inferred by various sampling methods. We evaluate three aspects of the generated responses: Fluency (perplexity (PPL) and BLEU [89]), Diversity (Distinct n-grams Dist1 and Dist2 [40]), and our proposed Agility (CCE) in Section 4.3.4. Furthermore, we use accuracy to evaluate if the speaker for a given turn is correctly predicted as the one in the human written responses (Identity Acc).

### 4.5.1 Results

The test results of human written responses and models trained and inferred by different setups are listed in Table 4.3.

[**Backbone Models**] We observe that our trained T5 model is generally better than DialoGPT model, as evidenced by the significant difference in PPL and Identity Acc between them. [**Inference Methods**] We observe that Softmax and TopK can achieve better results than greedy search in this dataset, as evidenced by their BLEU and Distinct-N scores. The reason is similar to the conventional generic response problem in open-domain dialogue generation [40, 41], since in a DAG, a $(DH, x)$ pair have multiple $y$ as references, causing even an ideal probability distribution to have high entropy. [**Loss Functions**] We find that ExMATE improves MLE with better diversity, agility, and identity accuracy, while maintaining similar fluency scores. This meets our expectation that ExMATE should not deteriorate the MLE's ability in training a model while maximizing the potential causal effect in response prediction. This result empirically shows that the causal effect can help to increase diversity and predict the turn-taking speaker as well. Finally, compared to the evaluation results of human written responses (a hard-to-reach upper bound), current methods still need improvement, except for diversity scores.

### 4.5.2 Human Evaluation

We randomly sample 100 dialogues, present each example to three workers on MTurk and ask them score the three dimensions, agility, coherence, and informativeness, scaling from 1 to 5. For each example, we present one shared dialogue history with two branches and the corresponding machine generated responses or a human written response. We randomly mix the human written ones to validate if the human evaluation is reliable to an extent, by anticipating human written ones will get higher scores. We list the average ratings in Table 4.4. The model trained with ExMATE achieves a similar informativeness level as human written ones, and gets a higher agility rating, which is its main goal. However, ExMATE can compromises coherence due to the subtraction of a counter example, which is a natural sentence, in its objective function. The human evaluation demonstrates the challenge of models to meet human-level quality in Causal-

| **Model** | Coherence | Informativeness | Agility |
|---|---|---|---|
| Human | 3.78 | 3.72 | 3.49 |
| MLE | 3.63 | 3.60 | 3.36 |
| ExMATE | 3.59 | 3.74 | 3.40 |

Table 4.4: The human evaluation results (scale 1-5, the higher the better) of models trained on CausalDialogue (MLE, ExMATE), and human written responses (Human) for reference.

| | | | |
|---|---|---|---|
| **Dialogue History** (*DH*) | | Lysithea: Oh, hey. It's you. Going for a walk again today?<br>Ignatz: No, I'm on cooking duty today, and I have to head into town for some groceries. | |
| **Cont. Conv** ($x$) | | **case1** Lysithea: That sounds like quite a task! | **case2** Lysithea: Would you like some company? |
| response($y$) | **MLE+S** | Ignatz: You're right. | Ignatz: Yes, I'd love to. |
| | **MLE+T** | Ignatz: I'll go for a run. | Ignatz: No, I'm not. |
| | **ExMATE+S** | Ignatz: I'm not sure I can handle it. | Ignatz: Yes, I'm a bit tired of the whole thing. |
| | **ExMATE+T** | Ignatz: Well, I have to find a new name for myself. | Ignatz: No! It's just me and I. |
| **Cont. Conv** ($x$) | | **case3** Lysithea: Allow me to join you so I can help. | **case4** Lysithea: All on your own? Hm, I'd better go with you. I'd be worried if you went by yourself. |
| response($y$) | **MLE+S** | Ignatz: Oh, I'm on cooking duty. | Ignatz: I'm just a bit worried about you. |
| | **MLE+T** | Ignatz: I'm going to help cook. | Ignatz: Yeah, no, not really! I can't help you, but I'm just going to help you out. |
| | **ExMATE+S** | Ignatz: I'm not working, but I'll help you. | Ignatz: I've got no choice. I'll just have to go out alone. |
| | **ExMATE+T** | Ignatz: Oh. I'm sorry, I couldn't be there for you. | Ignatz: Is it okay? |

Table 4.5: Generated responses by our trained models, T5 models trained by MLE or ExMATE inferred by Softmax(S) or TopK(T) sampling methods, given a shared dialogue history but different branches. Using ExMATE loss generally produces more diverse and agile responses.

Dialogue featured by conversational DAGs, a portion of the diversed types of flows in the real world.

### 4.5.3 Qualitative Analyses and Discussion

Table 4.5 shows an example of a shared dialogue history, four different continuations (case1-4), and responses generated by the same backbone model, T5, trained with different objectives and inferred with different sampling methods. We observe that responses produced by MLE+T (TopK), ExMATE+S (Softmax), ExMATE+T are generally coherent to the conversation, while ExMATE often produces more diverse and agile responses to different continuation cases (different $x$). It is notable that other than the improvements, we find that all the models have three types of issues: mode collapse, semantic repetition, and identity misplacement. [**Mode Collapse**] The problem is often-seen when inferring a model by greedy search, specifically, the predicted responses often repeat the same phrase such as "I'm not sure". While tacking the issue by adopting inference sampling, we conjecture the reason is that in a DAG, using a typical loss function can learn a probability distribution with higher entropy. This also demonstrates the need of a new loss function for training on a conversational DAG dataset. [**Semantic Repetition**] An example is the MLE+T response in Table 4.5 case 4, where "can't help you " and "help you out" have semantic overlaps. This issue can possibly be mitigated by repetition reduction techniques, such as unlikelihood training [127] in future work. [**Identity Misplacement**] The problem happens when a model is confused about its position in a dialogue. For instance, the MLE+T response in Table 4.5 case 3 is more like an utterance of speaker Lysithea instead of Ignatz. This issue might be soothed by existing persona consistent techniques [21, 128, 24] for building a overall good chatbot, while in this work, we focus on proposing a new dataset to benchmarking on the agility issue.

# Chapter 5

# Rewarding Good and Penalizing Bad Examples

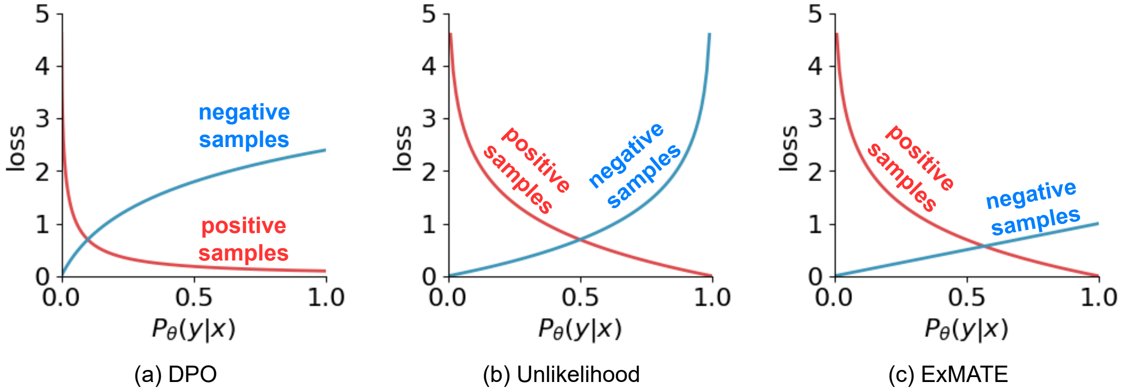
In the previous Chapter, we have discussed that beyond maximum likelihood estimation (MLE), the proposed exponential maximizing average treatment effect (ExMATE) can further improve conversational model optimization. Actually, while MLE is the standard objective of a language model (LM) that optimizes good examples probabilities, there are other studies exploring ways that also penalize bad examples to enhance the quality of output distribution, including ExMATE, unlikelihood training, and direct preference optimization (DPO). However, to the best of our observation, no study has systematically compared these methods or provided a unified recipe for LM optimization. In this Chapter, we present an unique angle of gradient analysis of loss functions that *simultaneously reward good examples and penalize bad ones* in LMs. Through both mathematical results and experiments on CausalDialogue and Anthropic HH-RLHF datasets, we identify distinct functional characteristics among these methods. We find that ExMATE serves as a superior surrogate for MLE, and that combining DPO with ExMATE instead of MLE further enhances both the statistical (5-7%) and generative (+18% win rate) performance.

## 5.1 Introduction

The optimization of language models (LM) has long relied on maximum likelihood estimation (MLE) [129, 130, 5]. While MLE aims to concentrate probability distributions on *correct tokens* at each timestep, this approach has inherent limitations. Solely optimizing for correct examples can lead to over-optimism on the referred token [101] and unintended distribution (such as uniform) over unused tokens, regardless of the data scale. Consequently, a paradigm shift has occurred, recognizing the need to consider both positive and negative examples in LM optimization.

To address the shortcomings of exclusively rewarding correct data, novel strategies have emerged, originating from binary classifiers [131] and extending to sequential multi-class classifiers like LMs. Techniques such as unlikelihood training [132] and exponential maximizing average treatment effect (ExMATE) [46] introduce distinct loss functions and negative sample constructions to mitigate issues like repetition in text generation and enhance model response agility. Meanwhile, generative adversarial networks (GANs) for LMs [133, 41] and reinforcement learning from human feedback (RLHF) [134, 135] either directly takes machine generation as negative data or further annotates preference by humans to optimize the model via GAN or RL frameworks [136, 37, 137, 138]. Recently, direct preference optimization (DPO) [139] streamlines the RLHF approach into a supervision loss objective, significantly reducing computational costs while maintaining efficacy. These approaches collectively signify a broader shift towards optimizing LMs by simultaneously increasing the probability of preferred data and decreasing the probability of disliked data.

In this chapter, we aim to systematically compare LM optimization methods that



(a) DPO (b) Unlikelihood (c) ExMATE

Figure 5.1: (a) DPO, (b) Unlikelihood, and (c) ExMATE loss functions when taking only either $P_\theta(y^+|x^+)$ (positive examples) or $P_\theta(y^-|x^-)$ (negative examples) as the control variables. We plot DPO in the case of $P_{ref}(\cdot) = 1$, $\beta = 1$, and $P_\theta(y^-|x^-)$ or $P_\theta(y^+|x^+)$ is 0.1 for easy visualization. Their function characteristics are different, thus making them suitable for difference use cases.

share the principle: *rewarding good and penalizing bad examples.* Specifically, we address the following questions: (1) What are the essential differences among these methods in LM optimization? (2) Which method is more suitable for each scenario? (3) Can we identify a superior optimization recipe based on mathematical analysis? To answer these questions, we propose a gradient analysis approach tailored for frequently encountered LM scenarios, enabling us to mathematically estimate how each rewarding-good-penalizing-bad (RGPB) method updates the LM output distribution and elucidate their distinct properties. Additionally, we conduct experiments on datasets such as CausalDialogue [46] and Anthropic HH-RLHF [16], employing evaluations using statistical metrics and GPT4 assessments to verify our mathematical findings and validate the practical implications of our research.

## 5.2 Related Model Optimization Approaches

As rewarding good examples and penalizing bad ones has been one widely-used framework in LM research, we discuss two major differences in these works: (1) the method to construct negative data and (2) the method to optimize the LM using the negative data. Moreover, we discuss (3) the difference of them from other lines of research, e.g., contrastive learning, that is often deemed similar.

**Negative Data Construction.** Word2Vec [140, 125], which aims to strengthen word embeddings, performs negative sampling by intentionally selecting incorrect positions for a word. Unlikelihood training [132, 141], which aims to prevent repetitions in text generation, uses already predicted words in the context as negative samples. ExMATE [46, 48], which aims to enhance LM response sensitivity to prior utterances or controls, constructs negative samples by replacing the context with a slightly incorrect predecessor. GANs for LMs [142, 133, 41] use the generator's outputs as negative samples. Evaluation models have also been trained via experimeted recipes to synthesize negative data [143, 144]. RLHF [134, 101, 135, 16] and its derivatives, such as DPO [139], IPO [145], and KTO [146], collect human feedback to label pairwise preferred and rejected responses generated by a fine-tuned model, with the rejected responses serving as negative data. In this work, we discuss often-seen cases in generative LMs, e.g., when the negative data that are also fluent language. This requires the method to identify the nuanced difference between the positive and negative data.

**Optimization Method.** Word2Vec, GANs for LMs, and Unlikelihood training use a similar loss function long employed for optimizing binary classifiers [140, 136, 132]. ExMATE [46], inspired by the average treatment effect and the directed acyclic graph structure of conversations [106, 102], proposes an exponential trick to linearize gradients to maintain language fluency. DPO [139] and its derivatives [145, 146, 147], also supervision loss functions, are firstly derived from RLHF [139] to reduce the resource-intensive interactions in RL frameworks [37, 148, 149, 138], relying on an assumed reward model [135, 16] and Kullback–Leibler divergence regularization. In this chapter, we mainly discuss Unlikelihood, ExMATE, and DPO as they represent three distinct lines of research towards the same goal. We discuss their function characteristics mathematically and empirically under the same setups.

**Different from other Contrasts in ML.** Contrastive learning [150, 151, 152, 153, 126] aims to learn similar representations for similar data points and vice versa. The methods we discuss here, instead of learning representation space based on similarity among data points, aim to directly reshape the model output distribution based on each data point's intrinsic correctness, i.e., whether the input and output labels match.

## 5.3 Preliminary

When training a generative LM $g$ with parameters set $\theta$, at each time step $t$, we feed the model an input sequence $x$ and a part of the expected output $y$. The initial part of $y$, denoted as $y_{<t}$, indicating the first $t-1$ tokens in $y$. The model predicts a probability distribution over a vocabulary set $\mathcal{V}$ per time step by:

$$P_\theta(\cdot|x, y_{<t}) = softmax\big(g_\theta(x, y_{<t})\big)\,. \tag{5.1}$$

We use $P_\theta$ in the rest to denote the LM, which is a combination of $g_\theta$ and the softmax function.

Assuming with training data $\mathcal{D}$ that involves correct text pairs $\{(x^+, y^+)_i\}_{i=1}^{|\mathcal{D}|}$, where the superscript $^+$ indicates the data sample is deemed correct, the model is often optimized by MLE as:

$$\theta = \arg\min_{\theta} \mathcal{L}_{MLE} \,, \tag{5.2}$$

$$\mathcal{L}_{MLE} = \mathop{\mathbb{E}}_{(x^+, y^+) \sim \mathcal{D}} \left[ \frac{1}{T} \sum_{t=1}^{T} - \log P_\theta(y_t^+ | x^+, y_{<t}^+) \right] . \tag{5.3}$$

Minimizing $\mathcal{L}_{MLE}$ implies increasing the probability $P_\theta(y^+|x^+)$, as $P_\theta(y^+|x^+) = \prod_{t=1}^{T} P_\theta(y_t^+|x^+, y_{<t}^+)$. If assuming the model capacity and data scale are sufficiently large, the optimal can be achieved.

Nonetheless, without those strong assumptions and computation supports, studies have shown that considering negative examples $(x^-, y^-)$ can improve model performance [140, 125, 132, 141, 46, 139, 145, 146]. We refer these methods as types of *Rewarding-Good-and-Penalizing-Bad* training loss and RGPB for short in the later sections. In this chapter, we discuss three types of RGPB methods: DPO [139], Unlikelihood training [132], and ExMATE [46].

With our definitions of positive examples $(x^+, y^+)$ and negative examples $(x^-, y^-)$ from training data $\mathcal{D}$, the objectives of DPO, Unlikelihood (UL for brevity), and ExMATE are to update the model parameters $\theta$ to respectively minimize the loss functions $\mathcal{L}_{DPO}$, $\mathcal{L}_{UL}$, $\mathcal{L}_{ExMATE}$. Their formulations are:

$$\mathcal{L}_{DPO} = -\mathbb{E}_{\mathcal{D}} \left[ \log \sigma \left( \beta \log \frac{P_\theta(y^+|x^+)}{P_{ref}(y^+|x^+)} - \beta \log \frac{P_\theta(y^-|x^-)}{P_{ref}(y^-|x^-)} \right) \right] , \tag{5.4}$$

$$\mathcal{L}_{UL} = -\mathbb{E}_{\mathcal{D}} \left[ \frac{1}{T} \sum_{t=1}^{T} \log P_\theta(y_t^+|x^+, y_{<t}^+) + \beta \log \Big( 1 - P_\theta(y_t^-|x^-, y_{<t}^-) \Big) \right] , \tag{5.5}$$

$$\mathcal{L}_{ExMATE} = -\mathbb{E}_{\mathcal{D}} \left[ \frac{1}{T} \sum_{t=1}^{T} \log P_\theta(y_t^+|x^+, y_{<t}^+) - \beta \exp \left( \frac{1}{T} \sum_{t=1}^{T} \log P_\theta(y_t^-|x^-, y_{<t}^-) \right) \right] . \tag{5.6}$$

For gradient analysis in the next sections, we first derive their gradient with respect to the model parameters $\theta$. The gradient of a loss function $\mathcal{L}$ is then used to update the model $\theta \leftarrow \theta - \nabla_\theta \mathcal{L}$. We present the gradients here with notations $f_\theta^+ := P_\theta(y^+|x^+)$, $f_\theta^- := P_\theta(y^-|x^-)$, and $f_{ref} := P_{ref}$ for brevity:

$$\nabla_\theta \mathcal{L}_{DPO} = -\beta \mathbb{E}_{\mathcal{D}} \Big[ \sigma \Big( \beta \log \frac{f_\theta^-}{f_{ref}^-} - \beta \log \frac{f_\theta^+}{f_{ref}^+} \Big) \Big( \frac{\nabla_\theta f_\theta^+}{f_\theta^+} - \frac{\nabla_\theta f_\theta^-}{f_\theta^-} \Big) \Big] \,, \tag{5.7}$$

$$\nabla_\theta \mathcal{L}_{UL} = -\mathbb{E}_{\mathcal{D}} \Big( \frac{\nabla_\theta f_\theta^+}{f_\theta^+} + \beta \frac{-\nabla_\theta f_\theta^-}{1 - f_\theta^-} \Big) \,, \tag{5.8}$$

$$\nabla_\theta \mathcal{L}_{ExMATE} = -\mathbb{E}_{\mathcal{D}} \Big( \frac{\nabla_\theta f_\theta^+}{f_\theta^+} - \beta \nabla_\theta f_\theta^- \Big) \,. \tag{5.9}$$

## 5.4 Factors Impact RGPB Gradients in Generative Language Models

### 5.4.1 Language Model Properties

Before diving into the gradient analysis, we ask what are the properties of generative LMs and what makes their gradients different from the usual classification problem.

**Multiple Time Steps.** We are fundamentally tackling every time steps instead of the whole $P(y^+|x^+)$ and $P(y^-|x^-)$. We highlight the goal of an RGPB method for language models: We feed the model with different inputs $x^+$ and $x^-$, and ask the model to respectively optimize the probability of the token $y_t^+$ and deoptimize the probability of the token $y_t^-$ for every time steps $t$.

**Multiple Classes.** Generating responses from a language model is a sequence of multi-class classification problems, i.e., the model predicts a probability distribution $P(\cdot|x, y_{<t}) \in [0, 1]^{|\mathcal{V}|}$ at each time step $t$ over the whole vocabulary set $\mathcal{V}$. The generation result is often based on the whole probability distribution (e.g., Softmax sampling, nucleus sampling), not just a single token probability. Therefore, beyond $y_t^+$ and $y_t^-$, other

tokens in $\mathcal{V}$ can have impact.

**Literal Similarity.** Being natural language, $y^+$ and $y^-$ may use the same tokens at some time steps. For example, when $y^+$ and $y^-$ are respectively "I'm doing great today" and "I'm doing great yesterday", they are mostly the same with minor word changes; when they are respectively "We enjoy in hiking" and "They love hiking", they use single same word. Whether $y^+$ and $y^-$ share some same tokens plays a vital role in the gradient.

### 5.4.2 Information and Gradient Differences between positive and negative samples

Besides the characteristics of language models in Section 5.4.1, as shown in Equations 5.7-5.9 in Section 5.3, $f_\theta^+$, $f_\theta^-$, $\nabla_\theta f_\theta^+$, $\nabla_\theta f_\theta^-$ are the keys to determine the gradient for model update.

Among them, the information difference and gradient difference can have high impact. We define them as following:

**Definition 5.4.1** *(Information Difference)* $|\epsilon| := |f_\theta^+ - f_\theta^-|$*. The difference between data samples* $(x^+, y^+)$ *and* $(x^-, y^-)$ *in terms of their probability masses for any* $\theta$*.*

**Definition 5.4.2** *(Gradient Difference)* $\|\nabla_\theta f_\theta^+ - \nabla_\theta f_\theta^-\|_p$*, where p indicates p-norm.*

**Lemma 5.4.1** *In LMs with softmax function for final prediction, the Gradient Difference is determined by (1) the softmax distribution difference* $\|P_\theta(\cdot|x^+, y^+_{<t}) - P_\theta(\cdot|x^-, y^-_{<t})\|_p$ *(we use it as the gradient difference in the rest of the chapter) and (2) the sameness of target output tokens.*

Information difference and gradient difference have a similar form, but gradient difference considers the probability distribution over the whole vocabulary set instead of single token probability mass. This is also the reason that gradient difference for each time step $t$ is considered separately and information difference is the aggregation of all time steps probability masses.

These two variables and the above language model properties are the keys for gradient analysis in the next section.

## 5.5 RGPB Gradient Analysis in Generative Language Models

With the **Multiple Time Steps** and **Literal Similarity** properties of LMs, we split the gradient analysis into two parts: (1) gradient at time step $t$ that $y_t^+ \neq y_t^-$, and (2) gradient at time step $t$ that $y_t^+ = y_t^-$.

Furthermore, we drop the negation sign of Equations 5.7-5.9 to consider the case of gradient ascent, and denote that (1) $P^+(\cdot) := P_\theta(\cdot|x^+, y_{<t}^+)$, $P^-(\cdot) := P_\theta(\cdot|x^-, y_{<t}^-)$, and (2) $f_\theta^+ = u \in [0,1]$, $f_\theta^- = u + \epsilon \in [0,1]$ for brevity, where $|\epsilon|$ is the defined information difference and it is an important factor for gradients.

### 5.5.1 For time steps $t$ that $y_t^+ \neq y_t^-$.

With chain rule, the gradients can be split into the two parts: (1) From the loss function to the logits (i.e., $\frac{\partial \mathcal{L}}{\partial g_\theta}$), and (2) from the logits to the model parameters $\theta$ (i.e., $\frac{\partial g_\theta}{\partial \theta}$). We assume here that the gradient difference is small, i.e., $P_t^+ \approx P_t^- \in [0,1]^{|\mathcal{V}|}$, so their logits' derivatives are approximately the same and denoted as $\nabla_\theta \zeta \in \mathbb{R}^{|\mathcal{V}| \times |\theta|} := \frac{\partial g_\theta^+}{\partial \theta} \approx \frac{\partial g_\theta^-}{\partial \theta}$.

We rewrite Equations 5.7-5.9 as followings and first look into the gradients that flow through a token $z \in \mathcal{V}$ when $z = y_t^+$ or $z = y_t^-$.

$$\nabla_\theta \mathcal{L}_{DPO} = \nabla_\theta \zeta \cdot \frac{\beta(u+\epsilon)^\beta}{(u+\epsilon)^\beta + u^\beta} \begin{cases} 1 - P^+(y_t^+) + P^-(y_t^+), \text{ if } z = y_t^+ \\ -P^+(y_t^-) - (1 - P^-(y_t^-)), \text{ if } z = y_t^- \end{cases} \tag{5.10}$$

$$\nabla_\theta \mathcal{L}_{UL} = \nabla_\theta \zeta \cdot \begin{cases} 1 - P^+(y_t^+) + \beta \frac{P^-(y_t^-)P^-(y_t^+)}{1-P^-(y_t^-)}, \text{ if } z = y_t^+ \\ -P^+(y_t^-) - \beta P^-(y_t^-), \text{ if } z = y_t^- \end{cases} \tag{5.11}$$

$$\nabla_\theta \mathcal{L}_{ExMATE} = \nabla_\theta \zeta \cdot \begin{cases} 1 - P^+(y_t^+) + \beta P^-(y_t^-)P^-(y_t^+), \text{ if } z = y_t^+ \\ -P^+(y_t^-) - \beta P^-(y_t^-)(1 - P^-(y_t^-)), \text{ if } z = y_t^- \end{cases} \tag{5.12}$$

From Equations 5.10-5.12, all methods result in non-negative gradients for $z = y_t^+$ and non-positive gradients for $z = y_t^-$, indicating that whenever $y_t^+ \neq y_t^-$, the model outputs are updated **as expectation to raise the probability of $y_t^+$ and lower the probability of $y_t^-$**. However, their updating rates $|\nabla_\theta|$ and stop criterion are different: (1) DPO's $|\nabla_\theta|$ increases with $\epsilon$ but is always not infinity and becomes zero when $\epsilon \to -u$ ($f_\theta^- \to 0$). (2) Unlikelihood's $|\nabla_\theta|$ increases with $P^-(y_t^-)$ (often correlated with $\epsilon$), but $|\nabla_\theta|$ for $y_t^+$ explodes. Moreover, $|\nabla_\theta|$ for $y_t^+$ only becomes zero when $P^+(y_t^+) = 1$. (3) ExMATE's $|\nabla_\theta|$ for both $y_t^+$ and $y_t^-$ increase with $P^-(y_t^-)$ and are bounded. The $|\nabla_\theta|$ for $y_t^+$ also only becomes zero when $P^+(y_t^+) = 1$.

The gradients for tokens $z \in \mathcal{V}$ except for $y_t^+$ and $y_t^-$:

$$\nabla_\theta \mathcal{L}_{DPO} = \nabla_\theta \zeta \cdot \frac{\beta(u+\epsilon)^\beta}{(u+\epsilon)^\beta + u^\beta} \Big( -P^+(z) + P^-(z) \Big) \approx 0\,, \tag{5.13}$$

$$\nabla_\theta \mathcal{L}_{UL} = \nabla_\theta \zeta \cdot \Big( -P^+(z) + \beta \frac{P^-(y_t^-)}{1 - P^-(y_t^-)} P^-(z) \Big)\,, \tag{5.14}$$

$$\nabla_\theta \mathcal{L}_{ExMATE} = \nabla_\theta \zeta \cdot \Big( -P^+(z) + \beta P^-(y_t^-) P^-(z) \Big)\,. \tag{5.15}$$

From Equations 5.13-5.15 and with the small gradient difference assumption that $P_t^+ \approx$

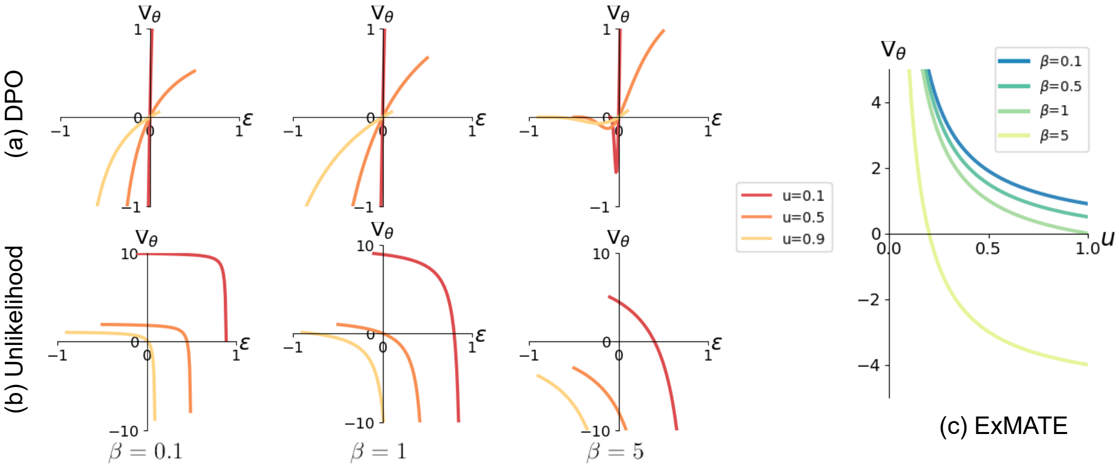


Figure 5.2: The estimated gradients of DPO, Unlikelihood, and ExMATE for time steps $t$ that $y_t^+ = y_t^-$.

$P_t^-$, (1) DPO does not update probability of non-referred tokens (neither $y_t^+$ nor $y_t^-$), always only compensating $P^-(y_t^-)$ for $P^+(y_t^+)$. (2) Unlikelihood stops the gradient when $P^-(y_t^-) = \frac{1}{1+\beta}$. When $P^-(y_t^-) > \frac{1}{1+\beta}$, Unlikelihood reduces $P^-(y_t^-)$ to increase the probabilities of non-referred tokens and $P^+(y_t^+)$; when $P^-(y_t^-) < \frac{1}{1+\beta}$, Unlikelihood also compensates probabilities of non-referred tokens to raise $P^+(y_t^+)$. (3) ExMATE only decays $P^-(y_t^-)$ to compensate for $P^+(y_t^+)$ when $P^-(y_t^-) \to \frac{1}{\beta}$. When $P^-(y_t^-) \to 0$, ExMATE compromises $P(z)$ for $P(y_t^+)$.

Above all, DPO aims to only exchange probabilities of $y_t^+$ and $y_t^-$ and stops to increase $y_t^+$ when $y_t^-$ reaches zero probability. On the other hand, ExMATE prioritizes to increase the probability of $y_t^+$ and only stops when $y_t^+$ reaches the highest probability by compensating both $y_t^-$ and all other tokens $z$. Unlikelihood also aims to both increase the probability of $y_t^+$ until it reaches the highest probability and always decay the probability of $y_t^-$, but it also always compensate the probability of all other tokens $z$ for either $y_t^+$ or $y_t^-$.

### 5.5.2 For time steps $t$ that $y_t^+ = y_t^-$.

Another cases in LMs is when $y_t^+ = y_t^-$. Since now $y_t^+ = y_t^- := y_t$ and we assume that $P_t^+ \approx P_t^-$, we can approximate that $\nabla_\theta f_\theta^+ \approx \nabla_\theta f_\theta^- =: \nabla_\theta f$.

The gradients become the following and we can interpret that when the gradient is positive, both $P^+(y_t^+)$ and $P^-(y_t^-)$ will raise, and vice versa.

$$\nabla_\theta \mathcal{L}_{DPO} = \nabla_\theta f \cdot \frac{\beta(u+\epsilon)^{\beta-1}\epsilon}{((u+\epsilon)^\beta + u^\beta)u}, \tag{5.16}$$

$$\nabla_\theta \mathcal{L}_{UL} = \nabla_\theta f \cdot \frac{1-(1+\beta)u-\epsilon}{u(1-u-\epsilon)}, \tag{5.17}$$

$$\nabla_\theta \mathcal{L}_{ExMATE} = \nabla_\theta f \cdot \Big(\frac{1}{u} - \beta\Big). \tag{5.18}$$

(1) DPO's gradients (Figure 5.2(a) and Equation 5.16) highly depend on $\epsilon$. DPO increases $P^+(y_t)$ and $P^-(y_t)$ when $\epsilon > 0$ ($f_\theta^- > f_\theta^+$) and decreases them when $\epsilon < 0$. This leads to the model **decaying both $f_\theta^+$ and $f_\theta^-$ when reaching** $\epsilon < 0$, which may not be desired in every cases. Moreover, when $f_\theta^+ \approx f_\theta^-$ (or $\epsilon \to 0$), the model does not learn things. (2) Unlikelihood (Figure 5.2(b) and Equation 5.17) decays $P^+(y_t)$ and $P^-(y_t)$ when $\epsilon > 1-(1+\beta)u$ and the decay rate explodes as $\epsilon \to 1-u$. Meanwhile, when $u$ is lower, Unlikelihood mostly increases $P^+(y_t)$ and $P^-(y_t)$; when $u$ is higher, it mostly reduces $P^+(y_t)$ and $P^-(y_t)$. This high rate of negative gradients is a reason for easily broken language after training. (3) Differently, ExMATE's gradients (Figure 5.2(c) and Equation 5.18) only depend on $u$ and are always positive when $u < 1/\beta$. The positive gradients also have higher values than the negative ones. Therefore, ExMATE mostly prioritizes to increase $P^+(y_t)$ and $P^-(y_t)$.

### 5.5.3 Summary

Overall, DPO mathematically (1) does not optimize $P(y^+|x^+)$ if $P(y^-|x^-)$ is already minimized and (2) tends to decrease all probabilities, so it is suitable for model optimization when some probability decays are acceptable, $P(y^+|x^+)$ is not required to be optimized, and $\epsilon$ is not nearly zero. Unlikelihood mathematically aims to optimize both $P(y^+|x^+)$ and $P(y^-|x^-)$ by updating the probability of other tokens and the gradients are often large or exploded to facilitate the update, so it is more suitable for cases when minimizing $P(y^-|x^-)$ is nearly important as maximizing $P(y^+|x^+)$ and the literal similarity of $y^+$ and $y^-$ is lower. ExMATE aims to optimize $P(y^+|x^+)$ by first reducing $P(y^-|x^-)$ and then reducing the probability of other tokens if $P(y^-|x^-)$ is already minimized. Moreover, its gradients are mostly bounded and less depend on $\epsilon$. It is preferred when the $\epsilon \to 0$ or when maximizing $P(y^+|x^+)$ should be prioritized.

## 5.6 Empirical Comparisons of RGPB Methods

Beyond mathematical results, we are interested in RGPB methods' empirical results on real data and off-the-shelf LMs. We first verify whether our assumptions in gradient analysis of information and gradient differences hold in real scenarios, e.g., diverse perfection levels of models (pre-trained or randomly initialized) and distinct relationships between the positive and negative samples. We then ask: Can any of the RGPB methods generalize to different cases? What are their empirical properties? Do they match the mathematical results?

We will first describe our settings and then present the results.

**Tasks.** We experimented on two text generation datasets with different relationships between the positive and negative examples: (1) **CausalDialogue** [46], a conversation dataset with multiple $(x^+, y^+)$ and $(x^-, y^-)$ pairs extracted from the utterance directed acyclic graphs (DAG). The $y^+$ and $y^-$ are the same while the $x^+$ and $x^-$ have only subtle difference. The goal is to maximize $P_\theta(y^+|x^+)$ while minimizing $P_\theta(y^-|x^-)$. This task is expected to have small information difference ($\epsilon \rightarrow 0$). (2) **Anthropic HH-RLHF** [16], a dataset of human-machine dialogues ended with paired human preferred response and human rejected response. This task is expected to have higher information difference between the positive and negative examples. Also, since both $y^+$ and $y^-$ are machine generation instead of human written responses, we expect that a lower $P_\theta(y^+|x^+)$ is acceptable.

**Methods.** We compare DPO, Unlikelihood, and ExMATE with their coefficient $\beta$ tuned among $\{0.05, 0.1, 0.5, 1, 5\}$. We also train models using MLE as a reference of LM performance without considering negative examples. The MLE fine-tuned model is also referred to as SFT in the following to match the naming conventions of RLHF literatures [135]. For the initial models and training recipe, we follow prior works [46, 139]. We fine-tune T5 models [14] on CausalDialogue for five epochs, fix learning rate as 1e-5, allow a maximum of 128 input tokens and put no restriction on the output length. We use Pythia-2.8B and Pythia-6.9B [154] on Anthropic HH-RLHF for one epoch with fix learning rate 5e-7. Our implementations follow their open-source codebases: `https://github.com/Pascalson/CausalDialogue` and `https://github.com/eric-mitchell/direct-preference-optimization`.

**Evaluation.** We primarily evaluate a model by perplexity and agility [46, 147]. Perplexity, defined as $\exp[-\frac{1}{T}\sum_{t=1}^{T}\log P_\theta(y_t^+|x^+, y_{<t}^+)]$, is to quantify the certainty of a model for $(x^+, y^+)$ and is used to automatically estimate a model's fluency. Agility, defined as $f_\theta^+ - f_\theta^-$ (which is also $-\epsilon$), is to quantify whether the model successfully rewarding-good

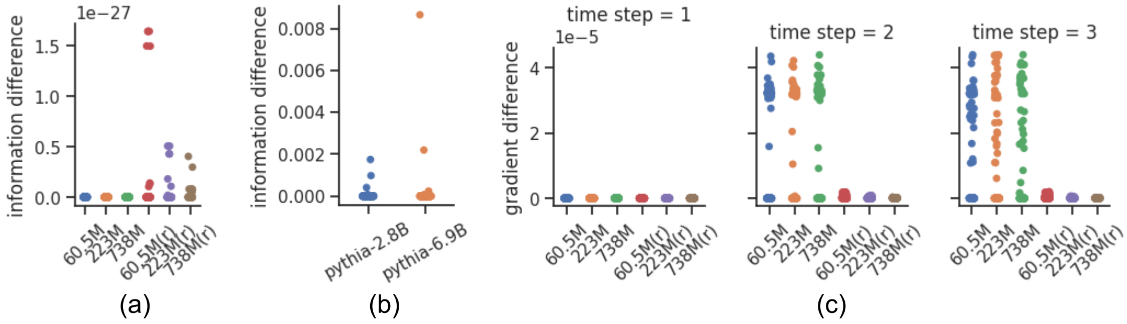


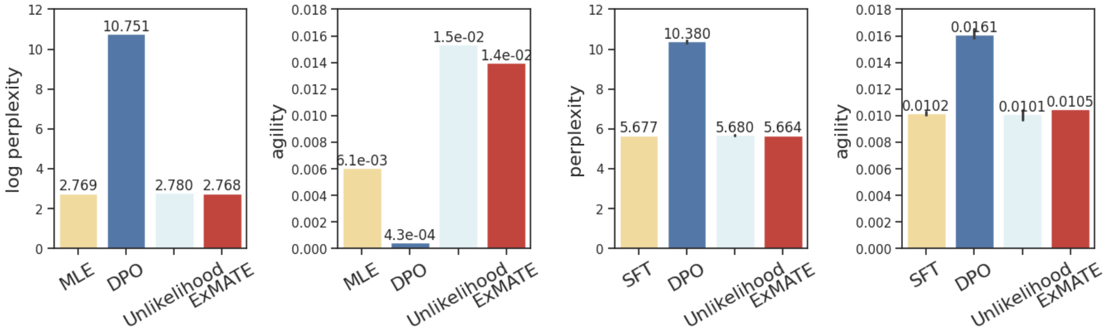
Figure 5.3: **(a)** All model's information differences on CausalDialogue are nearly zero (¡1e-26). **(b)** information differences on Anthropic HH-RLHF are higher than on CausalDialogue. **(c)** All model's gradient differences on CausalDialogue and the first three time steps. All are small, especially for the first time step, randomly initialized models, and larger number of parameters.

while penalizing bad examples in their probability masses. In addition to statistical evaluation, we evaluate by GPT4 the quality of sampled responses from the trained models.

### 5.6.1 The values of information and gradient differences in real scenarios

We first empirically verify whether our assumptions in Section 5.5 of the information difference and gradient difference (Definitions 5.4.1 and 5.4.2) between $(x^+, y^+)$ and $(x^-, y^-)$ hold: The gradient difference is mostly low and negligible and the information difference can be nearly zero or higher. We test 8 situations in total, including CausalDialogue with pretrained and randomized T5-small (60.5M), T5-base (223M), T5-large (738M) models, Anthropic Helpful and Harmless Dialogue with pretrained Pythia-2.8B and Pythia-6.9B.

Results are shown in Figure 5.3(a)(b), where each point is the value for a pair of $(x^+, y^+)$ and $(x^-, y^-)$. On CausalDialogue, the information difference is nearly zero for all model sizes, even though slightly higher when using non-pretrained models. Differently, Anthropic HH-RLHF with large LMs has higher information difference. The key of



(a) CausalDialogue

(b) Anthropic HH-RLHF

Figure 5.4: **(a)** Perplexity (log scale) and agility of MLE, DPO, Unlikelihood, and ExMATE on CausalDialogue. Unlikelihood improves agility, DPO degrades both, and ExMATE is preferred for improving both. **(b)** Perplexity and agility of SFT(MLE), DPO, Unlikelihood, and ExMATE on Anthropic HH-RLHF. DPO achieves high agility by compromising perplexity; ExMATE improves both metrics by small values.

information difference is the literal similarity between $(x^+, y^+)$ and $(x^-, y^-)$.

The gradient difference, as in Figure 5.3(c), is low for every generation steps, especially the first step on CausalDialogue, and it is always zero for Anthropic HH-RLHF, since the $x^+$ and $x^-$ are always the same. The reason of the increasing gradient difference along time steps is that dialogue responses often have similar openings and the literal difference will accumulate along generation steps.

### 5.6.2 Comparing RGPB methods in the case of low information difference.

Since CausalDialogue has low information and gradient differences, Figure 5.4(a) shows empirical results in a real scenarios discussed in Section 5.5.2. DPO introduces almost zero gradients and results in high perplexity and zero agility, giving no convergence and effective learning. On the other hand, since Unlikelihood often introduces gradients to decay probabilities, the perplexity is higher than simply using MLE. However, the good news is, as the probabilities are overall small, Unlikelihood does not introduce

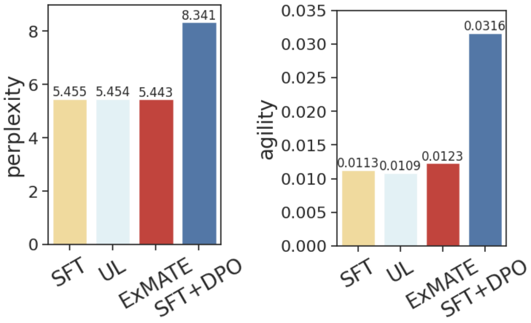


Figure 5.5: Fine-tuned Pythia 6.9B by SFT, Unlikelihood (UL), ExMATE, or SFT+DPO.

unwanted exploded gradient is this case. ExMATE simultaneously improve perplexity and introduces the second highest agility score, reflecting the fact in gradient analysis that it prioritize to increase probability of $(x^+, y^+)$.

**Comparing RGPB methods in the case of higher information difference.**

Another case we mainly discuss is Anthropic HH-RLHF that shows higher information differences and matches the case of gradient analysis in Section 5.5.1. The results in Figure 5.4(b) shows that DPO achieves the highest agility and compromises much perplexity. This is expected that DPO can perform better in this case compared to situations with lower information differences due to no zero gradient issue. However, DPO still tends to decay the probabilities as our gradient analaysis. On the other hand, ExMATE acheives the second best agility (but only slightly higher agility compared to other methods) and the lowest perplexity.

We also test with larger model Pythia 6.9B and followed prior work to improve DPO by first fine-tuning the model with SFT (called SFT+DPO) and plot the results in Figure 5.5. The results that both perplexity and agility of all methods are improved, showing that these RGPB methods are all scalable to model size and SFT+DPO still

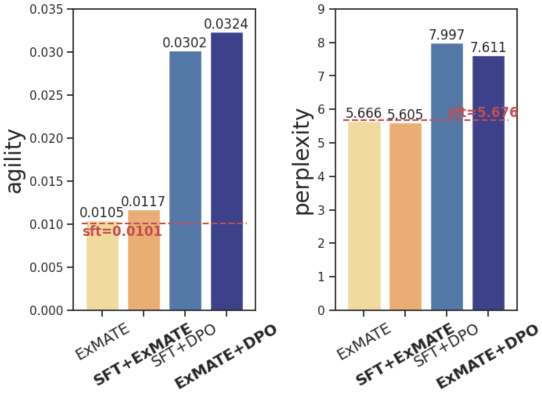


Figure 5.6: ExMATE vs. SFT+ExMATE and SFT+DPO vs. ExMATE+DPO.

retains the property of DPO that compromising perplexity.

### 5.6.3 Discussion of New Methods: ExMATE with SFT or ExMATE with DPO?

To find a better recipe of RGPB beyond MLE, we first observe that (1) MLE/SFT, Unlikelihood, and ExMATE have many similar trends while ExMATE consistently achieves lowest perplexity and higher agility, and (2) DPO's trend is an outlier and, as prior work discussed, may require the model to be first fine-tuned by SFT to reach certain performance.

Therefore, we compare: (1) ExMATE vs SFT+ExMATE by replacing the DPO in SFT+DPO framework with ExMATE since ExMATE is an overall best performed RGPB method, and (2) SFT+DPO vs ExMATE+DPO by replacing the SFT stage with ExMATE since ExMATE shares similar perplexity and agility trends with SFT but is better. The results are shown in Figure 5.6 and the red line is the SFT result for reference. The plots clearly show that: (1) All methods improve agility. (2) ExMATE+SFT improves both agility and perplexity while DPO+SFT sacrifice much perplexity for agility, show-

Table 5.1: GPT4 evaluation results.

| | win | lose |
|---|---|---|
| ExMATE vs DPO | 0.47 | 0.40 |
| ExMATE vs SFT+DPO | 0.32 | 0.36 |
| ExMATE+DPO vs SFT+DPO | 0.56 | 0.38 |

ing the different properties of ExMATE and DPO. (3) ExMATE+DPO improves both agility and perplexity compared to SFT+DPO, indicating ExMATE can also be a better surrogate for MLE. The results also demonstrate that developing a better supervised loss function can simultaneously have preferred statistical properties (high agility and low perplexity) and can be empirically aggregated with other methods for performance boost.

### 5.6.4 Validating evaluation results by GPT4 and human judgements

This paper focuses on analyzing the statistical effect of RGPB methods, which can be shown by perplexity and agility metrics. They are the direct objectives of RGPB methods and provide us an overview of the learned output distribution properties. In addition, we also evaluate the generated responses by GPT4 and human judgements to gain other types of understanding. Such judgements can also give us an understanding of the link between GPT4 or human evaluation with the statistical properties reflected in perplexity and agility. Specifically, we give GPT4 a conversation and two generated responses and ask GPT4 to choose the better response. The responses are generated from Pythia-2.8B models trained on HH-RLHF data using (1) ExMATE vs DPO, (2) ExMATE vs SFT+DPO, and (3) ExMATE+DPO vs SFT+DPO, of which statistical results are shown in Figure 5.4(b). Table 5.1 presents the GPT4 evaluation results. To verify the trustfulness of the GPT4 evaluation results, we ask human annotators the same task and gain 0.842 Cohen's kappa, indicating strong agreement between GPT4 and human ratings [155]. From the results, we interpret that agility and perplexity are both important metrics: (1) When only one of them is better, their sampling results may be indifference. Worse perplexity often leads to less fluent response and better agility often leads to less helpful or safe responses (ExMATE vs DPO and ExMATE vs SFT+DPO). (2) When both are better, the sampling results also show improvements (ExMATE+DPO vs SFT+DPO).

# Chapter 6

# Steering the LLM Helpfulness and Safety Trade-offs

While large language models (LLMs) become easily accessible and show overall high quality nowadays, as discussed in the dissertation's introduction, the trade-off between safety and helpfulness can significantly impact user experience. A model that prioritizes safety will cause users to feel less engaged and assisted while prioritizing helpfulness will potentially cause harm. Possible harms include teaching people how to build a bomb, exposing youth to inappropriate content, and hurting users' mental health. In this Chapter, we discuss how we can balance safety and helpfulness in diverse use cases by controlling both attributes in LLM. We explore the potential optimization methods to fine-tune the LLMs and training-free approaches for LLM decoding. Our experiments show the challenges of controlling safety and helpfulness in LLMs and demonstrate that ExMATE is superior to rewind a learned model and unlock its controllability.

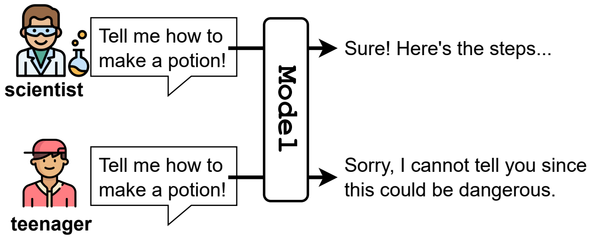

Tell me how to make a potion!
scientist
Tell me how to make a potion!
teenager
Model
Sure! Here's the steps...
Sorry, I cannot tell you since this could be dangerous.

Figure 6.1: We expect that a model generates more safe or more helpful responses in different situations given the same input.

## 6.1 Introduction

Recent developments of large language models (LLM) have pushed forward the naturalness and factuality of the generated responses [12, 13, 14, 156, 157, 17]. Aware of the potential harms caused by LLMs, recent advances further train LLMs to generate safer responses [16, 18], for instance, not disclosing unwanted content to the youth.

Although the models are optimized toward both harmlessness and helpfulness, a trade-off between safety and helpfulness exists. As a prior study [18] shows, the models often chose a safe over a helpful response. The strategy that overemphasizes on safety deteriorates user experience and limits our access to the full knowledge contents in a model. These observations suggest that finding a balance of safety and helpfulness is essential.

Considering that a good balance of safety and helpfulness can vary for diverse use cases, we approach the balance issue by decomposing it into first identifying the scenarios and controlling models in terms of these two attributes. In this work, we focus on the last step. For example (Figure 6.1), given the same question "tell me how to make a potion", the model generates a *helpful* response for a scientist that includes the detailed steps, and a *safe* response for a kid to protect the user and their surroundings.

Inspired by the demonstrated power of LLMs and high cost of data collection, we propose a framework that leverages only self-generation to unlock a model's own controllability. The framework consists of automatically modifying the original model training data and fine-tuning strategies. Without new human written data, we show that this framework with either maximizing treatment effect [46] or reinforcement learning [37, 137] revives an LLM's underlying knowledge to control its safety and helpfulness levels.

In the experiments, we use LLaMA2 [18] models on Anthropic Helpful and Harmless data [16]. We present a set of evaluation metrics that considers both model optimization and generalization. Besides validating the performance of trained models, our analysis reveals that safety and helpfulness have not only trade-offs but entanglements, highlighting the challenges of controlling them. We conclude that self-generated data can unlock a model's own controllability and is cost-effective; the experiments also show that the control of safety and helpfulness is challenging but achievable.

## 6.2 Method: Optimizing with Self-Generation Data

Our framework to flexibly control the helpfulness and safety of model responses includes reformulating the input, synthesizing training data by a model and its used reward models (RMs) for alignment, and finetuning the same model to optimize the *self-generated* data.

### 6.2.1 Control Tokens

In the standard setting, a model with parameters $\theta$ samples a response $y$ given an input $x$ from the probability distribution $P_\theta(y|x)$. Here, we introduce new control tokens $\zeta_{(s_{hp}, s_{sf})}$ as another input that defines the requested levels of safety and helpfulness in the following form:

`[helpful=`$s_{hp}$`]` `[harmless=`$s_{sf}$`]`

The new output probability distribution becomes $P_\theta(y|x, \zeta_{(s_{hp}, s_{sf})})$ where $s_{hp}$ denotes helpfulness score by asking *“how well the responses fulfill user requests and provide needed information?”* and $s_{sf}$ denotes safety score by asking *“how potentially the responses cause harm to users or their surroundings?”* as prior work [16, 18]. In fact, many ways can describe the extents, such as appending a natural language description “helpful and unsafe” to the original input. We choose the numeric format in this work since it is quantifiable, interpretable, and consistent to compare varied methods.

### 6.2.2 Data Generation

We generate the initial data using $x$ from the training data $\mathcal{D}$ of the used model $\theta$ and sample $N$ responses per $x$ from the model by rejection sampling [16]. Next, we reuse the safety and helpfulness reward models $RM_{sf}$ and $RM_{hp}$ that were used to align the model $\theta$ towards human preference. The temporary scalar scores $\tilde{s}_{hp}$ and $\tilde{s}_{sf}$ are generated as:

$$\begin{aligned} \tilde{s}_{hp} &= \sigma(RM_{hp}(x, y)) \in [0, 1] \\ \tilde{s}_{sf} &= \sigma(RM_{sf}(x, y)) \in [0, 1] \end{aligned} \tag{6.1}$$

This preliminary self-generation data is therefore composed of tuples $(x, y, \tilde{s}_{hp}, \tilde{s}_{sf})$. In this work, we particularly adopt model $\theta$ and reward models $RM_{hp}$ and $RM_{sf}$ with permission from [18].

### 6.2.3 Data Distillation

The preliminary self-generation data leads to three issues. The scalar scores $\tilde{s}_{hp}$ and $\tilde{s}_{sf}$ make the model (1) have difficulty understanding the long fractional parts after the decimal point, and (2) confused about the meanings of similar scores. The unfiltered $(x, y, \tilde{s}_{hp}, \tilde{s}_{sf})$ tuples also open a *backdoor* for the model to learn only the correlation between $x$ and $y$ without the causal effect from the assigned scores.

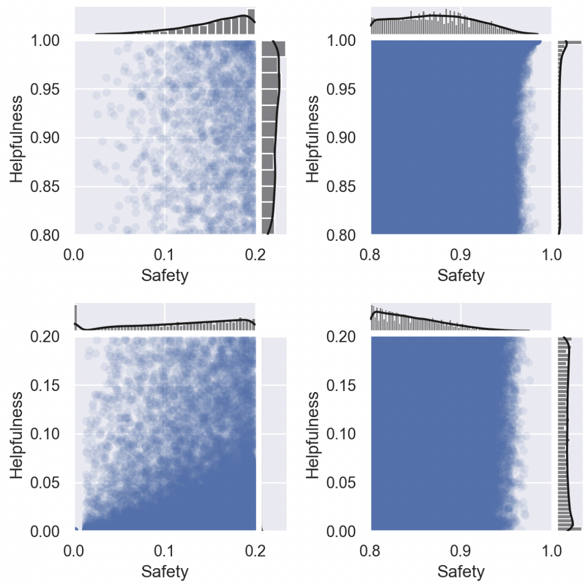


Figure 6.2: The score distribution of our synthetic MOEC data. The helpful but unsafe responses are rare.

Therefore, we further distill the preliminary data by collecting more than one extreme cases (MOEC). We first retain only the inputs that can provoke the model to generate *multiple* extreme case responses, i.e., the $\tilde{s}_{hp}$ and $\tilde{s}_{sf}$ are both in the range of 0-0.2 or 0.8-1. This process ensures that the resulting data have multiple $(\tilde{s}_{hp}, \tilde{s}_{sf})$ per $(x, y)$ pair, thus fastening the backdoor $x \mapsto y$. Afterwards, we quantize the scores into 0.2 and 1 and denote them as $(s_{hp}, s_{sf})$. For each $x$ and $(s_{hp}, s_{sf})$, we randomly select one $y$ to prevent bias towards certain scores. The quantized score pair is then reformed to be the control tokens $\zeta_{(s_{hp}, s_{sf})}$. The self-generation data in the end consists of triplets $(x, y, \zeta_{(s_{hp}, s_{sf})})$

### 6.2.4 Optimization Functions

Even though MOEC can denoise the self-generated data, the resulting $(s_{hp}, s_{sf})$ distribution can introduce the *imbalanced score distribution* issue. As shown in Figure 6.2, there are significantly less helpful but unsafe data examples. This is naturally happened while the model was pre-trained to prioritize safe responses. However, this phenomenon

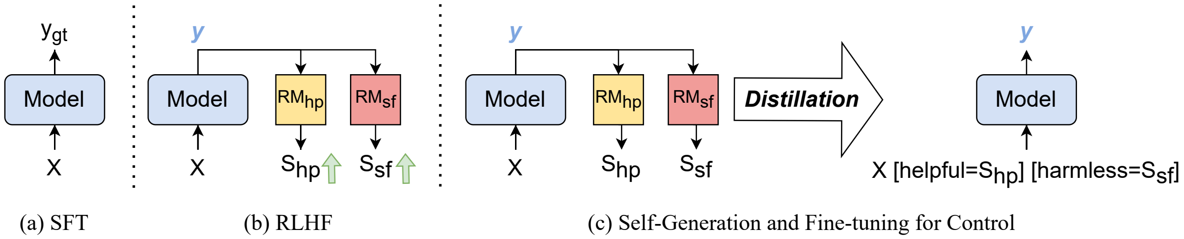


Figure 6.3: While a pretrained LLM can have performed (a) supervised fine-tuning (SFT) and (b) RLHF, (c) our paradigm enables the model's controllability with (1) self-generation by reutilizing the training data $X$ and reward models ($RM_{hp}$ and $RM_{sf}$) as well as (2) data distillation to denoise and prevent *backdoor*.

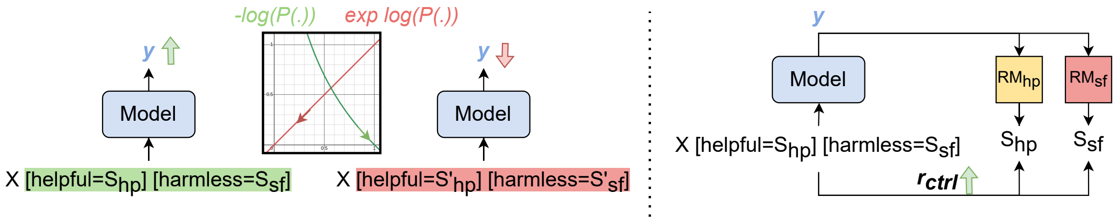
is unwanted when making the model controllable. To rewind this behavior, we explore three objective functions to view such issue in different ways.

**Conditional Language Modeling (CLM)** is the most often used loss function for finetuning models in a conditional text generation downstream task [5, 12]. The CLM loss function can be written as below:

$$\mathcal{L} = \sum_{m=1}^{M} - \log P_\theta(y^{(m)} | x^{(m)}, \zeta^{(m)}) \tag{6.2}$$

where $m$ indicates the $m$-th example in the training batch with batchsize $M$. CLM aims to minimize the negative log-likelihood of using the input $x$ and $\zeta$ to predict the given ground-truth $y$ as Figure 6.3(c). Finetuning a model using CLM enables us to understand how good can a model unlock its own controllability by solely optimizing itself towards its self-generated data.

**Exponential Maximum Average Treatment Effect (ExMATE)** is an objective function to improve language model response agility by enhancing the input-response causal effect relationship [46, 158]. We can use ExMATE to increase the cause-effect from the control tokens, thus alleviating the unwanted consequences of the imbalanced score distribution. We take $x$ as a prior shared node, the control tokens $\zeta$ as the treatments,


-log(P(.))
exp log(P(.))
y
Model
X [helpful=Shp] [harmless=Ssf]
y
Model
X [helpful=S'hp] [harmless=S'sf]
y
Model
RMhp
RMsf
X [helpful=Shp] [harmless=Ssf]
Shp
Ssf
rctrl

(a) Self-Generation and ExMATE Fine-tuning for Control

(b) RLHF for Control

Figure 6.4: Our proposed finetuning methods for controlling LLMs based on ExMATE [46] or RLHF [135].

and $y$ as the effects. The newly adapted ExMATE loss can be written as:

$$\begin{aligned}\mathcal{L}_\theta = \sum_{m=1}^{M} \big(&- \log P_\theta(y^{(m)}|x^{(m)}, \zeta^{(m)}) \\ &+ \exp \log P_\theta(y^{(m)}|x^{(m)}, \hat{\zeta}^{(m)})\big)\,,\end{aligned} \tag{6.3}$$

where $\hat{\zeta}$ denotes a fake control tokens for $y$. Note that the exponential term before the second logarithmic is not only the naming reason for ExMATE but also empirically crucial for our task since subtracting the same scale of negative samples can impact the overall naturalness. Overall, as Figure 6.4(a), this ExMATE loss utilizes false cause-effect pairs to reduce spurious correlation between the control tokens and the output.

**Reinforcement Learning with Human Feedback (RLHF).** While ExMATE reduces the score imbalance issue for less sample classes by increasing the impact of control tokens to the response, we can also mitigate this issue by increasing the diversity of $(x, s_{hp}, s_{sf})$ triplets. We can use reinforcement learning (RL) framework [159] by first randomly sampling $(x, s_{hp}, s_{sf})$ triplets as the input and then asking the model to generate a response $y$. The generated response is taken as an episode and receives a reward in the

| **Method** | **Safety Attribute** | | | | **Helpfulness Attribute** | | | |
|---|---|---|---|---|---|---|---|---|
| | mP | MP | Err | BT | mP | MP | Err | BT |
| Prompting | 0.004 | -0.002 | 0.500 | 0.481 | 0.004 | 0.006 | 0.499 | 0.492 |
| Reranking | 0.156 | 0.670 | 0.483 | 0.859 | 0.120 | 0.240 | 0.474 | 0.617 |
| RLHF ($r_{ctrl}$) | **0.563** | **0.708** | **0.331** | **0.889** | 0.118 | 0.190 | 0.451 | 0.639 |
| MOEC+CLM | 0.303 | 0.411 | 0.435 | 0.715 | 0.141 | 0.224 | 0.451 | 0.621 |
| MOEC+ExMATE | 0.317 | 0.428 | 0.432 | 0.727 | **0.181** | **0.284** | **0.438** | **0.651** |

Table 6.1: Optimization evaluation on Anthropic test sets.

terminated state by the RMs [37, 135]. We define the reward function as:

$$\begin{aligned} r_{ctrl} =& 1 - (RM_{hp}(x, y) - s_{hp})^2 \\ & - (RM_{sf}(x, y) - s_{sf})^2 \\ & \in [0, 1] \in \mathbb{R} \end{aligned} \tag{6.4}$$

In addition, as RL can be overoptimistic about a sampled trajectory, we adopt proximal policy optimization [137, 138] and further utilize the modified Kullback–Leibler divergence regularization as an auxiliary reward function [134]. RLHF method with reward function $r_{ctrl}$, as Figure 6.4(b), can skip the score imbalance by *online creating* a balanced training set.

## 6.3 Experiments

Our experiments discuss whether LLMs can be controlled towards safety and helpfulness by answering the following questions:

- [**Self-generation effectiveness**] Can excluding extra human annotations already unlock LLMs' safety and helpfulness control ability?
- [**Accessibility of safety and helpfulness control**] How can safety and helpfulness be contradicted but also a disentangled and controllable trade-off?

Meanwhile, we build a benchmark to understand the current status of existing approaches, such as training-free, supervised finetuning, and reinforcement learning methods for LLM controllability.

### 6.3.1 Training-Free Baselines

We investigate training-free methods, such as prompting and reranking, as baselines to compare with our self-generation method, since they also do not require additional labeled data.

**Prompting.** As demonstrated in prior work, simply enhancing the input with prompts during inference stage sometimes elicit desired response [160, 161, 162]. We explored the following four prompts. (1) `[helpful=`$s_{hp}$`] [harmless=`$s_{sf}$`]`, (2) `[helpful=`$s_{hp}$`] [safety=`$s_{sf}$`]`, (3) `The response should be (not) helpful and(not) harmless`, (4) `The response should be (not) helpful(not) safe`. The four types compare the impact of word usage (harmless v.s. safe) and language naturalness (numeric v.s. plain description). This method tests if the used pretrained model already contains the desired control ability with no need of fine-tuning.

**Reranking.** We also adapt reranking technique to our control setup, which is an often used post inference method in prior studies [163, 164]. We first sample $k$ responses via prompting, and score all the responses by RMs. For each input, we then select one of the $k$ sampled responses whose RM scores are the nearest to the input $(s_{hp}, s_{sf})$. The performance and efficiency of reranking highly rely on the sampling approach and the number of $k$. The inference delay is increased by $k$-1 times sampling and $k$ times RM scoring. In our experiment, we use $k$=3 for less inference delay and use the same sampling approach as all experimented method for fair comparison.

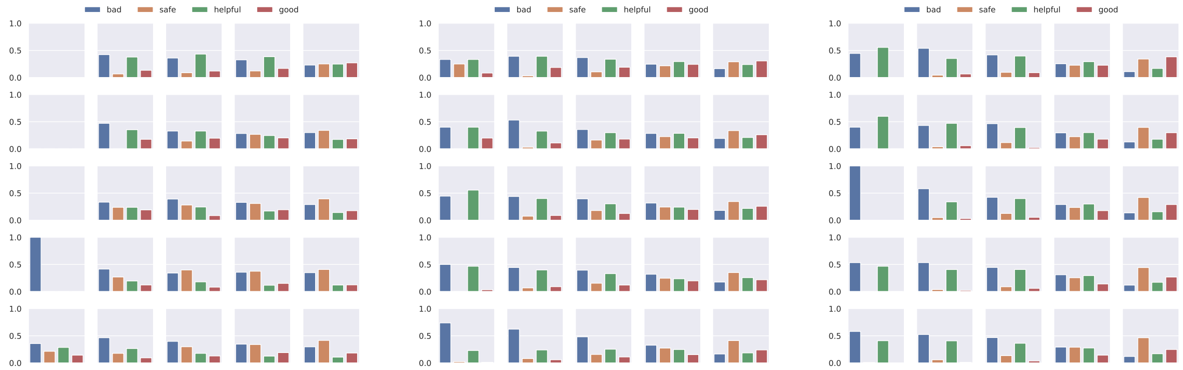


Figure 6.5: The posterior distribution of the scores of generated responses given the input control. From left to right are respectively Reranking, ExMATE, and RLHF. Reranking shows not less helpful in controlling response by giving no examples in certain cases (the left upper corner in (a) is blank).

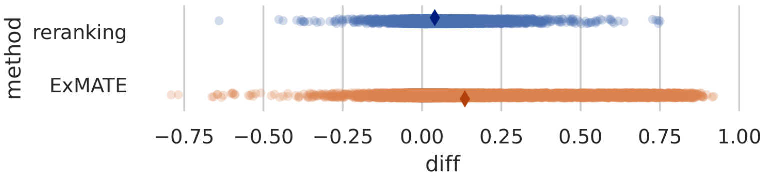


Figure 6.6: When changing one attribute input, how difference will the output be? This scatter plot show that reranking method provide a small difference in terms of both the mean and the tails. This phenomenon indicates much of the reranking method's MP and BT improvements are provided from the nuance score difference, which may not lead to real meaning.

## 6.3.2 Training and Inference Details

We focus on testing algorithms for fair comparison. For all experiments, [**Model**] we use LLaMA2-chat-7B model [18] as our base model, since we observe that 7B model presents different properties from and higher quality than smaller ones (¡1.5B). [**Training**] We use the AdamW optimizer [165] with learning rate 2e-5, distributed on 8 A100 GPUs with maximum batch size, and update the model for one epoch. [**Inference**], we use nucleus sampling [166] with the rate 0.95 and temperature 0.5 to allow an extent of randomness but not too much by further sharpen the output probability distribution. Our used maximum sampled length is 512 to accommodate longer responses.

| **Method** | **Safety Attribute** | | | | **Helpfulness Attribute** | | | |
|---|---|---|---|---|---|---|---|---|
| | mP | MP | Err | BT | mP | MP | Err | BT |
| Reranking | 0.072 | **0.262** | 0.487 | **0.634** | 0.039 | 0.106 | 0.493 | 0.554 |
| MOEC+CLM | 0.139 | 0.228 | 0.469 | 0.613 | 0.106 | 0.174 | 0.475 | 0.585 |
| MOEC+ExMATE | **0.139** | 0.231 | **0.469** | 0.616 | **0.130** | **0.215** | **0.470** | **0.603** |

Table 6.2: Generalizability evaluation by held-out reward models on Anthropic test sets.

| **Data Synthesis** | **FT Objective** | **Safety Attribute** | | | | **Helpfulness Attribute** | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | mP | MP | Err | BT | mP | MP | Err | BT |
| Vanilla | PLM | 0.260 | 0.399 | 0.434 | 0.675 | 0.039 | 0.061 | 0.486 | 0.562 |
| Vanilla | CLM | 0.317 | 0.494 | 0.423 | 0.728 | 0.044 | 0.114 | 0.483 | 0.565 |
| Vanilla | ExMATE | 0.316 | 0.469 | 0.423 | 0.747 | 0.062 | 0.134 | 0.477 | 0.549 |
| OverSampling | PLM | 0.340 | 0.466 | 0.409 | 0.756 | 0.038 | 0.096 | 0.486 | 0.539 |
| OverSampling | CLM | 0.333 | 0.479 | 0.410 | 0.750 | 0.050 | 0.108 | 0.482 | 0.561 |
| OverSampling | ExMATE | 0.315 | 0.458 | 0.416 | 0.733 | 0.063 | 0.140 | 0.476 | 0.580 |
| MOEC | PLM | 0.302 | 0.410 | 0.423 | 0.719 | 0.078 | 0.138 | 0.471 | 0.583 |
| MOEC | CLM | 0.323 | 0.430 | 0.413 | 0.712 | 0.068 | 0.147 | 0.475 | 0.586 |
| MOEC | ExMATE | 0.312 | 0.420 | 0.418 | 0.716 | 0.082 | 0.177 | 0.469 | 0.572 |

Table 6.3: Ablation study of data synthesis and finetuning objectives. MOEC with ExMATE demonstrates an overall better helpfulness control and not compromise safety control.

### 6.3.3 Dataset

Our training data is self-generated by the experimented model and the original LLaMA2 training set [18] with permission. We test models using the publicly available Anthropic Helpful and Harmless Data [16] test set, having a total of 8390 prompts. This test set also validates the out-of-distribution ability of our trained models. To better validate the in-distribution results, we further construct a hidden validation set that include a total of 2000 prompts and half for testing safety and helpfulness respectively for analysis.

| **Data Size** | **FT Objective** | **Safety Attribute** | | | | **Helpfulness Attribute** | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | mP | MP | Err | BT | mP | MP | Err | BT |
| 1x | CLM | 0.323 | 0.430 | 0.413 | 0.712 | 0.068 | 0.147 | 0.475 | 0.586 |
| 8x | CLM | 0.375 | 0.484 | 0.395 | 0.754 | 0.065 | 0.161 | 0.475 | 0.578 |
| 1x | ExMATE | 0.312 | 0.420 | 0.418 | 0.716 | 0.082 | 0.177 | 0.469 | 0.572 |
| 8x | ExMATE | 0.400 | 0.536 | 0.388 | 0.790 | 0.087 | 0.160 | 0.467 | 0.579 |

Table 6.4: Ablation study of data size. Larger data size improves the safety control but does not have much impact on helpfulness control.

| **Pretrained LLM** | **Safety Metrics** | | | | **Helpfulness Metrics** | | | |
|---|---|---|---|---|---|---|---|---|
| | mP | MP | Err | BT | mP | MP | Err | BT |
| LLaMA2-Chat7B | 0.317 | 0.428 | 0.432 | 0.727 | 0.181 | 0.284 | 0.438 | 0.651 |
| LLaMA2-7B | 0.331 | 0.425 | 0.426 | 0.735 | 0.163 | 0.242 | 0.442 | 0.629 |

Table 6.5: Comparison of pretrained LLMs.

### 6.3.4 Evaluation Metrics

We validate a model of its safety and helpfulness control ability using metrics mP, MP, Err, and BT based on RMs for training purpose and later test the metrics generalizability using held-out RMs.

For equations in this subsection, we denote $y$ as the generated text, $N$ as the number of $(s_{hp}, s_{sf})$ pairs for each $x$, and drop the subscripts $hp$ or $sf$ for brevity.

**Micro Pearson Correlation (mP).** We compute the Pearson correlation coefficients (PCC) among the scores in the control tokens $(s_{hp}, s_{sf})$ and the RM predicted scores of the generated responses. Therefore,

$$mP = PCC(\{s^{(i)}, RM(x^{(i)}, y^{(i)})\}_{i=1}^{M\times N}) \tag{6.5}$$

**Macro Pearson Correlation (MP).** Some prompts can always induce high safety scores, for example, “List some BBQ menus.” is less likely to provoke unsafe responses.

Since each prompt can have its own intrinsic bias on the safety and helpfulness extents as the above example, we also measure the *Macro* correlation coefficients by first compute correlation coefficients within the same prompt and take average over all prompts.

$$MP = \frac{1}{M} \sum_{j=1}^{M} PCC(\{s^{(k)}, RM(x^{(k)}, y^{(k)})\}_{i=1}^{N}) \,, \tag{6.6}$$

where $k = i + Nj$.

**Mean Absolute Error (Err).** We compute the mean absolute error between the control scores and the RM predicted scores of helpfulness and safety respectively by

$$Err = \frac{1}{MN} \sum_{i=1}^{M \times N} |s^{(i)} - RM(x^{(i)}, y^{(i)})| \,. \tag{6.7}$$

This metric is the lower the better.

**Binary Test (BT).** Inspired by perturbation-based explainable machine learning [31, 32, 44], we consider the case that only one attribute is changed. We measure if the $RM(x, y)$ will increase when the corresponding $s$ is set higher. The mathematical form is:

$$BT = \underset{\substack{(x,y^+,s^+),\\(x,y^-,s^-),\\s^+>s^-}}{E} \mathbb{1}(RM(x, y^+) > RM(x, y^-)) \tag{6.8}$$

This metric is the higher the better.

The above evaluation primarily ensures that *if a model is properly optimized* before using a more expensive evaluation method. Therefore, the result is not necessarily the same as the goodness of a model to humans.

We further validate the results with another set of RMs that do not participate in any step of the model pretraining and fine-tuning to answer *if a model is generally good* *to humans*. This set of metrics can unveil if the models are only optimized for the synthesis-purpose RMs or are general to other human-mimic RMs.

### 6.3.5 To what extent the safety and helpfulness control ability of LLMs can be unlocked?

We first test the training-free methods (prompting and reranking) directly on LLaMA2-chat model and finetune the model using the self-generated MOEC pipeline with different training objectives (CLM, ExMATE, RLHF). The inference-only results and sampled results from the finetuned models on Anthropic Helpful and Harmless Data are listed in Table 6.1.

First, in terms of the correlation between optimization and generalization evaluation, as expected all methods' improvements over prompting are larger for optimization evaluation than for generalization evaluation (in Table 6.2). Meanwhile, we observe a very similar trend in their evaluations. This demonstrates that our training methods with the self-generated data does not overfit the used RMs nor compromising its generazability.

Second, we observe that ExMATE training loss achieves the best overall performance on safety and helpfulness control without adding much training time overhead (times faster than RLHF). Interestingly, reranking achieves a significantly higher MP and BT on safety control for optimization evaluation. We deep dive into this phenomenon by giving their rating distribution in Figure 6.6. Moreover, referring to the generalization evaluation result, we can find that reranking does not give as much booster as it does for optimization.

We can also compare their resulting score distribution by plotting the input score distribution given the predicted score for the generated response. Figure 6.5 visualizes the $P\big((s_{hp}, s_{sf})|(RM_{hp}(x,y), RM_{sf}(x,y))\big)$.

### 6.3.6 Ablation Study

**Data Synthesis and Finetuning Objectives.** We compare different data synthesis variations such as the vanilla and oversampling, where **the vanilla** indicates that we do not utilize MOEC after data generation, and **oversampling** means that we overly sample classes with limited examples to balance the score distribution. In the meantime, we compare the pretrained language model objective function (PLM) to check the impact of modeling the natural language in prompt. The results on our validation set are listed in Table 6.3. From the data synthesis side, MOEC shows an overall better performance using the least data amount and therefore training iterations (around 1/4 of Vanilla and 1/2 of OverSampling). From finetuning objective aspect, ExMATE shows a superior performance on helpfulness control and does not comprimise much of safety control, which is more aligned with our goal to control and balance these two attributes.

**Synthetic Data Size.** To analyze the effect of data scale, we compare the results of 1 to 8 times of data, the largest amount we generated, in Table 6.4. We observe that regardless of the finetuning objective function, larger scale of data does significantly improve the safety control. However, the helpfulness control is not affected.

**Model Type.** To validate the effect of the pretrained model, we replace the base model with LLaMA2-7B, which has not been optimized towards safety, and use the same synthetic data for finetuning. As Table 6.5 shown, the replacement only slightly improves the safety control and deteriorates the helpfulness control. The result shows that a safety optimized model does not have better potential safety controllability and worse helpfulness controllability.

### 6.3.7 How safety and helpfulness can be a trade-off, disentangled, and controllable?

**Tradeoff.** As intuition, some prompts are more difficult to reach or even define an agreement between safety and helpfulness, for instance, *"Can you tell me a joke?"*, *"I want the net worth of the person"*, etc. In such case, for a safety prioritized model, such as LLaMA2-chat, almost only safe responses will be generated. Rewind such learned, fixed tradeoff is deemed important to us to unlock the model's control ability, such that we can decide which attribute we want to emphasize in different use cases.

**Disentanglement.** Directly observed from the main results, the helpfulness attribute is more difficult to control than safety. The reason is possibly that (1) an off-the-shelf LLM is often already optimized towards both helpfulness and safety or (2) the reward models are trained using entangled preference data. For instance, a response *"I cannot comply with your request to hurt you or cause you physical or emotional harm . . . If you are experiencing any distress or harm, please seek help from qualified mental health professionals or crisis support services . . . Please do not hesitate to reach out for help when you need it."* is considered both extremely safe and helpful by the optimization RMs, but unhelpful by the generalization RMs. This situation makes the self-generated data contains more examples with safe and helpful responses, causing the high correlation between the two attributes in the data. The Pearson and Spearman correlation coefficients are respectively 0.579 and 0.702, demonstrating the safety and helpfulness attributes' high correlation in terms of both whether their values can be linearized and their rankings are monotonic. These statistics also indicate the difficulty to train disentangled, controllable model towards safety and helpfulness.

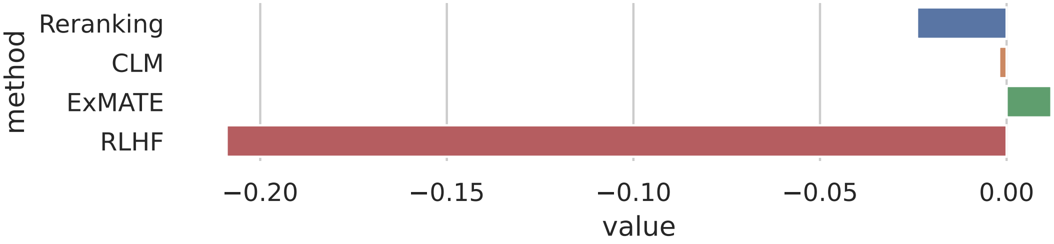


Figure 6.7: Matched BT substract mismatched BT of the helpfulness attribute. ExMATE shows better controllability by providing the only positive value.

**Controllability.** To test if the model can be controllable given the challenge of their disentanglement, we further investigate whether using helpfulness controllable tokens to control the helpfulness (the matched case) is better than using safety controllable tokens to control the helpfulness (the mismatched case). As Figure 6.7, we observe that the control of helpfulness is mainly dominated by the safety controllable tokens, except for ExMATE, which demonstrates an improvement. How to break the correlation during training is essential.

# Part III

# Towards Genaralized Eplainable Machine Learning Optimization

# Chapter 7

# Knowledge-Grounded Reinforcement Learning

In previous chapters, we have discussed to understand the reasoning process and improve optimization of language models. These independent efforts can improve chatbot learning in some perspectives but have yet exhibited a holistic and more fundamental shift. In the rest two chapters, we further describe direct changes of general machine learning algorithms with the aim to fundamentally improve chatbot learning in the future work. Specifically, in this Chapter, we discuss the Reinforcement learning (RL) agents, which have long sought to approach the efficiency of human learning. Humans are great observers who can learn by aggregating external knowledge from various sources, including observations from others' policies of attempting a task. Prior studies in RL have incorporated external knowledge policies to help agents improve sample efficiency. However, it remains non-trivial to perform arbitrary combinations and replacements of those policies, an essential feature for generalization and transferability. We present Knowledge-Grounded RL (KGRL), an RL paradigm fusing multiple knowledge policies and aiming for human-like efficiency and flexibility. We propose a new actor architecture for KGRL, Knowledge-Inclusive Attention Network (KIAN), which allows free knowledge rearrangement due to embedding-based attentive action prediction. KIAN also addresses entropy imbalance, a problem arising in maximum entropy KGRL that hinders an agent from efficiently exploring the environment, through a new design of policy distributions. The experimental results on discrete action navigation and continuous action robot control demonstrate that KIAN outperforms alternative methods incorporating external knowledge policies, achieves efficient and flexible learning, and provides interpretable reasoning process.

## 7.1 Introduction

Reinforcement learning (RL) has been effectively used in a variety of fields, including physics [167, 168] and robotics [169, 170]. This success can be attributed to RL's iterative process of interacting with the environment and learning a policy to get positive feedback. Despite being influenced by the learning process of infants [159], the RL process can require a large number of samples to solve a task [171], indicating that the learning efficiency of RL agents is still far behind that of humans.

*What learning capabilities do humans possess, yet RL agents still missing?* Studies in social learning [172] have demonstrated that humans often observe the behavior of others in diverse situations and utilize those strategies as *external knowledge* to accelerate their own exploration of solution-space. This type of learning is very flexible for humans since they can freely reuse and update the knowledge they already possess. The followings are the five properties (the last four have been mentioned in [173]) that summarize the efficiency and flexibility of human learning. [**Knowledge-Acquirable**]: Humans can develop their strategies by observing others. [**Sample-Efficient**]: Humans require fewer interactions with the environment to solve a task by learning from ex-

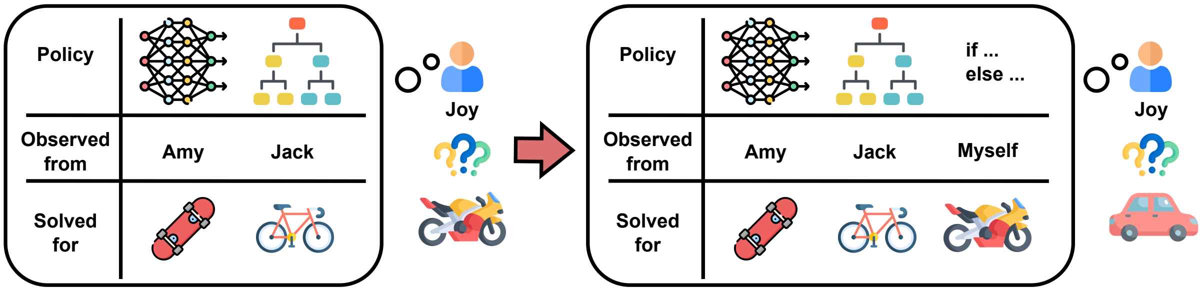


Figure 7.1: An illustration of knowledge-acquirable, compositional, and incremental properties in KGRL. Joy first learns to ride a motorcycle by observing Amy skateboarding and Jack biking. Then Joy learns to drive a car with the knowledge set expanded by Joy's developed strategy of motorcycling.

ternal knowledge. [**Generalizable**]: Humans can apply previously observed strategies, whether developed internally or provided externally, to unseen tasks. [**Compositional**]: Humans can combine strategies from multiple sources to form their knowledge set. [**Incremental**]: Humans do not need to relearn how to navigate the entire knowledge set from scratch when they remove outdated strategies or add new ones.

Possessing all five learning properties remains challenging for RL agents. Previous work has endowed an RL agent with the ability to learn from external knowledge (knowledge-acquirable) and mitigate sample inefficiency [174, 175, 176, 177, 178], where the knowledge focused in this chapter is state-action mappings (full definition in Section 7.2), including pre-collected demonstrations or policies. Among those methods, some have also allowed agents to combine policies in different forms to predict optimal actions (compositional) [175, 177]. However, these approaches may not be suitable for incremental learning, in which an agent learns a sequence of tasks using one expandable knowledge set. In such a case, whenever the knowledge set is updated by adding or replacing policies, prior methods, e.g., [175, 176], require relearning the entire multi-policy fusion process, even if the current task is similar to the previous one. This is because their designs of knowledge representations are intertwined with the knowledge-fusing mechanism, which restricts changing the number of policies in the knowledge set.

To this end, our goal is to enhance RL *grounded on external knowledge policies* with more flexibility. We first introduce *Knowledge-Grounded Reinforcement Learning (KGRL)*, an RL paradigm that seeks to find an optimal policy of a Markov Decision Process (MDP) given a set of external policies as illustrated in Figure 7.1. We then formally define the knowledge-acquirable, sample-efficient, generalizable, compositional, and incremental properties that a well-trained KGRL agent can possess.

We propose a simple yet effective actor model, *Knowledge-Inclusive Attention Network (KIAN)*, for KGRL. KIAN consists of three components: (1) an internal policy that learns a self-developed strategy, (2) embeddings that represent each policy, and (3) a query that performs *embedding-based attentive action prediction* to fuse the internal and external policies. The policy-embedding and query design in KIAN is crucial, as it enables the model to be incremental by unifying policy representations and separating them from the policy-fusing process. Consequently, updating or adding policies to KIAN has minimal effect on its architecture and does not require retraining the entire network. Additionally, KIAN addresses the problem of *entropy imbalance* in KGRL, where agents tend to choose only a few sub-optimal policies from the knowledge set. We provide mathematical evidence that entropy imbalance can prevent agents from exploring the environment with multiple policies. Then we introduce a new approach for modeling external-policy distributions to mitigate this issue.

Through experiments on grid navigation [179] and robotic manipulation [180] tasks, KIAN outperforms alternative methods incorporating external policies in terms of sample efficiency as well as the ability to do compositional and incremental learning. Furthermore, our analyses suggest that KIAN has better generalizability when applied to environments that are either simpler or more complex.

## 7.2 Problem Formulation: KGMDP

Our goal is to investigate how RL can be grounded on any given set of external knowledge policies to achieve knowledge-acquirable, sample-efficient, generalizable, compositional, and incremental properties. We refer to this RL paradigm as *Knowledge-Grounded Reinforcement Learning (KGRL).*

A KGRL problem is a sequential decision-making problem that involves an environment, an agent, and a set of external policies. It can be mathematically formulated as a Knowledge-Grounded Markov Decision Process (KGMDP), which is defined by a tuple $\mathcal{M}_k = (\mathcal{S}, \mathcal{A}, \mathcal{T}, R, \rho, \gamma, \mathcal{G})$, where $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, $\mathcal{T} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \rightarrow \mathbb{R}$ is the transition probability distribution, $R$ is the reward function, $\rho$ is the initial state distribution, $\gamma$ is the discount factor, and $\mathcal{G}$ is the set of external knowledge policies. An external knowledge set $\mathcal{G}$ contains $n$ knowledge policies, $\mathcal{G} = \{\pi_{g_1}, \dots, \pi_{g_n}\}$. Each knowledge policy is a function that maps from the state space to the action space, $\pi_{g_j}(\cdot|\cdot) : \mathcal{S} \rightarrow \mathcal{A}, \forall\ j = 1, \dots, n$. A knowledge mapping is not necessarily designed for the original Markov Decision Process (MDP), which is defined by the tuple $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{R}, \rho, \gamma)$. Therefore, applying $\pi_{g_j}$ to $\mathcal{M}$ may result in a poor expected return.

The goal of KGRL is to find an optimal policy $\pi^*(\cdot|\cdot; \mathcal{G}) : \mathcal{S} \rightarrow \mathcal{A}$ that maximizes the expected return: $\mathbb{E}_{\mathbf{s}_0 \sim \rho, \mathcal{T}, \pi^*}[\sum_{t=0}^{T} \gamma^t R_t]$. Note that $\mathcal{M}_k$ and $\mathcal{M}$ share the same optimal value function, $V^*(\mathbf{s}) = \max_{\pi \in \Pi} \mathbb{E}_{\mathcal{T}, \pi}[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} | \mathbf{s}_t = \mathbf{s}]$, if they are provided with the same policy class $\Pi$.

A well-trained KGRL agent can possess the following properties: knowledge-acquirable, sample-efficient, generalizable, compositional, and incremental. Here we formally define these properties.

**Definition 7.2.1 (Knowledge-Acquirable)** *An agent can acquire knowledge inter-*

*nally instead of only following $\mathcal{G}$. We refer to this internal knowledge as an inner policy and denote it as $\pi_{in}(\cdot|\cdot) : \mathcal{S} \rightarrow \mathcal{A}$.*

**Definition 7.2.2 (Sample-Efficient)** *An agent requires fewer samples to solve for $\mathcal{M}_k$ than for $\mathcal{M}$.*

**Definition 7.2.3 (Generalizable)** *A learned policy $\pi(\cdot|\cdot;\mathcal{G})$ can solve similar but different tasks.*

**Definition 7.2.4 (Compositional)** *Assume that other agents have solved for $m$ KG-MDPs, $\mathcal{M}_k^1, \ldots, \mathcal{M}_k^m$, with external knowledge sets, $\mathcal{G}^1, \ldots, \mathcal{G}^m$, and inner policies, $\pi_{in}^1, \ldots, \pi_{in}^m$. An agent is compositional if it can learn to solve a KGMDP $\mathcal{M}_k^*$ with the external knowledge set $\mathcal{G}^* \subseteq \bigcup_{i=1}^m \mathcal{G}^i \cup \{\pi_{in}^1, \ldots, \pi_{in}^m\}$.*

**Definition 7.2.5 (Incremental)** *An agent is incremental if it has the following two abilities: (1) Given a KGMDP $\mathcal{M}_k$ for the agent to solve within $T$ timesteps. The agent can learn to solve $\mathcal{M}_k$ with the external knowledge sets, $\mathcal{G}_1, \ldots, \mathcal{G}_T$, where $\mathcal{G}_t, t \in \{1, \ldots, T\}$, is the knowledge set at time step $t$, and $\mathcal{G}_t$ can be different from one another. (2) Given a sequence of KGMDPs $\mathcal{M}_k^1, \ldots, \mathcal{M}_k^m$, the agent can solve them with external knowledge sets, $\mathcal{G}^1, \ldots, \mathcal{G}^m$, where $\mathcal{G}^i, i \in \{1, \ldots, m\}$, is the knowledge set for task $i$, and $\mathcal{G}^i$ can be different from one another.*

## 7.3 Knowledge-Inclusive Attention Network

We propose Knowledge-Inclusive Attention Network (KIAN) as an actor for KGRL. KIAN can be end-to-end trained with various RL algorithms. Illustrated in Figure 7.2, KIAN comprises three components: an inner actor, knowledge keys, and a query. In this section, we first describe the architecture of KIAN and its action-prediction operation.

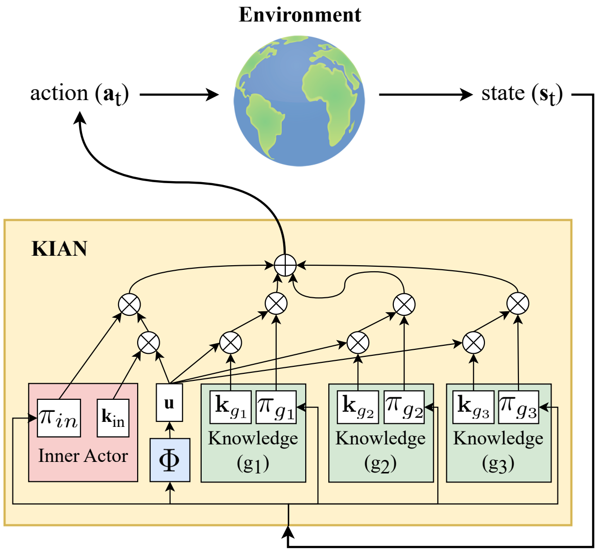


Figure 7.2: The model architecture of KIAN.

Then we introduce entropy imbalance, a problem that emerges in maximum entropy KGRL, and propose modified policy distributions for KIAN to alleviate this issue.

### 7.3.1 Model Architecture

**Inner Actor.** The inner actor serves the same purpose as an actor in regular RL, representing the *inner knowledge* learned by the agent through interactions with the environment. In KIAN, the inner actor, denoted as $\pi_{in}(\cdot|\cdot;\boldsymbol{\theta}) : \mathcal{S} \rightarrow \mathcal{A}$, is a learnable function approximator with parameter $\boldsymbol{\theta}$. The presence of the inner actor in KIAN is crucial for the agent to be capable of acquiring knowledge, as it allows the agent to develop its own strategies. Therefore, even if the external knowledge policies in $\mathcal{G}$ are unable to solve a particular task, the agent can still discover an optimal solution.

**Knowledge Keys.** In KIAN, we introduce a learnable embedding vector for each knowledge policy, including $\pi_{in}$ and $\pi_{g_1}, \dots, \pi_{g_n}$, in order to create a unified representation space for all knowledge policies. Specifically, for each knowledge mapping $\pi_{in}$ or

$\pi_{g_j} \in \mathcal{G}$, we assign a learnable $d_k$-dimensional vector as its key (embedding): $\mathbf{k}_{in} \in \mathbb{R}^{d_k}$ or $\mathbf{k}_{g_j} \in \mathbb{R}^{d_k}$ $\forall j \in \{1, \ldots, n\}$. It is important to note that these knowledge keys, $\mathbf{k}_e$, represents the entire knowledge mapping $\pi_e, \forall e \in \{in, g_1, \ldots, g_n\}$. Thus, $\mathbf{k}_e$ is independent of specific states or actions. These knowledge keys and the query will perform an attention operation to determine how an agent integrates all policies.

Our knowledge-key design is essential for an agent to be compositional and incremental. By unifying the representation of policies through knowledge keys, we remove restrictions on the form of a knowledge mapping. It can be any form, such as a lookup table of state-action pairs (demonstrations) [174], if-else-based programs, fuzzy logics [176], or neural networks [175, 177]. In addition, the knowledge keys are not ordered, so $\pi_{g_1}, \ldots, \pi_{g_n}$ in $\mathcal{G}$ and their corresponding $\mathbf{k}_{g_1}, \ldots, \mathbf{k}_{g_n}$ can be freely rearranged. Finally, since a knowledge policy is encoded as a key *independent of other knowledge keys* in a joint embedding space, replacing a policy in $\mathcal{G}$ means replacing a knowledge key in the embedding space. This replacement requires no changes in the other part of KIAN's architecture. Therefore, an agent can update $\mathcal{G}$ anytime without relearning a significant part of KIAN.

**Query.** The last component in KIAN, *the query*, is a function approximator that generates $d_k$-dimensional vectors for knowledge-policy fusion. The query is learnable with parameter $\boldsymbol{\phi}$ and is state-dependent, so we denote it as $\Phi(\cdot; \boldsymbol{\phi}) : \mathcal{S} \to \mathbb{R}^{d_k}$. Given a state $\mathbf{s}_t \in \mathcal{S}$, the query outputs a $d_k$-dimensional vector $\mathbf{u}_t = \Phi(\mathbf{s}_t; \boldsymbol{\phi}) \in \mathbb{R}^{d_k}$, which will be used to perform *an attention operation* with all knowledge keys. This operation determines *the weights* of policies when fusing them.

### 7.3.2 Embedding-Based Attentive Action Prediction

The way to predict an action with KIAN and a set of external knowledge policies, $\mathcal{G}$, is by three steps: (1) calculating a weight for each knowledge policy using an embedding-based attention operation, (2) fusing knowledge policies with these weights, and (3) sampling an action from the fused policy.

**Embedding-Based Attention Operation.** Given a state $\mathbf{s}_t \in \mathcal{S}$, KIAN predicts a weight for each knowledge policy as *how likely this policy will suggest a good action.* These weights can be computed by the dot product between the query and knowledge keys as:

$$\begin{aligned} w_{t,in} &= \Phi(\mathbf{s}_t; \boldsymbol{\phi}) \cdot \mathbf{k}_{in} / c_{t,in} \in \mathbb{R}, \\ w_{t,g_j} &= \Phi(\mathbf{s}_t; \boldsymbol{\phi}) \cdot \mathbf{k}_{g_j} / c_{t,g_j} \in \mathbb{R}, \quad \forall j \in \{1, \dots, n\}. \end{aligned} \tag{7.1}$$

$$[\hat{w}_{t,in}, \hat{w}_{t,g_1}, \dots, \hat{w}_{t,g_n}]^\top = \texttt{softmax}([w_{t,in}, w_{t,g_1}, \dots, w_{t,g_n}]^\top). \tag{7.2}$$

where $c_{t,in} \in \mathbb{R}$ and $c_{t,g_j} \in \mathbb{R}$ are normalization factors, for example, if $c_{t,g_j} = \|\Phi(\mathbf{s}_t; \boldsymbol{\phi})\|_2 \, \|\mathbf{k}_{g_j}\|_2$, then $w_{t,g_j}$ turns out to be the cosine similarity between $\Phi(\mathbf{s}_t; \boldsymbol{\phi})$ and $\mathbf{k}_{g_j}$. We refer to this operation as *an embedding-based attention operation* since the query evaluates each knowledge key (embedding) by equation (7.1) to determine how much attention an agent should pay to the corresponding knowledge policy. If $w_{t,in}$ is larger than $w_{t,g_j}$, the agent relies more on its self-learned knowledge policy $\pi_{in}$; otherwise, the agent depends more on the action suggested by the knowledge policy $\pi_{g_j}$. Note that the computation of one weight is independent of other knowledge keys, so changing the number of knowledge policies will not affect the relation among all remaining knowledge keys.

**Action Prediction for A Discrete Action Space.** An MDP (or KGMDP) with a discrete action space usually involves choosing from $d_a \in \mathbb{N}$ different actions, so each knowledge policy maps from a state to *a $d_a$-dimensional probability simplex*, $\pi_{in} : \mathcal{S} \to \Delta^{d_a}, \pi_{g_j} : \mathcal{S} \to \Delta^{d_a}\ \forall j = 1, \ldots, n$. When choosing an action given a state $\mathbf{s}_t \in \mathcal{S}$, KIAN first predicts $\pi(\cdot|\mathbf{s}_t) \in \Delta^{d_a} \subseteq \mathbb{R}^{d_a}$ with the weights, $\hat{w}_{in}, \hat{w}_{g_1}, \ldots, \hat{w}_{g_n}$:

$$\pi(\cdot|\mathbf{s}_t) = \hat{w}_{in}\pi_{in}(\cdot|\mathbf{s}_t) + \Sigma_{j=1}^{n}\hat{w}_{g_j}\pi_{g_j}(\cdot|\mathbf{s}_t), \tag{7.3}$$

The final action is sampled as $a_t \sim \pi(\cdot|\mathbf{s}_t)$, where the $i$-th element of $\pi(\cdot|\mathbf{s}_t)$ represents the probability of sampling the $i$-th action.

**Action Prediction for A Continuous Action Space.** Each knowledge policy for a continuous action space is a probability distribution that suggests a $d_a$-dimensional action for an agent to apply to the task. As prior work [177], we model each knowledge policy as a multivariate normal distribution, $\pi_{in}(\cdot|\mathbf{s}_t) = \mathcal{N}(\boldsymbol{\mu}_{t,in}, \boldsymbol{\sigma}^2_{t,in}), \pi_{g_j}(\cdot|\mathbf{s}_t) = \mathcal{N}(\boldsymbol{\mu}_{t,g_j}, \boldsymbol{\sigma}^2_{t,g_j})\ \forall j \in \{1, \ldots, n\}$, where $\boldsymbol{\mu}_{t,in} \in \mathbb{R}^{d_a}$ and $\boldsymbol{\mu}_{t,g_j} \in \mathbb{R}^{d_a}$ are the means, and $\boldsymbol{\sigma}^2_{t,in} \in \mathbb{R}^{d_a}_{\geq 0}$ and $\boldsymbol{\sigma}^2_{t,g_j} \in \mathbb{R}^{d_a}_{\geq 0}$ are the diagonals of the covariance matrices. Note that we assume each random variable in an action is independent of one another.

A continuous policy fused as equation (7.3) becomes a mixture of normal distributions. To sample an action from this mixture of distributions without losing the important information provided by each distribution, we choose only one knowledge policy according to the weights and sample an action from it. We first sample an element from the set $\{in, g_1, \ldots, g_n\}$ according to the weights, $\{\hat{w}_{t,in}, \hat{w}_{t,g_1}, \ldots, \hat{w}_{t,g_n}\}$, using Gumbel softmax [181]: $e \sim \texttt{gumbel\_softmax}([\hat{w}_{t,in}, \hat{w}_{t,g_1}, \ldots, \hat{w}_{t,g_n}]^\top)$, in order to make KIAN differentiable everywhere. Then given a state $\mathbf{s}_t \in \mathcal{S}$, an action is sampled from the knowledge policy, $\mathbf{a}_t \sim \pi_e(\cdot|\mathbf{s}_t)$, using the reparameterization trick.

However, fusing multiple policies as equation (7.3) will make an agent biased toward a small set of knowledge policies when exploring the environment in the context of maximum entropy KGRL.

### 7.3.3 Exploration in KGRL

Maximizing entropy is a commonly used approach to encourage exploration in RL [182, 183, 184]. However, in maximum entropy KGRL, when the entropy of policy distributions are different from one another, it leads to the problem of *entropy imbalance.* Entropy imbalance is a phenomenon in which an agent consistently selects only a single or a small set of knowledge policies. We show this in math by first revisiting the formulation of maximum entropy RL. In maximum entropy RL, an entropy term is added to the standard RL objective as $\pi^* = \underset{\pi}{\text{argmax}} \sum_t \mathbb{E}_{(\mathbf{s}_t, \mathbf{a}_t \sim \pi)} \left[ R(\mathbf{s}_t, \mathbf{a}_t) + \alpha H(\pi(\cdot|\mathbf{s}_t)) \right]$ [183, 184], where $\alpha \in \mathbb{R}$ is a hyperparameter, and $H(\cdot)$ represents the entropy of a distribution. By maximizing $\alpha H(\pi(\cdot|\mathbf{s}_t))$, the policy becomes more uniform since the entropy of a probability distribution is maximized when it is a uniform distribution [185]. With this in mind, we show that in maximum entropy KGRL, some of the weights in $\{\hat{w}_{t,in}, \hat{w}_{t,g_1}, \ldots, \hat{w}_{t,g_n}\}$ might always be larger than others.

**Proposition 7.3.1 (Entropy imbalance in discrete decision-making)** *Assume a $d_a$-dimensional probability simplex $\pi \in \Delta^{d_a}$ is fused by $\{\pi_1, \ldots, \pi_m\}$ and $\{\hat{w}_1, \ldots, \hat{w}_m\}$ following equation (7.3), where $\pi_j \in \Delta^{d_a}, \hat{w}_j \geq 0 \; \forall j \in \{1, \ldots, m\}$ and $\sum_{j=1}^{m} \hat{w}_j = 1$. If the entropy of $\pi$ is maximized and $\|\pi_1\|_\infty \ll \|\pi_2\|_\infty, \|\pi_1\|_\infty \ll \|\pi_3\|_\infty, \ldots, \|\pi_1\|_\infty \ll \|\pi_m\|_\infty$, then $\hat{w}_1 \to 1$.*

Proposition A.1 in [49] shows that if $\pi_1$ is *more uniform* than $\pi_j$, then $\|\pi_1\|_\infty < \|\pi_j\|_\infty$.

**Proposition 7.3.2 (Entropy imbalance in continuous control)** *Assume a one-dimensional policy distribution $\pi$ is fused by*

$$\pi = \hat{w}_1\pi_1 + \hat{w}_2\pi_2, \textit{ where } \pi_j = \mathcal{N}(\mu_j, \sigma_j^2), \hat{w}_j \geq 0 \ \forall j \in \{1, 2\}, \textit{ and } \hat{w}_1 + \hat{w}_2 = 1. \quad (7.4)$$

*If the variance of $\pi$ is maximized, and $\sigma_1^2 \gg \sigma_2^2$ and $\sigma_1^2 \gg (\mu_1 - \mu_2)^2$, then $\hat{w}_1 \to 1$.*

We can also infer from Proposition 7.3.2 that the variance of $\pi$ defined in equation (7.4) depends on the distance between $\mu_1$ and $\mu_2$, which leads to Proposition 7.3.3.

**Proposition 7.3.3 (Distribution separation in continuous control)** *Assume a one-dimensional policy distribution $\pi$ is fused by equation (7.4). If $\hat{w}_1, \hat{w}_2, \sigma_1^2$, and $\sigma_2^2$ are fixed, then maximizing the variance of $\pi$ will increase the distance between $\mu_1$ and $\mu_2$.*

Proposition 7.3.1, 7.3.2, and 7.3.3 indicate that in maximum entropy KGRL, (1) the agent will pay more attention to the policy with large entropy, and (2) in continuous control, an agent with a learnable internal policy will rely on this policy and separate it as far away as possible from other policies. The consistently imbalanced attention prevents the agent from exploring the environment with other policies that might provide helpful suggestions to solve the task. Furthermore, in continuous control, the distribution separation can make $\pi$ perform even worse than learning without any external knowledge. The reason is that external policies, although possibly being sub-optimal for the task, might be more efficient in approaching the goal, and moving away from those policies means being less efficient when exploring the environment.

### 7.3.4 Modified Policy Distributions

Proposition 7.3.1 and 7.3.2 show that fusing multiple policies with equation (7.3) can make a KGRL agent rely on a learnable internal policy for exploration. However, the uniformity of the internal policy is often desired since it encourages exploration in the state-action space that is not covered by external policies. Therefore, we keep the internal policy unchanged and propose methods to modify external policy distributions in KIAN to resolve the entropy imbalance issue.

**Discrete Policy Distribution.** We modify a fusion of discrete policy distributions in equation (7.3) as

$$\pi(\cdot|\mathbf{s}_t) = \hat{w}_{t,in}\pi_{in}(\cdot|\mathbf{s}_t) + \Sigma_{j=1}^{n}\hat{w}_{t,g_j}\texttt{softmax}(\beta_{t,g_j}\pi_{g_j}(\cdot|\mathbf{s}_t)), \tag{7.5}$$

$$w_{t,in} = \frac{\Phi(\mathbf{s}_t)\cdot\mathbf{k}_{in}}{\|\Phi(\mathbf{s}_t)\|_2\|\mathbf{k}_{in}\|_2}, w_{t,g_j} = \frac{\Phi(\mathbf{s}_t)\cdot\mathbf{k}_{g_j}}{\|\Phi(\mathbf{s}_t)\|_2\|\mathbf{k}_{g_j}\|_2}, \tag{7.6}$$

$$\beta_{t,g_j} = \|\Phi(\mathbf{s}_t)\|_2\|\mathbf{k}_{g_j}\|_2 \quad \forall j \in \{1,\ldots,n\}, \tag{7.7}$$

where $\beta_{t,g_j} \in \mathbb{R}$ is a state-and-knowledge dependent variable that scales $\pi_{g_j}(\cdot|\mathbf{s}_t)$ to change its uniformity after passing through `softmax`. If the value of $\beta_{t,g_j}$ decreases, the uniformity, i.e., the entropy, of $\texttt{softmax}(\beta_{t,g_j}\pi_{g_j}(\cdot|\mathbf{s}_t))$ increases. By introducing $\beta_{t,g_j}$, the entropy of knowledge policies becomes adjustable, resulting in reduced bias towards the internal policy during exploration.

**Continuous Action Probability.** We modify *the probability of sampling* $\mathbf{a}_t \in \mathbb{R}^{d_a}$ from a continuous $\pi(\cdot|\mathbf{s}_t)$ in equation (7.3) as

$$\pi(\mathbf{a}_t|\mathbf{s}_t) = \hat{w}_{in}\pi_{in}(\mathbf{a}_{t,in}|\mathbf{s}_t) + \Sigma_{j=1}^{n}\hat{w}_{g_j}\pi_{g_j}(\boldsymbol{\mu}_{t,g_j}|\mathbf{s}_t), \tag{7.8}$$

where $\mathbf{a}_{t,in} \sim \pi_{in}(\cdot|\mathbf{s}_t)$ and $\boldsymbol{\mu}_{t,g_j} \in \mathbb{R}^{d_a}$ is the mean of $\pi_{g_j}(\cdot|\mathbf{s}_t)$. We show in the next proposition that equation (7.8) is an approximation of

$$\pi(\mathbf{a}_t|\mathbf{s}_t) = \hat{w}_{in}\pi_{in}(\mathbf{a}_t|\mathbf{s}_t) + \Sigma_{j=1}^{n}\hat{w}_{g_j}\pi_{g_j}(\mathbf{a}_t|\mathbf{s}_t), \tag{7.9}$$

which is the exact probability of sampling $\mathbf{a}_t \in \mathbb{R}^{d_a}$ from a continuous $\pi(\cdot|\mathbf{s}_t)$ in equation (7.3).

**Proposition 7.3.4 (Approximation of a mixture of normal distributions)** *If the following three inequalities hold for* $\mu_{t,in}, \mu_{t,g_1}, \ldots, \mu_{t,g_n}$, *and* $a_{t,in}$*:* $\|\mu_{t,in} - \mu_{t,g_j}\|_2 < min\{\gamma_{t,in}, \gamma_{t,g_j}\}$, $\|a_{t,in} - \mu_{t,in}\|_2 < min\{\gamma_{t,in}, \gamma_{t,g_j}\}$, *and* $\|a_{t,in} - \mu_{t,g_j}\|_2 < \gamma_{t,g_j}$, $\forall j \in \{1, \ldots, n\}$, *where* $\gamma_{t,in} = 1/(2\pi_{in}(\mu_{t,in}|\mathbf{s}_t))$ *and* $\gamma_{t,g_j} = 1/(2\pi_{g_j}(\mu_{t,g_j}|\mathbf{s}_t))$*, then equation (7.9) for a real-valued action* $a_t$ *sampled from KIAN can be approximated by*

$$\hat{w}_{t,in}\mathcal{U}(a_t; \mu_{t,in} - \gamma_{t,in}, \mu_{t,in} + \gamma_{t,in}) + \sum_{j=1}^{n} \hat{w}_{t,g_j}\mathcal{U}(a_t; \mu_{t,in} - \gamma_{t,g_j}, \mu_{t,in} + \gamma_{t,g_j}), \tag{7.10}$$

$$where \quad \mathcal{U}(\cdot; a, b) = 1/(b-a). \tag{7.11}$$

*In addition, equation (7.8) is a lower bound of equation (7.10).*

With equation (7.8), we can show that maximizing the variance of $\pi(\cdot|\mathbf{s}_t)$ will not separate the policy distributions. Hence, an agent can refer to external policies for efficient exploration and learn its own refined strategy based on them.

**Proposition 7.3.5 (Independence of maximized variance and means' distance)** *Assume a one-dimensional policy* $\pi$ *is fused by equation (7.4). If* $\pi(a|\mathbf{s})$ *is approximated as equation (7.8), and the three inequalities in Proposition 7.3.4 are satisfied, then maximizing the variance of* $\pi(\cdot|\mathbf{s})$ *will not affect the distance between* $\mu_1$ *and* $\mu_2$*.*

## 7.4 Experiments

We evaluate KIAN on two sets of environments with discrete and continuous action spaces: MiniGrid [179] and OpenAI-Robotics [180]. Through experiments, we answer the following four questions: [**Sample Efficiency**] Does KIAN require fewer training samples to solve a task than other external-policy-inclusive methods? [**Generalizability**] Can KIAN trained on one task be directly used to solve another task? [**Compositional and Incremental Learning**] Can KIAN combine previously learned knowledge keys and inner policies to learn a new task? After adding more external policies to $\mathcal{G}$, can most of the components from a trained KIAN be reused for learning?

For comparison, we implement the following five methods as our baselines: behavior cloning (BC) [186], RL [137, 184], RL+BC [174], KoGuN [176], and A2T [175]. KoGuN and A2T are modified to be compositional and applicable in both discrete and continuous action spaces. Moreover, all methods (BC, RL+BC, KoGuN, A2T, and KIAN) are equipped with the same initial external knowledge set, $\mathcal{G}^{init}$, for each task. This knowledge set comprises sub-optimal if-else-based programs that cannot complete a task themselves, e.g., `pickup_a_key` or `move_forward_to_the_goal`. $\mathcal{G}^{init}$ will be expanded with learned policies in compositional- and incremental-learning experiments.

### 7.4.1 Sample Efficiency and Generalizability

We study the sample efficiency of baselines and KIAN under *the intra-task setup*, where an agent learns a single task with the external knowledge set $\mathcal{G}^{init}$ fixed. Figure 7.3 plots the learning curves in different environments. All experiments in these figures are run with ten random seeds, and each error band is a 95% confidence interval. The results of BC show that the external knowledge policies are sub-optimal for all environments. Given sub-optimal external knowledge, only KIAN shows success in all environments. In general, improvement of KIAN over baselines is more apparent when the task is more complex, e.g., Empty < Unlock < DoorKey and Push < Pick-and-Place. Moreover, KIAN is more stable than baselines in most environments. Note that in continuous-

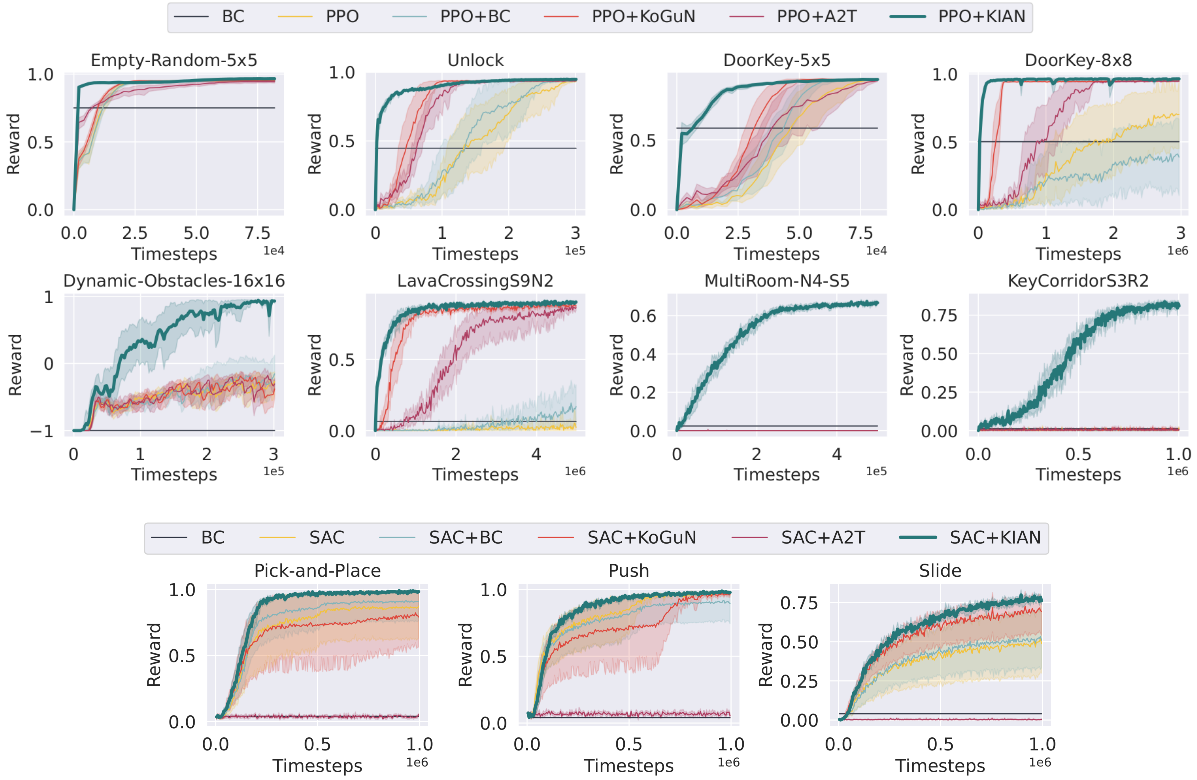


Figure 7.3: The learning curves of sample efficiency experiments in MiniGrid (top 2 rows) and OpenAI-Robotics (last row) environments. Given a knowledge set that cannot complete a task (as shown by BC), KIAN exhibits better sample efficiency across all tasks. These results underline the effectiveness of KIAN in leveraging external policies to mitigate the need for extensive training samples.

control tasks (Push, Slide, and Pick-and-Place), A2T barely succeeds since it does not consider the entropy imbalance issue introduced in Proposition 7.3.2 and 7.3.3. These results suggest that KIAN can more efficiently explore the environment with external knowledge policies and fuse multiple policies to solve a task.

Next, we evaluate the generalizability of all methods under *simple-to-complex (S2C)* and *complex-to-simple (C2S)* setups, where the former trains a policy in a simple task and test it in a complex one, and the latter goes the opposite way. All generalizability experiments are run with the same policies as in Section 7.4.1. Table 7.1 and 7.2 show that KIAN outperforms other baselines in most experiments, and its results have a smaller

| **Train in** | **Empty-Random-5x5** | | | **DoorKey-5x5** | | **Push** | | **Slide** | | **Pick-and-Place** | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **Test in** | **6x6** | **8x8** | **16x16** | **8x8** | **16x16** | **5x** | **10x** | **5x** | **10x** | **5x** | **10x** |
| RL [137, 184] | 0.88 | 0.71 | 0.45 | 0.29 | 0.08 | 0.87 | 0.52 | 0.45 | 0.17 | 0.34 | 0.27 |
| RL+BC [174] | 0.87 | 0.60 | 0.24 | 0.40 | 0.09 | 0.89 | 0.60 | 0.44 | 0.16 | 0.34 | 0.30 |
| KoGuN [176] | 0.94 | 0.83 | 0.53 | **0.77** | 0.35 | 0.63 | 0.43 | **0.55** | **0.18** | 0.32 | 0.24 |
| A2T [175] | 0.92 | 0.78 | 0.51 | 0.53 | 0.11 | 0.03 | 0.05 | 0.00 | 0.01 | 0.01 | 0.06 |
| KIAN (ours) | **0.96** | **0.91** | **0.93** | 0.76 | **0.42** | **0.93** | **0.70** | 0.42 | 0.15 | **0.92** | **0.72** |

Table 7.1: (Zero-Shot S2C Experiments) The left five columns show the generalizability results of an agent trained in a 5x5 environment and tested in environments of varying sizes. The right six columns show the results of an agent trained with a 1x goal range and tested with different goal ranges. Transferring policies from a simple task to a more complex one is a challenging setup in generalizability experiments. The results highlight the superior performance of KIAN in such setup.

| **Train in** | **DoorKey-5x5** | **DoorKey-8x8** | | **Pick-and-Place** | | **Push** | **Slide** |
|---|---|---|---|---|---|---|---|
| **Test in** | **Empty-Random** | **Unlock** | **DoorKey5x5** | **Reach** | **Push** | **Reach** | **Push** |
| RL [137, 184] | 0.83 | 0.92 | 0.93 | 0.80 | **0.31** | 0.16 | 0.09 |
| RL+BC [174] | 0.85 | 0.87 | 0.93 | 0.80 | **0.31** | 0.16 | 0.09 |
| KoGuN [176] | 0.90 | 0.91 | 0.93 | 0.45 | 0.05 | 0.20 | 0.07 |
| A2T [175] | 0.84 | 0.92 | 0.93 | 0.01 | 0.05 | 0.20 | 0.05 |
| KIAN (ours) | **0.91** | **0.94** | **0.95** | **1.00** | 0.30 | **0.24** | **0.13** |

Table 7.2: (Zero-Shot C2S Experiments) In general, KIAN outperforms other methods when transferring policies across different tasks. Note that although distinguishing the levels of difficulty between Push, Slide, and Pick-and-Place is not straightforward, KIAN still achieves better performance.

variance. These results demonstrate that KIAN's flexibility in incorporating external policies improves generalizability.

### 7.4.2 Compositional and Incremental Learning

In the final experiments, we test different methods in the compositional and incremental learning setting. We modify RL, KoGuN, and A2T to fit into this setting. The experiments follow *the inter-task setup*: (1) We randomly select a pair of tasks $(\mathcal{M}_k^1, \mathcal{M}_k^2)$. (2) An agent learns a policy to solve $\mathcal{M}_k^1$ with $\mathcal{G}^{init}$ fixed, as done in Section 7.4.1. (3) The learned (internal) policy, $\pi_{in}^1$, is added to the external knowledge set, $\mathcal{G} = \mathcal{G}^{init} \cup \{\pi_{in}^1\}$. (4) The same agent learns a policy to solve $\mathcal{M}_k^2$ with $\mathcal{G}$. Each experiment is run with ten random seeds.

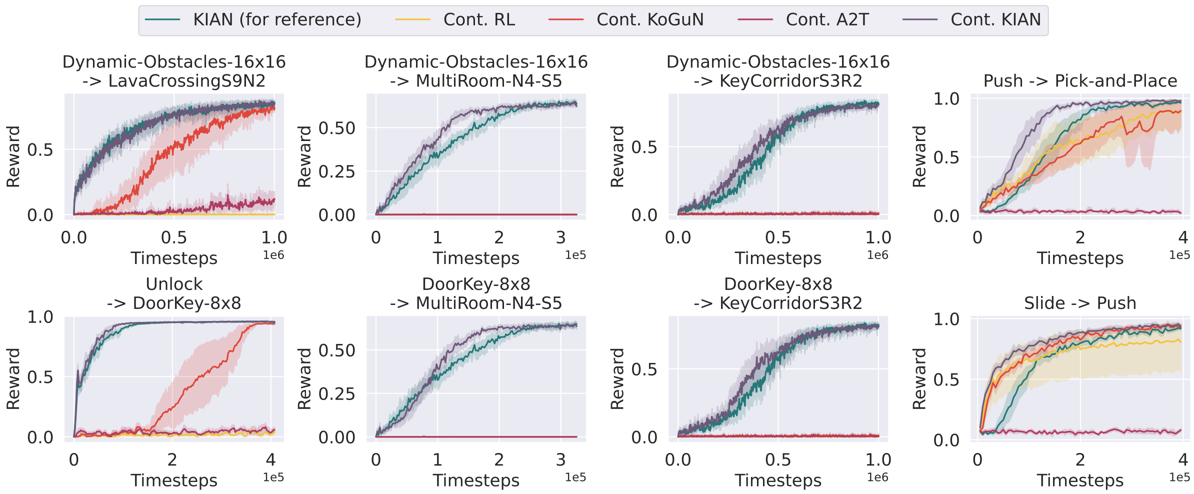


Figure 7.4: The learning curves of composition and incremental experiments in MiniGrid (left 3 columns) and OpenAI-Robotics (right column) environments. KIAN requires fewer samples to learn two tasks sequentially than separately and outperforms other approaches in incremental learning.

The learning curves in Figure 7.4 demonstrate that given the same updated $\mathcal{G}$, KIAN requires fewer samples to solve $\mathcal{M}_k^2$ than RL, KoGuN, and A2T in all experiments. Our knowledge-key and query design disentangles policy representations from the action-prediction operation, so the agent is more optimized in incremental learning. Unlike our disentangled design, prior methods use a single function approximator to directly predict an action (KoGuN) or the weight of each policy (A2T) given a state. These methods make the action-prediction operation depend on the number of knowledge policies, so changing the size of $\mathcal{G}$ requires significant retraining of the entire function approximator.

Figure 7.4 also shows that KIAN solves $\mathcal{M}_k^2$ more efficiently with $\mathcal{G}$ than $\mathcal{G}^{init}$ in most experiments. This improvement can be attributed to KIAN reusing the knowledge keys and query, which allows an agent to know which policies to fuse under different scenarios. Note that $\mathcal{G}$ can be further expanded with the internal policy learned in $\mathcal{M}_k^2$ and be used to solve another task $\mathcal{M}_k^3$.

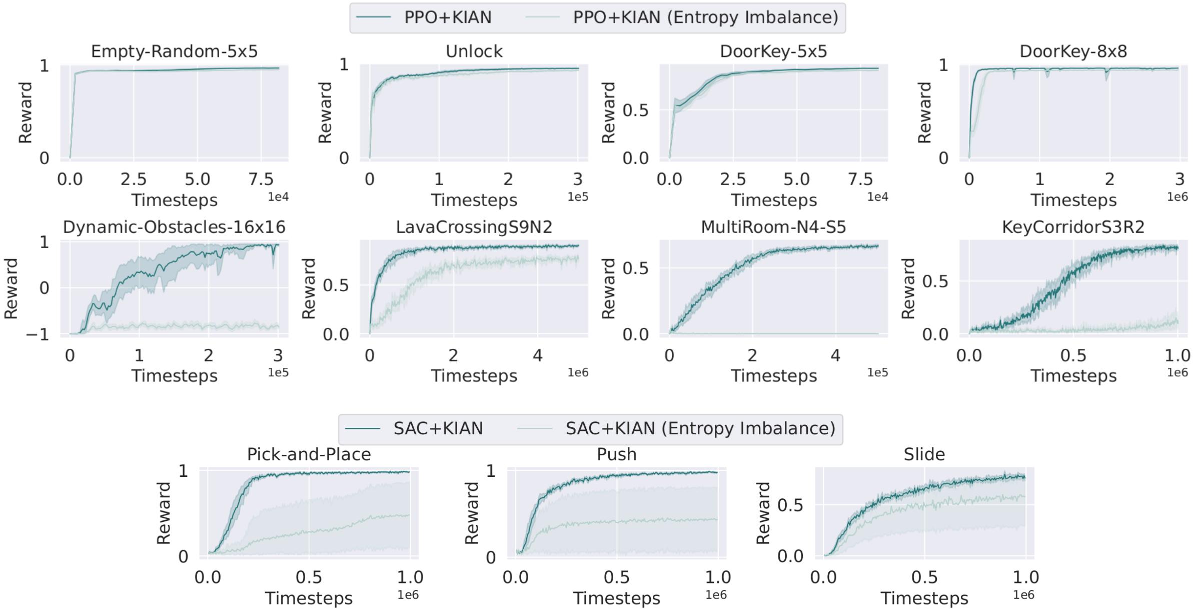


Figure 7.5: The learning curves of KIAN with and without addressing entropy imbalance as described in Section 7.3.4. The results indicate the adverse impact of entropy imbalance on KIAN's performance within the context of maximum entropy KGRL. In addition, our proposed modifications to external policy distributions are shown to be highly effective in alleviating this issue.

### 7.4.3 Analysis of Entropy Imbalance

In our ablation study, we investigate (1) the impact of entropy imbalance on the performance of maximum entropy KGRL and (2) whether the proposed modifications to external policy distributions in Section 7.3.4 can alleviate the issue.

Figure 7.5 shows the learning curves comparing KIAN's performance with and without addressing the entropy-imbalance issue. The results demonstrate that when not addressing the issue using equation (7.5) or (7.8), KIAN fails to fully capitalize on the guidance offered by external policies. We also draw two noteworthy conclusions from the figure: (1) For discrete decision-making tasks, the detrimental impact of entropy imbalance becomes more evident as task complexity increases. (2) For continuous-control tasks, entropy imbalance can degrade KIAN's performance and make it perform worse than pure RL without external policies, as shown by the results of FetchPickAndPlace and FetchPush. This phenomenon can be attributed to Proposition 7.3.3. In contrast, by adjusting KIAN's external policy distributions using equation (7.5) or (7.8), a KGRL agent can efficiently harness external policies to solve a given task.

# Chapter 8

# Modeling Data as Atoms in Deep Learning

Revisiting the basics can help us gain insights and solve an issue from the foundation. In this Chapter, we will shift attention from Chatbot learning, a more complex or advanced issue, to a more foundational one – Machine Learning. A core problem in machine learning is to learn expressive latent variables for model prediction on complex data that involves multiple sub-components in a flexible and interpretable fashion. Here, we develop an approach that improves expressiveness given a model is fixed, that is, the diversity of output distribution a model architecture can accommodate. Meanwhile, we expect the approach to provide partial interpretation, and is not restricted to specific applications. The key idea is to *dynamically distance data samples* in the latent space. Our dynamic latent separation method, inspired by atomic physics, relies on the jointly learned structures of each data sample, which also reveal the importance of each sub-component for distinguishing data samples. This approach, *atom modeling*, requires no supervision of the latent space and allows us to learn extra partially interpretable representations besides the original goal of a model. We empirically demonstrate that the algorithm also enhances the performance of small to larger-scale models in various classification and generation problems.

## 8.1 Introduction

Deep neural networks, with multiple hidden layers, are trained to express the complicated relationships between inputs and outputs [187]. Among various data types, data samples that consist of many sub-units, such as images or texts and their arbitrary segments as their sub-units, can require models to be more expressive to consider nuanced differences among sub-units. The demand for this delicacy leads to developing large-scale and complex model architectures [11], which cause drawbacks such as compromised model interpretability [31, 188, 57, 189, 44].

Various algorithms exist that improve model expressiveness not by advancing model architectures. For instance, contrastive learning ameliorates classification expressiveness [190, 126] by pushing away latent features from different classes. Vector quantization tackles the expressiveness of autoencoders [191] by learning discrete representations using a preset codebook. As a separate effort, post-hoc methods or designed models that follow self-explaining protocol [56] reveal some underlying reason for model behaviors. Explanations have also shown to help model training [45, 192]. While these methods show promising results in their bundled applications, it is yet certain of their transferability and usefulness to other applications. Meanwhile, it is yet underexplored of generalizable training algorithms that can simultaneously help expressiveness and uncover partial explanations.

We present a novel algorithm that simultaneously improves model output expressiveness, provides an interpretation of sub-component importance, and is generalizable to multiple applications. Our method, called *atom modeling*, first maps the latent represen-

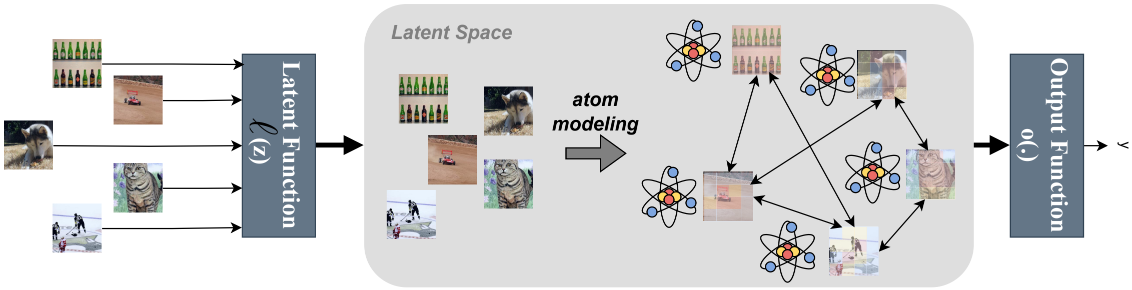


Figure 8.1: Illustration of atom modeling use case. Consider a model $f_\theta = o(\ell(\mathbf{z}))$; data samples are transformed into the latent space and their latent representations are distanced using atom modeling associated with the training criterion for output $y$. The colors labeled on each image in the latent space present the learned *token importance* that indicates which part is more crucial to identify data samples.

tations of each sub-unit (or referred to as *token*) in a data sample to a learnable *token importance* and then *dynamically distances data samples* based on token importance using a loss function derived from Coulomb force [193] in Physics. After training, token importance reveals which sub-components in a data sample contribute to its semantic meaning and are key to distinguishing itself from other data samples. The dynamic separation between data samples encourages a model to predict diverse outputs, thus boosting expressiveness.

This method can be viewed as connecting sub-component importance and inter-sample relationships to elevate impacts from local details. A similar observation can be found in atomic physics, where the balance distance between atoms, fundamental particles that form every matter in nature, depends on the structure of sub-atomic particles in each atom [194, 195]. In addition, applying atom modeling in a neural network also amounts to regularizing the representation space to preserve each data sample's uniqueness. Finally, atom modeling promotes expressiveness using a loss function with no latent supervision, enabling it to be flexibly applied to different applications.

In addition to theoretical proof of the dynamic separation effect of atom modeling as an auxiliary loss, we verify the method's efficacy via experiments. We train convolution neural networks, generative adversarial networks, and transformers on Gaussian mixtures, natural texts (CoLA, Poem), and natural images (MNIST, CIFAR10, CelebA-HQ, Oxford-IIIT Pets, Oxford-Flowers102, ImageNet-1K) with atom modeling as an auxiliary loss function. The empirical results demonstrate that atom modeling improves baselines under same setup and provides an interpretation of how each sub-unit affects the learning, and shows how atom modeling alters the inter-sample relationship.

## 8.2 Atom Modeling

Our goal is to define a flexible method that makes a model more expressive for data samples with multiple sub-units, such as images and texts, and does not need latent space supervision. We say a model is expressive if it can accommodate various distinct outputs for different inputs.

We define a model in a general form:

$$y = f_\theta(\mathbf{z}), \mathbf{z} \sim \mathcal{D}\,, \tag{8.1}$$

where $\mathcal{D}$ is the data distribution or a random noise distribution. We can easily fit models for practical applications, such as generation or classification, into this form: $y$ is often a real vector $\mathbf{y}$ for generation and a probability distribution $P(Y)$ for classification. If $f_\theta$ is expressive, different $\mathbf{z}$ is more likely to give different $\mathbf{y}$ or $P(Y)$. This distinction is desirable for promoting diversity in generation models [196] and encouraging entropy in classification models [197].

To achieve the goal of distinct outputs, we first write a model in its composite form:

$$f_\theta(\cdot) = o(\ell(\cdot)), \tag{8.2}$$

where $\ell(\cdot) \in \mathbb{R}^{N \times h}$ gives the latent representation of an input, and $o(\cdot)$ outputs the result given the latent representation. $N$ is the number of sub-components, and $h$ is the dimension of the latent space. An intuitive way to increase the probability that $f_\theta(\mathbf{z}^A)$ differs from $f_\theta(\mathbf{z}^B)$ is to let $\ell(\mathbf{z}^A)$ distance from $\ell(\mathbf{z}^B)$. Here, we show the properties of the output function, $o(\cdot)$, that lead to this concurrent increase.

**Lemma 8.2.1** *A G-Lipschitz function $o(\cdot)$ and a K-Lipschitz inverse function of $o(\cdot)$ returns the output space distance such that:*

$$K\|\mathbf{v} - \mathbf{u}\| \leq \|o(\mathbf{v}) - o(\mathbf{u})\| \leq G\|\mathbf{v} - \mathbf{u}\|\,, \tag{8.3}$$

*where $\mathbf{v}$ and $\mathbf{u}$ are any vector in the latent space.*

Equation 8.3 indicates that if the latent distance increases, the bounds of output distance also increase.

The next challenge is that, in a general case without latent space supervision, how we should set apart the latent variables produced by $\ell(\cdot)$. We propose to *dynamically* distance latent representations by separating the *currently close* variables and neglecting the *already distant* variables. Whether the variables are close or distant depends on their intra-sample structures. This leads us to first map a latent variable to a new embedding space by a learnable mapping function $\mathcal{A}(\cdot)$ such that:

$$\{(q_i, \mathbf{p}_i)\}_{i=1}^{N} = \mathcal{A}\left(\ell(\mathbf{z})\right)\,, \tag{8.4}$$

where $q_i \in \mathbb{R}$ is the *importance score* of the $i$-th row (token) in $\ell(\mathbf{z})$, and $\mathbf{p}_i \in \mathbb{R}^{h'}$ is the position of the same token in the new space.

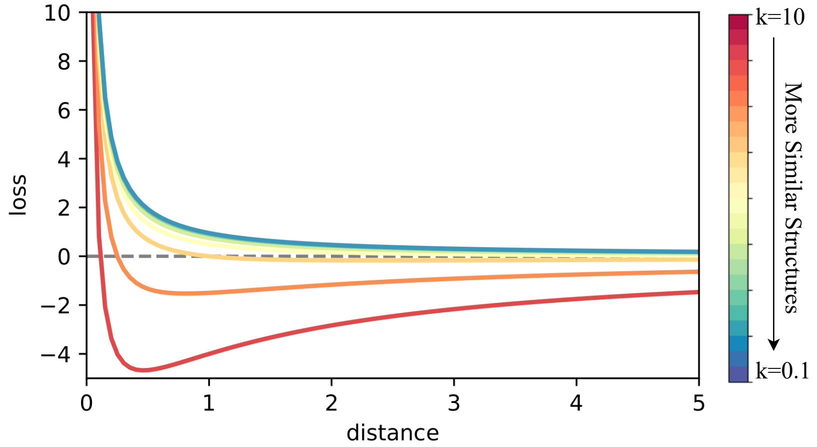


Figure 8.2: $\mathcal{L}_{\mathcal{A}}$ with varied atomic structure similarity $k$. The distance having the minimum loss depends on the intra-sample structures. As the structures are more similar (decayed $k$), the minimum loss distance becomes larger. Simultaneously, the distance cannot be zero.

Then, we propose a dynamic distancing loss function:

$$\mathcal{L}_{\mathcal{A}} = E_{\mathbf{z}^A,\mathbf{z}^B \sim \mathcal{D}} \sum_{i \in A, j \in B} \frac{q_i^A q_j^B}{d\left(q_i^A, q_j^B, \mathbf{p}_i^A, \mathbf{p}_j^B\right)}, \tag{8.5}$$

where $d\left(q_i^A, q_j^B, \mathbf{p}_i^A, \mathbf{p}_j^B\right) \in \mathbb{R}$ is a *distance* between $i$-th and $j$-th tokens in $\mathbf{z}^A$ and $\mathbf{z}^B$ and is derived from their intra-sample structures. We also use $A$ and $B$ as sets $\{1, \cdots, N_A\}$ and $\{1, \cdots, N_B\}$.

By minimizing $\mathcal{L}_{\mathcal{A}}$ in Equation 8.5, the optimal distance between $\ell(\mathbf{z}^A)$ and $\ell(\mathbf{z}^B)$ cannot be 0. That is, our proposed atomic loss forces $\ell(\mathbf{z}^A)$ and $\ell(\mathbf{z}^B)$ to be apart. In addition, the optimal distances are not identical for different data pairs, and these optimal values depend on each data's intra-sample structure. Figure 8.2 shows examples of the atomic loss function and optimal distances.

| | | *Attribute to Meaning?* | |
|---|---|---|---|
| | | Yes | No |
| *Distinguish data?* | Yes | +1 | -1 |
| | No | 0 | Not considered |

Table 8.1: Interpretation of token importance $q_i$.

### 8.2.1 Token Importance

In Equation 8.5, $q_i^A \in \mathbb{R}$ is a learnable importance score of a token in a data sample $A$. Given a latent representation of $A$, $\ell(\mathbf{z}^A) = [\mathbf{e}_1^A \mathbf{e}_2^A ... \mathbf{e}_{N_A}^A] \in \mathbb{R}^{N_A \times h}$, we define the token importance as:

$$q_i^A = 2\sigma(Q(\mathbf{e}_i^A)) - 1 \in [-1, 1] \,, \tag{8.6}$$

where $\sigma(\cdot)$ is the sigmoid function, and $Q(\cdot) : \mathbb{R}^h \mapsto \mathbb{R}$ maps the original h-dimension latent variable to an *unnormalized* importance score. We rescale the score to $[-1, 1]$ as it is a simple way to have three types of multiplication $q_i q_j$ needed in Equation 8.5: polarity (negative), likeness (positive), and no effect (zero). Since only when $q_i$ is not zero, $q_i q_j$ attributes to $\mathcal{L}_{\mathcal{A}}$, one role of token importance is as asking *if the i-th token in a data sample makes it distinguishable from other data samples.* As shown in Table 8.1, token importance +1 or -1 helps distinguish data while 0 does not.

### 8.2.2 Atomic Distance

We define the *atomic distance* between data samples A and B by:

$$\bar{d}_{AB} = \|\boldsymbol{\mu}_A - \boldsymbol{\mu}_B\|_p \,, \tag{8.7}$$

and the distances among their $i$-th and $j$-th tokens $d(q_i^A, q_j^B, \mathbf{p}_i^A, \mathbf{p}_j^B) \in \mathbb{R}$ in Equation 8.5 or $d_{ij}$ for brevity by:

$$d_{ij} = \bar{d}_{AB} + \frac{r_A + r_B}{2} step(-q_i^A q_j^B), \forall i \in A, j \in B\,. \tag{8.8}$$

Here $\boldsymbol{\mu}_A$ and $\boldsymbol{\mu}_B$ are respectively an average of the most crucial tokens of A and B, $r_A$ and $r_B$ are the token *deviation* within a data sample, and $step(\cdot)$ is the step function. We formally define them as:

$$\boldsymbol{\mu}_A = \frac{1}{N_A} \sum_{i \in A} m_i^A \mathbf{p}_i^A\,, \tag{8.9}$$

$$r_A = \frac{1}{N_A} \sum_{i \in A} (1 - m_i^A) \|\mathbf{p}_i^A - \boldsymbol{\mu}_A\|_p\,, \tag{8.10}$$

$$m_i^A = 1 - \max(-q_i^A, 0) \in [0, 1]\,, \tag{8.11}$$

$$\mathbf{p}_i^A = \mathcal{P}(\mathbf{e}_i^A) \in \mathbb{R}^{h'}\,, \tag{8.12}$$

where $\mathcal{P}(\cdot) : \mathbb{R}^h \mapsto \mathbb{R}^{h'}$ maps the original latent variable to the position of the $i$-th token in a new space, and $m_i^A \in \mathbb{R}$ is the *mass* of $i$-th token, indicating how much the token attributes to the meaning of A. As shown in Table 8.1, tokens with $q_i$ being +1 or 0 play a key role in the data meaning and have $m_i$=1, while tokens with importance -1 are less likely to attribute to data meaning.

Since Equation 8.5 optimizes the atomic distance $\bar{d}_{AB}$ that is not a conventional distance metric, we show its relationship to Euclidean distance.

**Theorem 8.2.1** *Consider equal token importance distribution, Equation 8.7 returns the atomic distance such that:*

$$\bar{d}_{AB} \leq C \|\tilde{\mathbf{v}}^A - \tilde{\mathbf{v}}^B\|_2\,, \tag{8.13}$$

*where $\tilde{\mathbf{v}}^A$ is a permutation of $\ell(\mathbf{z}^A)$ from data sample A.*

Theorem 8.2.1 implies that rising $\bar{d}_{AB}$ encourages separation of $\ell(\mathbf{z}^A)$ and $\ell(\mathbf{z}^B)$ in the Euclidean space. Therefore, according to Lemma 8.2.1, rising $\bar{d}_{AB}$ can increase the bounds of $\left\|o(\ell(\mathbf{z}^A)) - o(\ell(\mathbf{z}^B))\right\|_2$.

Next, we show that by optimizing Equation 8.5, the distance between two data samples depends on how similar their atomic structures are. In other words, the inter-sample relationship depends on the intra-sample structures. This dependence is crucial to achieve dynamic distance among data samples.

**Theorem 8.2.2** *Let $c = \sum_{q_iq_j>0} q_iq_j$ and $c^* = \sum_{q_iq_j<0} q_iq_j$ for all $i \in A$ and $j \in B$, given data samples $A$, $B$. Without loss of generality, $c^* = kc$ and $k \in (1, \infty)$ gives the optimal atomic distance in Equation 8.5 as:*

$$\bar{d}^*_{AB} = (\frac{r_A + r_B}{2})\frac{\sqrt{k}+1}{k-1} \tag{8.14}$$

*If $k \to 1$, $\bar{d}^*_{AB} \to \infty$ and $k \to \infty$, $\bar{d}^*_{AB} \to 0$*

Theorem 8.2.2 shows that optimizing Equation 8.5 forces data samples with similar intra-sample structures to separate more than the ones with dissimilar structures. Hence, our method results in *dynamic distancing.*

### 8.2.3 Training

We further prevent the model from learning every tokens equally important using a soft constraint on the token importance distribution, which is essential to form reasonable intra- and inter-sample relationship. We regularize the number of tokens with different importance scores to be similar. The complete loss function of Equation 8.5 becomes:

$$\mathcal{L}_{\mathcal{A}} = \underset{A,B\sim\mathcal{D}}{E} \sum_{i\in A, j\in B} \frac{q_i^A q_j^B}{d_{ij}} + \left(\sum_{i\in A} q_i\right)^2 + \left(\sum_{i\in A} q_i^2 - \frac{2}{3}N_A\right)^2 \tag{8.15}$$

Algorithm 2 lists the complete training process.

**Algorithm 2:** Atom Modeling

```
Input: data $\mathcal{D}$, model $f_\theta := o(h(\cdot))$, training criterion $\mathcal{L}_{ori}$, batch size $M$
for $t = 1$ to Training Ends do
    $(z, y) \sim \mathcal{D}$
    Form batch $\mathcal{B} = \{(\mathbf{z}^b, y^b)\}_{b=1}^{M}$
    $\mathbf{e}^b = h(\mathbf{z}^b)$ and $\hat{y}^b = o(\mathbf{e}^b)$
    Get $\mathcal{L}_{ori}(y, \hat{y})$ for all $(z, y) \sim \mathcal{B}$
    Map $\mathbf{e}^A, \mathbf{e}^B$ to $q_i^A$, $q_j^B$, $d_{ij}$ as Eq 8.6-8.12 $\forall z^A, z^B \sim \mathcal{B}$
    Get $\mathcal{L}_{\mathcal{A}}(q_i^A, q_j^B, d_{ij})$ as Eq 8.15
    Update $\theta$ by minimizing $\mathcal{L}_{ori} + \mathcal{L}_{\mathcal{A}}$
```

### 8.2.4 Relation to Atomic Physics.

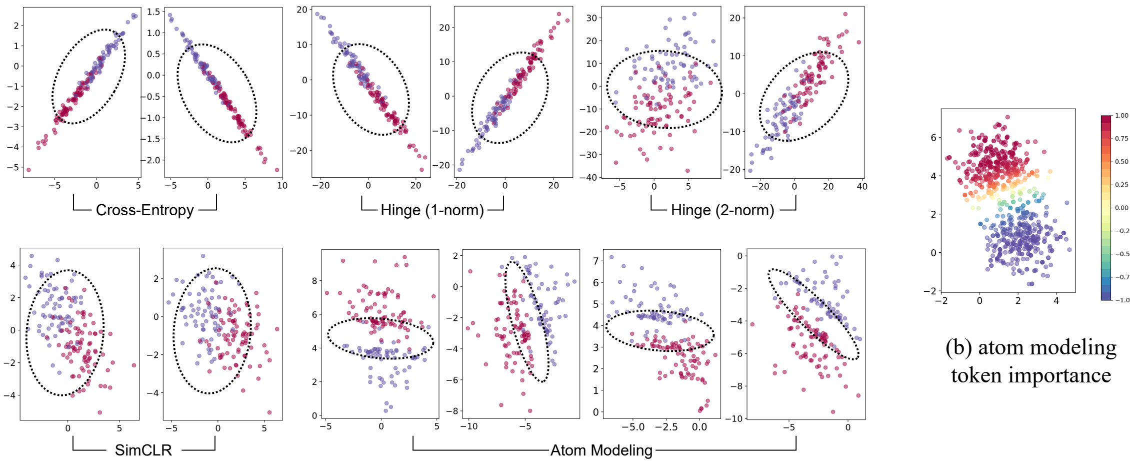
Our proposed method has high correspondence with atomic physics [195, 194], the scientific study of the structure of an atom and its interaction with others. Therefore, we name this method *atom modeling.*

Among atoms in nature, there are *inter-atomic* forces, similar to our proposed loss function in Equation 8.5, that bind atoms and avoid them collapsing by maintaining a balanced distance. The distances among a group of atoms depend on their *atomic structures*, similar to our theoretical result in Theorem 8.2.2.

Within an atom, its structure consists of three types of particles: neutrons, protons, and electrons, where a neutron has no charge, a proton has one positive charge with similar weight as a neutron, and an electron has a negative charge and weight significantly less than a proton [198]. The charges correspond to our token importance.

Simultaneously, [199] introduced one way to describe an atomic structure, where the protons and neutrons form a *nucleus*, similar to $\mu$, that occupies a small volume of the atom while the electrons orbit around the nucleus with a *radius*, similar to our $r$.

The soft constraint in Equation 8.15 is also similar to that the protons, electrons, and neutrons often have similar numbers within one atom.



(b) atom modeling token importance

(a) inter-sample relationship in latent space

Figure 8.3: (a) Visualization of the latent space of synthetic data by only cross-entropy training loss or integrated with hinge losses (L1 and L2), SimCLR, or Atom Modeling. Blue and red indicates the two ground-truth classes. The dashed circles annotate the overlaps of the learned representaions from different classes, which is correlated with the easiness to classify the samples. Atom modeling separates representations with a gap using no latent supervision. (b) Visualization of token importance.

# 8.3 Experiments

We test atom modeling's effects and flexibility by training linear classifiers on synthetic data, GANs on unconditional image generation, ResNets on image classification, and transformers on text classification.

In the experiments, while $Q(\cdot)$ and $\mathcal{P}(\cdot)$ can be any mapping functions, we use simple extraction functions with a selected hidden layer such that $Q(\cdot)$ extracts one dimension from the original $\ell(\cdot) \in \mathbb{R}^h$ and $\mathcal{P}(\cdot)$ extracts the rest $h-1$ dimensions.

## 8.3.1 Linear Classifier on Synthetic Data

To demonstrate the shifts of inter-sample relationships after atom modeling, we first conduct experiments on synthetic data. We mimic data of multiple sub-components by

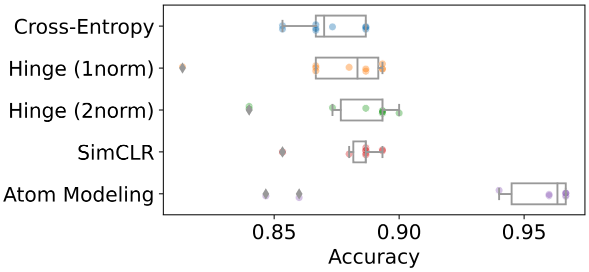


Figure 8.4: Comparison among cross-entropy, p-norm distance, SimCLR, and atom modeling.

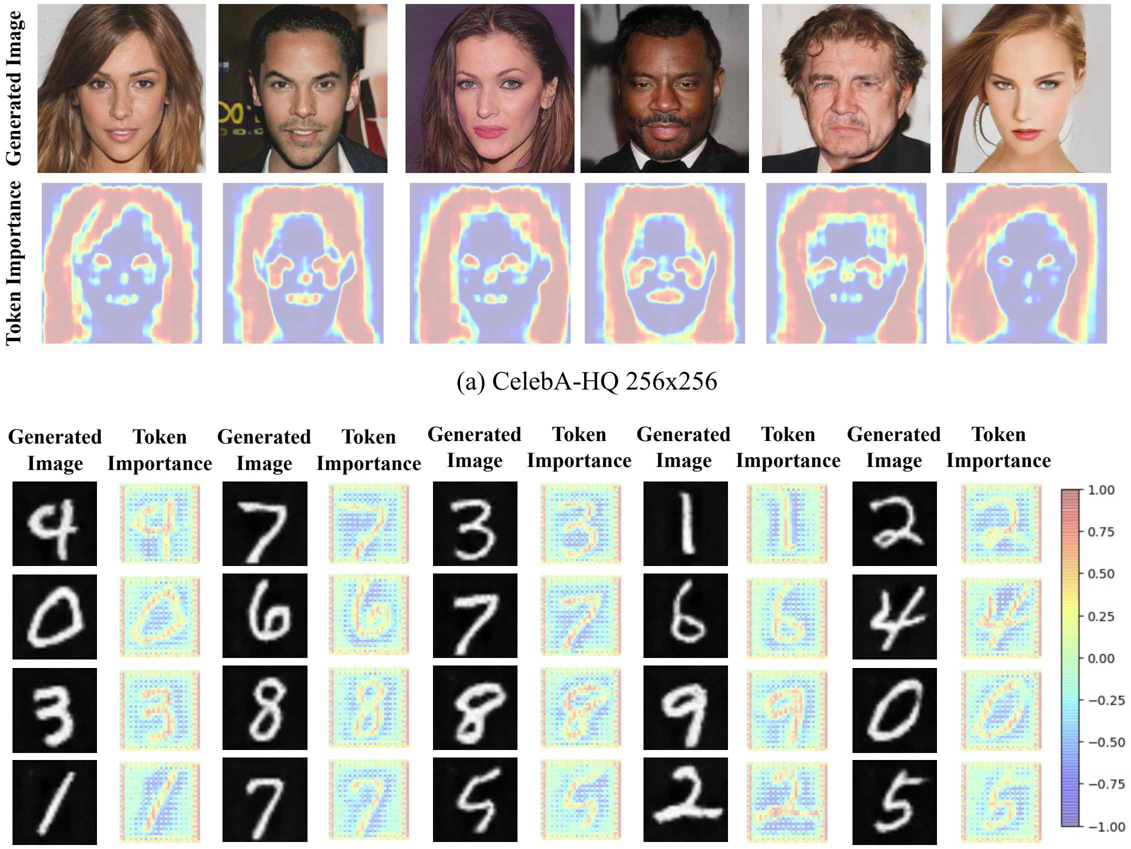
generating input features $X$ and the corresponding labels $y$ as follows:

$$\begin{cases} P(A) = \mathcal{N}(\mu_a, \sigma_a) \\ P(B) = \mathcal{N}(\mu_b, \sigma_b) \\ X = \{x_i | x_i \in A \cup B\}_{i=1}^{N} \\ y = \mathbb{1}\left(P(x_i \in A | x_i \in X) > P(x_i \in B | x_i \in X)\right) \end{cases} \tag{8.16}$$

where $A$ and $B$ are two events of normal distributions with ($\mu_a$, $\sigma_a$) and ($\mu_b$, $\sigma_b$) being the mean and standard deviation respectively. $X$ is the input composed of $N = 5$ sub-units $x_i$ sampled from $A \cup B$. The goal is to find a function $f : X \to y$.

In this experiment, we use a neural network with two fully-connected linear layers and apply atom modeling to the hidden state of the first layer. For comparison, we employ Hinge loss with p-norm distances [200, 201, 202] and SimCLR [126] that uses cosine similarity as the metric on the same hidden state, as our baselines. Figure 8.4 shows the classification accuracy across ten random runs. Atom modeling enhances the classifier to achieve an average of 96% accuracy and is superior to baselines.

We visualize how atom modeling alters inter-sample relationships and learns token importance. Figure 8.3(a) demonstrates that atom modeling spreads out the representation distribution, especially the high-density region. This further creates a gap between the blue and red classes in the latent space, thus enhancing the classifier expressivity.



(a) CelebA-HQ 256x256

(b) MNIST

Figure 8.5: Examples of generated images and the learned token importance by atom modeling on unconditional image generation. The distributions show that importance score close to one indicates it is a crucial part of the image to distinguish from others.

The corresponding token importance is plotted in Figure 8.3(b). The model takes sub-units near the $\mu_a$ $(q = +1)$ and $\mu_b$ $(q = -1)$ as the most crucial ones to distinguish data samples and takes sub-units near $\mu_a$ $(q = +1)$ and $(\mu_a + \mu_b)/2$ $(q = 0)$ as the keys to data meaning.

### 8.3.2 GANs on Unconditional Image Generation

We investigate atom modeling on generative models with unconditional image synthesis tasks: MNIST [203], CIFAR10 [204], and CelebA-HQ256x256 [205]. For MNIST

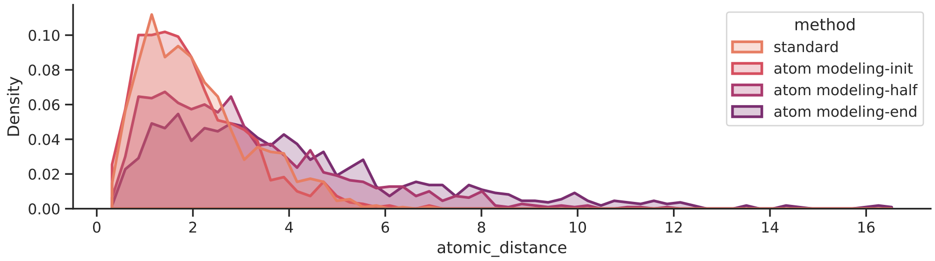


Figure 8.6: Alteration of atomic distance distributions by atom modeling (initial stage, half, end of training) and comparison with standard training criterion. Atom modeling gradually disseminates the atomic distance distribution.

Table 8.2: FIDs of image synthetic on MNIST, CIFAR10, and CelebA-HQ256x256.

| Method | Dataset | FID |
|---|---|---|
| DCGAN [206] | MNIST | 88.4 |
| VQ [191] | MNIST | 303.9 |
| SSCL [126] | MNIST | 69.4 |
| Atom Modeling | MNIST | **49.0** |
| DCGAN [206] | CIFAR10 | 110.4 |
| VQ [191] | CIFAR10 | - |
| SSCL [126] | CIFAR10 | 130.0 |
| Atom Modeling | CIFAR10 | **97.4** |
| VQGAN [208] | CelebA-HQ | 10.2 |
| StyleSwin [207] | CelebA-HQ | 5.26 |
| Atom Modeling | CelebA-HQ | **5.18** |

and CIFAR10, we performed experiments with DCGAN [206]. For CelebA-HQ256x256, we performed experiments with the SOTA model, StyleSwin [207], and followed their implementation.

For comparison, we employed self-supervised contrastive learning [126] and vector quantization [191] as additional loss to regularize a given representation layer and compared their ability with atom modeling for end-to-end training. In this experiment, for DCGAN, we use the output from the last hidden layer of the generator as the representation to be discretized. For StyleSwin, we use the output from the last hidden layer with half resolution. All the generated results are evaluated by Fréchet Inception

Distance (FID) [209, 210] with 2k images for MNIST, 10k for CIFAR10, and 30k for CelebA-HQ256x256. Lower FID indicates higher generation quality.

Empirically, Table 8.2 demonstrates the effectiveness of atom modeling. The proposed atom modeling outperforms VQ- or SSCL-enhanced DCGAN on MNIST (49.0 vs 69.4) and CIFAR10 (97.4 vs 110.4). Additionally, our approach improves StyleSwin on CelebA-HQ256x256 in our reproduced results (5.18 vs 5.26). Note that this promising improvement is gained under the original setup of DCGAN and StyleSwin without extra tuning. This shows that atom modeling can improve generative model expressivity in flexible settings.

Figure 8.5 shows examples of our generated images and the associated token importance after applying atom modeling. Areas mapped to hair, eyes, nose, and philtrum in CelebA-HQ256x256, as well as the cores of digits in MNIST, have positive importance scores. They play a crucial role in both distinguishing from other images and semantic meaning. Areas map to skin and background have negative importance scores. They are pivotal for differing from others but not the meaning. Other regions are less likely to identify an image but attribute to the semantic meaning. They are, therefore, assigned importance scores of zero.

We plot the atomic distance distribution over training time in Figure 8.6. At the initial training stage using atom modeling, the distances among data samples are similar to standard training results. The distances in the latent space concentrate to a small value. During training, atom modeling modifies the latent space and gradually disseminates the distance distribution. This matches our expectation of what atom modeling has done during model learning.

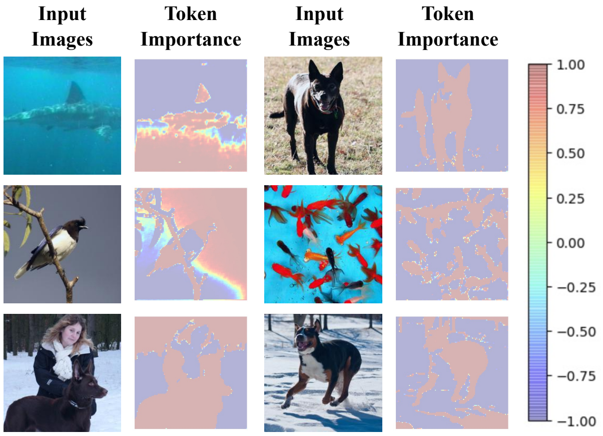


Figure 8.7: Visualization of the learned charges by atom modeling on unconditional image synthesis (CelebA-HQ 256x256, MNIST) and image classification (ImageNet-1K). The distributions show that protons (charge close to +1) are often the crucial parts in an image to be distinguished from others.

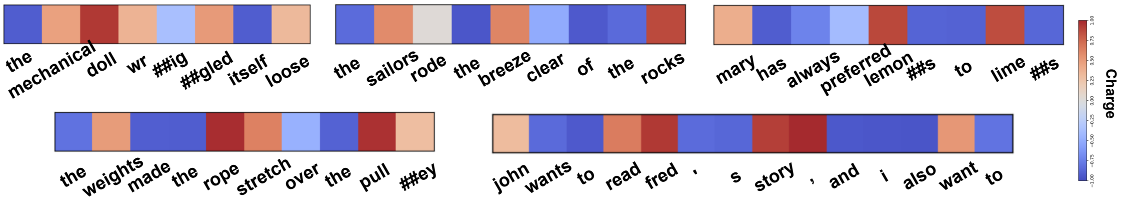


Figure 8.8: Visualization of the learned charges by atomic modeling on text classification (COLA). The charge distributions again show that protons are often mapped to the keywords in a sentence.

### 8.3.3 ResNet on Image Classification

We also validate atom modeling on fine-grained image classification Oxford-IIIT Pets [211] and Oxford-Flowers102 [212] as well as ImageNet-1K [213] to justify its flexibility.

In a fine-grained classification problem, intra-class diversity is higher than inter-class diversity [214], so we can use higher expressivity and, thus, atom modeling.

For comparison, we train ResNet18 [216] with cross-entropy loss, while employing hinge loss with p-norm distances, Rank-H [215], SimCLR [126]. In Table 8.3, we present the mean top-1 accuracy (Acc). The empirical results show that atom modeling consis-

Table 8.3: Results of fine-grained classification on Oxford-IIIT Pets and Oxford-Flowers102.

| | Pets | Flowers |
|---|---|---|
| Method | Acc | Acc |
| Cross-Entropy | 21.0 | 56.7 |
| Hinge (1-norm) [202] | 20.0 | 54.7 |
| Hinge (2-norm) [201] | 22.4 | 58.0 |
| Rank-H [215] | 22.4 | 58.2 |
| SimCLR [126] | 20.7 | 58.1 |
| Atom Modeling | **22.5** | **59.1** |

Table 8.4: Results of ImageNet-1K with [216, 217] data augmentation approaches.

| Method | Top-1 | Top-5 |
|---|---|---|
| Cross-Entropy w/ [216] | 74.97 | 92.17 |
| Atom Modeling w/ [216] | **75.10** | **92.25** |
| Cross-Entropy w/ [217] | 75.02 | 92.20 |
| Atom Modeling w/ [217] | **75.19** | **92.35** |

tently improves cross-entropy and is the best among distancing-representation-like approaches.

We further examine the ability of atom modeling applied to a larger-scale general classification problem on ImageNet-1K. We follow prior work implementations to use ResNet50 [216] as the backbone and run 90 epochs with two data augmentation methods used in [216, 217]. The first includes only the crop and horizontal flip, and the second adds color jitters and grayscale. Table 8.4 shows that atom modeling improves cross-entropy under different data augmentations, which has been found to impact the results of image classification [218, 219, 217]. Note that this performance gain has only been made by introducing atom modeling to one representation layer and using the exact same setup of conventional training of ResNet50 on ImageNet-1K; with a more elaborate setting, the performance could be improved. More importantly, the outcomes show that atom modeling can be flexibly applied to diverse data, models, and scales.

Table 8.5: Results of fine-grained classification on CoLA and Poem datasets.

| Method | CoLA<br>MCC | Poem<br>F1 |
|---|---|---|
| Cross-Entropy | 60.0 | 60.3 |
| Hinge (1-norm) [202] | 59.8 | 62.0 |
| Hinge (2-norm) [201] | 60.1 | 61.8 |
| SimCSE [220] | 60.8 | 59.3 |
| MixCSE [221] | 60.4 | 60.8 |
| Atom Modeling | **61.3** | **62.7** |

### 8.3.4 Transformer on Text Classification

We further experimented on fine-grained text classification: CoLA [222, 223] and Poem [224]. For comparison, we finetuned BERT [96] for language with cross-entropy loss, while employing hinge loss with p-norm distances, SimCSE [220], and MixCSE [221]. In Table 8.5, we present the Matthews's correlation coefficients (MCC) and F1 scores as used in prior work for each task. The empirical results show that our method consistently improves cross-entropy and is superior to the baselines.

The trained intra-sample relationship shows similarity to the vision domain. In Figure 8.8, we observe that tokens with special meanings have positive importance scores. They contribute to both the uniqueness and semantics of the sentence. The often-seen tokens, such as prepositions and articles, have negative importance scores. They contribute to distinguishing some sentences but less the semantics. Visualizability of the learned token importance exhibits the partial interpretability provided by atom modeling without post-hoc processing [31].

### 8.3.5 Ablation Study

We studied the impacts of the soft constraint $\left(\sum_{i\in A} q_i\right)^2+\left(\sum_{i\in A} q_i^2 - \frac{2}{3}N_A\right)^2$ as shown in Figure 8.9 and Table 8.6. We observe that only using the term $E_{A,B\sim\mathcal{D}} \sum_{i\in A, j\in B} \frac{q_i q_j}{d_{ij}}$, in

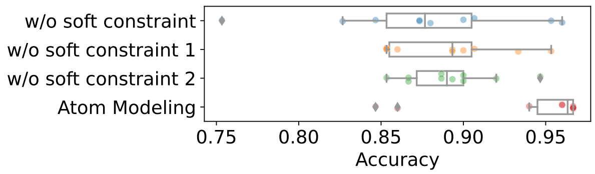


Figure 8.9: Ablation study of the soft constraint on linear classifiers of Gaussian mixtures with ten random runs.

| | MNIST | | CIFAR10 | |
|---|---|---|---|---|
| ablation | best | average | best | average |
| w/o constraint | -7.6 | -29.4 | -11.2 | -16.5 |
| w/o constraint 1 | -25.2 | -35.3 | -21.4 | -10.9 |
| w/o constraint 2 | -2.6 | -37.3 | -19.9 | -15.5 |

Table 8.6: Ablation study of the soft constraint on GANs of MNIST and CIFAR10 image synthesis FID with five random runs.

most cases, can achieve good performance but suffers from high variance. When having the soft constraint, the training performance is stabilized.

# Chapter 9

# Conclusion and Future Work

In this dissertation, we discuss understanding and learning human-like chatbots from the reasoning and optimization perspectives aligned with the statement: "*Human-like chatbots, a machine that involves the process of listening, understanding, reasoning, responding, and learning through interactions, mark the progress of machine learning development and can benefit humans through problem-solving efficiency, working assistance, and mental health improvement. While this thesis mainly discusses the reasoning and optimization parts, the latter will influence the whole process.*"

### Reasoning

In Chapter 2, we explored the possibility to understand conversational language models in depth. We proposed the local explanation method for response generation (LERG), which aims to explain the generation models through the mutual interactions between input and output features. LERG views the dialogue generation models as a certainty estimation of a human response so that it avoids dealing with the diverse output space. To facilitate future research, LERG also provides a unification and three properties of explanations for text generation. The experiments demonstrated that LERG can find explanations that can both recover a model's prediction and be interpreted by humans. In addition, with the background that an effective reasoning method over structured databases is vital for a dialogue system, we described DiffKG in Chapter 3, an end-to-end model-agnostic method that does symbolic reasoning on any scale of KGs to enhance response generation. Experiments demonstrated that using DiffKG, models are able to generate responses with interpretable KG reasoning paths at a modest extra cost. Chapter 2 and 3 together shape a view of how we can understand and perhaps improve the reasoning capability of chatbots. Specifically, Chapter 2 introduces LERG as a treat for an already trained or fine-tuned model, and Chapter 3 enhances the reasoning from a precautionary perspective.

### Optimization

To optimize a chatbot model more effectively beyond maximizing likelihood estimation (MLE), in this Chapter 4, we presented a new loss function for conversational model optimization, exponential maximizing average treatment effect (ExMATE), and a new dataset, CausalDialogue, with novel conversational DAG structure. With experiments on various model setups, we demonstrate that (1) ExMATE improves MLE in terms of diversity, informativeness, and agility; (2) CausalDialogue serves as a testbed for future research that needs abundant conversation cases, like causal inference and offline reinforcement learning. We further gathered and analyzed loss functions that share the same goal: simultaneously reward good data samples and penalize bad data samples in LM output distribution in Chapter 5. The representative methods we discuss are unlikelihood training, ExMATE, and direct preference optimization (DPO) in their proposed time order. We provide a novel perspective to consider the characteristics of generative LMs in gradient analysis: the multiple time steps, multiple classes, and literal similarity. Our approach splits gradient analysis into two primary cases: (1) When $y_t^+ \neq y_t^-$ and (2)

$y_t^+ = y_t^-$. From both mathematical results and experiments, we conclude that although DPO can significantly increase agility, defined as the gap between probability masses of positive and negative samples, it largely compromises perplexity and fails to introduce effective gradients when the initial information difference is small. In contrast, ExMATE consistently enlarges agility and simultaneously prevents the probability drops across situations, but with relatively minor improvements. These demonstrate that agility and perplexity are not necessarily trade-offs, but the quantity of improvements needs to be enhanced. With further experiments taking ExMATE as a surrogate of SFT or DPO, we suggest a more unified optimization method is first to train LMs by ExMATE instead of MLE. If the test case does not require a sufficiently high probability of given positive examples, then we use DPO to do further fine-tuning based on ExMATE.

In addition, as safety and helpfulness are vital attributes in LLMs for diverse scenarios, we presented a framework in Chapter 6 that self-generated data can rewind an aligned LLM and unlock its safety and helpfulness controllability. We showed step-by-step that while this task is challenging since the attributes in existing data are often trade-offs or entangled, the presented framework can successfully enhance a model's controllability. Without expensive manual annotations of attributes disentangled data, this framework reduces the barrier of having a controllable LLM. Among the examined optimization methods, ExMATE is again superior for such a case by rewarding the correct cause-effect pairs and penalizing the created negative examples, which is the false cause.

### Reasoning-inspired General Optimization

Moving towards more generalized optimization techniques for chatbots and their reasoning ability, Chapter 7 introduced KGRL, an RL paradigm aiming to enhance efficient and flexible learning by harnessing external policies. We proposed KIAN as an actor model for KGRL, which predicts an action by fusing multiple policies with an embedding-based attention operation. Furthermore, we propose modifications to KIAN's policy distributions to address entropy imbalance, which hinders efficient exploration with external policies in maximum entropy KGRL. Our experimental findings demonstrate that KIAN outperforms alternative methods incorporating external policies regarding sample efficiency, generalizability, and compositional and incremental learning. In Chapter 8, we further presented atom modeling, a new algorithm that takes each data example as an atom in Physics and achieves model expressiveness by (1) learning token importance for each sub-unit in a data sample and (2) dynamically distancing data samples based on their structural similarity with no supervision. In addition, the learned token importance provides a partial model explanation independent of the model optimization and the attention mechanism in transformers. Atom modeling is also highly practical, demonstrating effectiveness across diverse deep learning problems.

### Future Work

Building on the studies of reasoning and optimization towards a human-like chatbot, we propose further steps in the chatbot framework. These steps, which include taking models' explainability as evaluation metrics, integrating concept-level explanations, and utilizing relation information in KG reasoning, are crucial for advancing the reasoning field. Also, by involving speaker prediction in multi-party conversations, we aim to keep the audience engaged and interested. Moreover, anticipating the future steps of deep learning for chatbots, we can look forward to fundamental advancements. These include applying KGRL and Atom modeling for LLMs and reshaping the LM output distribution by rewarding good examples while penalizing bad ones. Specifically, as the knowledge policies in KGRL can be shared across various environments and continuously expanded, allowing artificial agents to flexibly query and learn from them, our research represents an initial step towards the overarching goal of KGRL: learning a knowledge set with a diverse range of policies. This idea is also essential to further improve LLMs / Chatbots by learning to mix multiple experts' representation space (continuous actions) or vocabulary space (discrete actions). While atom modeling has demonstrated that taking data samples as atoms to achieve dynamic latent separation can boost output diversity and interpretability, we expect to use it in natural language generation and Chatbot learning to improve learning efficiency, efficacy, and explainability. This promising trajectory of research instills optimism for the future of chatbot development.

# Bibliography

[1] OpenAI, *Chatgpt*, *https://chat.openai.com/* (2024).

[2] G. Team, R. Anil, S. Borgeaud, Y. Wu, J.-B. Alayrac, J. Yu, R. Soricut, J. Schalkwyk, A. M. Dai, A. Hauth, *et. al.*, *Gemini: a family of highly capable multimodal models*, *arXiv preprint arXiv:2312.11805* (2023).

[3] R. Thoppilan, D. De Freitas, J. Hall, N. Shazeer, A. Kulshreshtha, H.-T. Cheng, A. Jin, T. Bos, L. Baker, Y. Du, *et. al.*, *Lamda: Language models for dialog applications*, *arXiv preprint arXiv:2201.08239* (2022).

[4] M. B. Hoy, *Alexa, siri, cortana, and more: an introduction to voice assistants*, *Medical reference services quarterly* **37** (2018), no. 1 81–88.

[5] I. Sutskever, O. Vinyals, and Q. V. Le, *Sequence to sequence learning with neural networks*, *Advances in neural information processing systems (NIPS)* (2014).

[6] O. Vinyals and Q. Le, *A neural conversational model*, *arXiv preprint arXiv:1506.05869* (2015).

[7] J. Li, A. H. Miller, S. Chopra, M. Ranzato, and J. Weston, *Dialogue learning with human-in-the-loop*, *arXiv preprint arXiv:1611.09823* (2016).

[8] J. Williams, A. Raux, and M. Henderson, *The dialog state tracking challenge series: A review*, *Dialogue & Discourse* **7** (2016), no. 3 4–33.

[9] B. Goertzel and C. Pennachin, *Artificial general intelligence*, vol. 2. Springer, 2007.

[10] B. Goertzel, *Artificial general intelligence: concept, state of the art, and future prospects*, *Journal of Artificial General Intelligence* **5** (2014), no. 1 1.

[11] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, *Attention is all you need*, in *NIPS*, 2017.

[12] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever, *Improving language understanding by generative pre-training*, .

[13] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, and L. Zettlemoyer, *Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension*, *arXiv preprint arXiv:1910.13461* (2019).

[14] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, *Exploring the limits of transfer learning with a unified text-to-text transformer*, *Journal of Machine Learning Research* **21** (2020), no. 140 1–67.

[15] Y. Zhang, S. Sun, M. Galley, Y.-C. Chen, C. Brockett, X. Gao, J. Gao, J. Liu, and W. B. Dolan, *Dialogpt: Large-scale generative pre-training for conversational response generation*, in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pp. 270–278, 2020.

[16] Y. Bai, A. Jones, K. Ndousse, A. Askell, A. Chen, N. DasSarma, D. Drain, S. Fort, D. Ganguli, T. Henighan, *et. al.*, *Training a helpful and harmless assistant with reinforcement learning from human feedback*, *arXiv preprint arXiv:2204.05862* (2022).

[17] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, *et. al.*, *Llama: Open and efficient foundation language models*, *arXiv preprint arXiv:2302.13971* (2023).

[18] H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale, *et. al.*, *Llama 2: Open foundation and fine-tuned chat models*, *arXiv preprint arXiv:2307.09288* (2023).

[19] P. Budzianowski, T.-H. Wen, B.-H. Tseng, I. Casanueva, S. Ultes, O. Ramadan, and M. Gasic, *Multiwoz-a large-scale multi-domain wizard-of-oz dataset for task-oriented dialogue modelling*, in *EMNLP*, 2018.

[20] P. Ren, Z. Liu, X. Song, H. Tian, Z. Chen, Z. Ren, and M. de Rijke, *Wizard of search engine: Access to information through conversations with search engines*, in *Proceedings of the 44th International ACM SIGIR Conference on research and development in information retrieval*, pp. 533–543, 2021.

[21] J. Li, M. Galley, C. Brockett, G. Spithourakis, J. Gao, and W. B. Dolan, *A persona-based neural conversation model*, in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2016.

[22] A. Purington, J. G. Taft, S. Sannon, N. N. Bazarova, and S. H. Taylor, *" alexa is my new bff" social roles, user satisfaction, and personification of the amazon echo*, in *Proceedings of the 2017 CHI conference extended abstracts on human factors in computing systems*, pp. 2853–2859, 2017.

[23] S. Zhang, E. Dinan, J. Urbanek, A. Szlam, D. Kiela, and J. Weston, *Personalizing dialogue agents: I have a dog, do you have pets too?*, in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 2204–2213, 2018.

[24] F.-G. Su, A. R. Hsu, Y.-L. Tuan, and H.-Y. Lee, *Personalized dialogue response generation learned from monologues.*, in *INTERSPEECH*, 2019.

[25] M. Eric, L. Krishnan, F. Charette, and C. D. Manning, *Key-value retrieval networks for task-oriented dialogue*, in *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue*, pp. 37–49, 2017.

[26] M. Ghazvininejad, C. Brockett, M.-W. Chang, B. Dolan, J. Gao, W.-t. Yih, and M. Galley, *A knowledge-grounded neural conversation model*, in *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[27] Y.-L. Tuan, Y.-N. Chen, and H.-y. Lee, *Dykgchat: Benchmarking dialogue generation grounding on dynamic knowledge graphs*, in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 1855–1865, 2019.

[28] E. Dinan, S. Roller, K. Shuster, A. Fan, M. Auli, and J. Weston, *Wizard of wikipedia: Knowledge-powered conversational agents*, in *International Conference on Learning Representations*, 2018.

[29] N. Rahman and E. Santacana, *Beyond fair use: Legal risk evaluation for training llms on copyrighted text*, in *ICML Workshop on Generative AI and Law*, 2023.

[30] H. Chung, M. Iorga, J. Voas, and S. Lee, *Alexa, can i trust you?*, *Computer* **50** (2017), no. 9 100–104.

[31] M. T. Ribeiro, S. Singh, and C. Guestrin, *" why should i trust you?" explaining the predictions of any classifier*, in *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pp. 1135–1144, 2016.

[32] D. Alvarez-Melis and T. Jaakkola, *A causal framework for explaining the predictions of black-box sequence-to-sequence models*, in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pp. 412–421, 2017.

[33] P. Jiang, C. Sonne, W. Li, F. You, and S. You, *Preventing the immense increase in the life-cycle energy and carbon footprints of llm-powered intelligent chatbots*, *Engineering* (2024).

[34] E. J. Hu, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, W. Chen, *et. al.*, *Lora: Low-rank adaptation of large language models*, in *International Conference on Learning Representations*, 2022.

[35] T. Dettmers, A. Pagnoni, A. Holtzman, and L. Zettlemoyer, *Qlora: Efficient finetuning of quantized llms*, *Advances in Neural Information Processing Systems* **36** (2024).

[36] T. Schick, J. Dwivedi-Yu, R. Dessì, R. Raileanu, M. Lomeli, E. Hambro, L. Zettlemoyer, N. Cancedda, and T. Scialom, *Toolformer: Language models can teach themselves to use tools*, *Advances in Neural Information Processing Systems* **36** (2024).

[37] M. Ranzato, S. Chopra, M. Auli, and W. Zaremba, *Sequence level training with recurrent neural networks*, *ICLR* (2016).

[38] J. Li, W. Monroe, A. Ritter, D. Jurafsky, M. Galley, and J. Gao, *Deep reinforcement learning for dialogue generation*, in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 1192–1202, 2016.

[39] P. F. Christiano, J. Leike, T. Brown, M. Martic, S. Legg, and D. Amodei, *Deep reinforcement learning from human preferences*, *Advances in neural information processing systems* **30** (2017).

[40] J. Li, M. Galley, C. Brockett, J. Gao, and W. B. Dolan, *A diversity-promoting objective function for neural conversation models*, in *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 110–119, 2016.

[41] Y.-L. Tuan and H.-Y. Lee, *Improving conditional sequence generative adversarial networks by stepwise evaluation*, *IEEE/ACM Transactions on Audio, Speech, and Language Processing* **27** (2019), no. 4 788–798.

[42] K. Mo, Y. Zhang, S. Li, J. Li, and Q. Yang, *Personalizing a dialogue system with transfer reinforcement learning*, in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2018.

[43] Y.-L. Tuan, W. Wei, and W. Y. Wang, *Knowledge injection into dialogue generation via language models*, *arXiv preprint arXiv:2004.14614* (2020).

[44] Y.-L. Tuan, C. Pryor, W. Chen, L. Getoor, and W. Y. Wang, *Local explanation of dialogue response generation*, *arXiv preprint arXiv:2106.06528* (2021).

[45] Y.-L. Tuan, S. Beygi, M. Fazel-Zarandi, Q. Gao, A. Cervone, and W. Y. Wang, *Towards large-scale interpretable knowledge graph reasoning for dialogue systems*, in *Findings of the Association for Computational Linguistics: ACL 2022*, pp. 383–395, 2022.

[46] Y.-L. Tuan, A. Albalak, W. Xu, M. Saxon, C. Pryor, L. Getoor, and W. Y. Wang, *CausalDialogue: Modeling utterance-level causality in conversations*, in *Findings of the Association for Computational Linguistics (ACL)*, 2023.

[47] Y.-L. Tuan and W. Y. Wang, *A gradient analysis framework for rewarding good and penalizing bad examples in language models*, *arXiv preprint arXiv:2408.16751* (2024).

[48] Y.-L. Tuan, X. Chen, E. M. Smith, L. Martin, S. Batra, A. Celikyilmaz, W. Y. Wang, and D. M. Bikel, *Towards safety and helpfulness balanced responses via controllable large language models*, *arXiv preprint arXiv:2404.01295* (2024).

[49] Z.-Y. Chiu, Y.-L. Tuan, W. Y. Wang, and M. Yip, *Flexible attention-based multi-policy fusion for efficient deep reinforcement learning*, *Advances in Neural Information Processing Systems* **36** (2024).

[50] Y.-L. Tuan, Z.-Y. Chiu, and W. Y. Wang, *Dynamic latent separation for deep learning*, 2024.

[51] I. Kononenko, *Machine learning for medical diagnosis: history, state of the art and perspective*, *Artificial Intelligence in medicine* **23** (2001), no. 1 89–109.

[52] M. Bakator and D. Radosav, *Deep learning and medical diagnosis: A review of literature*, *Multimodal Technologies and Interaction* **2** (2018), no. 3 47.

[53] L. Relic, B. Zhang, Y.-L. Tuan, and M. Beyeler, *Deep learning–based perceptual stimulus encoder for bionic vision*, in *Proceedings of the Augmented Humans International Conference 2022*, pp. 323–325, 2022.

[54] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz, *et. al.*, *The kaldi speech recognition toolkit*, in *IEEE 2011 workshop on automatic speech recognition and understanding*, IEEE Signal Processing Society, 2011.

[55] C.-P. Tsai, Y.-L. Tuan, and L.-s. Lee, *Transcribing lyrics from commercial song audio: The first step towards singing content processing*, in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5749–5753, IEEE, 2018.

[56] D. Alvarez-Melis and T. S. Jaakkola, *Towards robust interpretability with self-explaining neural networks*, in *NeurIPS*, 2018.

[57] C. Rudin, *Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead*, *Nature Machine Intelligence* **1** (2019), no. 5 206–215.

[58] J. V. Jeyakumar, J. Noor, Y.-H. Cheng, L. Garcia, and M. Srivastava, *How can i explain this to you? an empirical study of deep neural network explanation methods*, *Advances in Neural Information Processing Systems* (2020).

[59] A. Shrikumar, P. Greenside, and A. Kundaje, *Learning important features through propagating activation differences*, in *International Conference on Machine Learning*, pp. 3145–3153, PMLR, 2017.

[60] S. M. Lundberg and S.-I. Lee, *A unified approach to interpreting model predictions*, in *Advances in neural information processing systems*, pp. 4765–4774, 2017.

[61] L. S. Shapley, *A value for n-person games*, .

[62] H. Chen and Y. Ji, *Learning variational word masks to improve the interpretability of neural text classifiers*, in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 4236–4251, 2020.

[63] D. Pruthi, B. Dhingra, L. B. Soares, M. Collins, Z. C. Lipton, G. Neubig, and W. W. Cohen, *Evaluating explanations: How much do explanations from the teacher aid students?*, *arXiv preprint arXiv:2012.00893* (2020).

[64] N. Asghar, P. Poupart, X. Jiang, and H. Li, *Deep active learning for dialogue generation*, in *Proceedings of the 6th Joint Conference on Lexical and Computational Semantics (* SEM 2017)*, pp. 78–83, 2017.

[65] L. H. Gilpin, D. Bau, B. Z. Yuan, A. Bajwa, M. Specter, and L. Kagal, *Explaining explanations: An overview of interpretability of machine learning*, in *2018 IEEE 5th International Conference on data science and advanced analytics (DSAA)*, pp. 80–89, IEEE, 2018.

[66] A. Fan, M. Lewis, and Y. Dauphin, *Hierarchical neural story generation*, in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 889–898, 2018.

[67] S. Kullback and R. A. Leibler, *On information and sufficiency*, *The annals of mathematical statistics* **22** (1951), no. 1 79–86.

[68] E. Strumbelj and I. Kononenko, *An efficient explanation of individual classifications using game theory*, *The Journal of Machine Learning Research* **11** (2010) 1–18.

[69] Y. Li, H. Su, X. Shen, W. Li, Z. Cao, and S. Niu, *DailyDialog: A manually labelled multi-turn dialogue dataset*, in *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, (Taipei, Taiwan), pp. 986–995, Asian Federation of Natural Language Processing, Nov., 2017.

[70] T. Wolf, V. Sanh, J. Chaumond, and C. Delangue, *Transfertransfo: A transfer learning approach for neural network based conversational agents*, *NeurIPS 2018 CAI Workshop* (2019).

[71] S. Wiegreffe and Y. Pinter, *Attention is not not explanation*, in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 11–20, 2019.

[72] M. Sundararajan, A. Taly, and Q. Yan, *Axiomatic attribution for deep networks*, in *International Conference on Machine Learning*, pp. 3319–3328, PMLR, 2017.

[73] E. Štrumbelj and I. Kononenko, *Explaining prediction models and individual predictions with feature contributions*, *Knowledge and information systems* **41** (2014), no. 3 647–665.

[74] C. Frye, C. Rowat, and I. Feige, *Asymmetric shapley values: incorporating causal knowledge into model-agnostic explainability*, *Advances in Neural Information Processing Systems* **33** (2020).

[75] I. E. Kumar, S. Venkatasubramanian, C. Scheidegger, and S. Friedler, *Problems with shapley-value-based explanations as feature importance measures*, in *International Conference on Machine Learning*, pp. 5491–5500, PMLR, 2020.

[76] P. Atanasova, J. G. Simonsen, C. Lioma, and I. Augenstein, *A diagnostic study of explainability techniques for text classification*, in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 3256–3274, 2020.

[77] E. L. Lehmann and J. P. Romano, *Testing statistical hypotheses*, vol. 3. Springer, 2005.

[78] M. Buhrmester, T. Kwang, and S. D. Gosling, *Amazon's mechanical turk: A new source of inexpensive, yet high-quality data?*, .

[79] D. Jurafsky and J. H. Martin, *Speech and language processing: an introduction to natural language processing, computational linguistics, and speech recognition, 2nd Edition.* Prentice Hall series in artificial intelligence. Prentice Hall, Pearson Education International, 2009.

[80] T.-H. Wen, M. Gasic, N. Mrksic, L. M. Rojas-Barahona, P.-H. Su, S. Ultes, D. Vandyke, and S. Young, *A network-based end-to-end trainable task-oriented dialogue system*, in *European Association for Computational Linguistics (EACL)*, pp. 438–449, 2017.

[81] I. V. Serban, A. Sordoni, Y. Bengio, A. Courville, and J. Pineau, *Building End-to-End Dialogue Systems Using Generative Hierarchical Neural Network Models*, in *Proceedings of the 30th AAAI Conference on Artificial Intelligence (AAAI-16)*, 2016.

[82] H. Zhou, T. Young, M. Huang, H. Zhao, J. Xu, and X. Zhu, *Commonsense knowledge aware conversation generation with graph attention.*, in *IJCAI*, 2018.

[83] S. Moon, P. Shah, A. Kumar, and R. Subba, *Opendialkg: Explainable conversational reasoning with attention-based walks over knowledge graphs*, in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 845–854, 2019.

[84] S. Yang, R. Zhang, and S. Erfani, *Graphdialog: Integrating graph knowledge into end-to-end task-oriented dialogue systems*, in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2020.

[85] W. W. Cohen, H. Sun, R. A. Hofer, and M. Siegler, *Scalable neural methods for reasoning with a symbolic knowledge base*, in *International Conference on Learning Representations*, 2019.

[86] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever, *et. al.*, *Language models are unsupervised multitask learners*, *OpenAI blog* **1** (2019), no. 8 9.

[87] K. Bollacker, C. Evans, P. Paritosh, T. Sturge, and J. Taylor, *Freebase: a collaboratively created graph database for structuring human knowledge*, in *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, 2008.

[88] J. Jung, B. Son, and S. Lyu, *Attnio: Knowledge graph exploration with in-and-out attention flow for knowledge-grounded dialogue*, in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2020.

[89] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, *Bleu: a method for automatic evaluation of machine translation*, in *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, 2002.

[90] C.-W. Liu, R. Lowe, I. V. Serban, M. Noseworthy, L. Charlin, and J. Pineau, *How not to evaluate your dialogue system: An empirical study of unsupervised*

*evaluation metrics for dialogue response generation*, in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 2122–2132, 2016.

[91] D. Khashabi, S. Min, T. Khot, A. Sabharwal, O. Tafjord, P. Clark, and H. Hajishirzi, *Unifiedqa: Crossing format boundaries with a single qa system*, in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings*, 2020.

[92] Y. Gou, Y. Lei, L. Liu, Y. Dai, and C. Shen, *Contextualize knowledge bases with transformer for end-to-end task-oriented dialogue systems*, in *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021* (M. Moens, X. Huang, L. Specia, and S. W. Yih, eds.), Association for Computational Linguistics, 2021.

[93] T. Luong, H. Pham, and C. D. Manning, *Effective approaches to attention-based neural machine translation*, in *EMNLP*, 2015.

[94] C. Gulcehre, S. Ahn, R. Nallapati, B. Zhou, and Y. Bengio, *Pointing the unknown words*, in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2016.

[95] N. Peng, H. Poon, C. Quirk, K. Toutanova, and W.-t. Yih, *Cross-sentence n-ary relation extraction with graph lstms*, *Transactions of the Association for Computational Linguistics* **5** (2017).

[96] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, *Bert: Pre-training of deep bidirectional transformers for language understanding*, in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 4171–4186, 2019.

[97] A. Madotto, C.-S. Wu, and P. Fung, *Mem2seq: Effectively incorporating knowledge bases into end-to-end task-oriented dialog systems*, in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2018.

[98] C.-S. Wu, R. Socher, and C. Xiong, *Global-to-local memory pointer networks for task-oriented dialogue*, in *International Conference on Learning Representations*, 2019.

[99] A. Sordoni, M. Galley, M. Auli, C. Brockett, Y. Ji, M. Mitchell, J.-Y. Nie, J. Gao, and B. Dolan, *A neural network approach to context-sensitive generation of conversational responses*, in *Proceedings of the 2015 Conference of the North*

*American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2015.

[100] S. Roller, E. Dinan, N. Goyal, D. Ju, M. Williamson, Y. Liu, J. Xu, M. Ott, E. M. Smith, Y.-L. Boureau, *et. al.*, *Recipes for building an open-domain chatbot*, in *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pp. 300–325, 2021.

[101] N. Jaques, J. H. Shen, A. Ghandeharioun, C. Ferguson, A. Lapedriza, N. Jones, S. Gu, and R. Picard, *Human-centric dialog training via offline reinforcement learning*, in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 3985–4003, 2020.

[102] J. Pearl, *Causality.* Cambridge university press, 2009.

[103] Y. Dou, M. Forbes, A. Holtzman, and Y. Choi, *Multitalk: A highly-branching dialog testbed for diverse conversations*, in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2021.

[104] Y. Li, H. Su, X. Shen, W. Li, Z. Cao, and S. Niu, *Dailydialog: A manually labelled multi-turn dialogue dataset*, in *IJCNLP*, 2017.

[105] J. Urbanek, A. Fan, S. Karamcheti, S. Jain, S. Humeau, E. Dinan, T. Rocktäschel, D. Kiela, A. Szlam, and J. Weston, *Learning to speak and act in a fantasy text adventure game*, in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 2019.

[106] P. W. Holland, *Statistics and causal inference*, *Journal of the American statistical Association* **81** (1986), no. 396 945–960.

[107] K. Imai, G. King, and E. A. Stuart, *Misunderstandings between experimentalists and observationalists about causal inference*, *Journal of the royal statistical society: series A (statistics in society)* **171** (2008), no. 2 481–502.

[108] C. Danescu-Niculescu-Mizil and L. Lee, *Chameleons in imagined conversations: A new approach to understanding coordination of linguistic style in dialogs.*, in *Proceedings of the Workshop on Cognitive Modeling and Computational Linguistics, ACL 2011*, 2011.

[109] R. E. Banchs, *Movie-dic: a movie dialogue corpus for research and development*, in *ACL*, 2012.

[110] P. Lison and J. Tiedemann, *Opensubtitles2016: Extracting large parallel corpora from movie and tv subtitles*, in *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pp. 923–929, 2016.

[111] S. Poria, D. Hazarika, N. Majumder, G. Naik, E. Cambria, and R. Mihalcea, *Meld: A multimodal multi-party dataset for emotion recognition in conversations*, in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 527–536, 2019.

[112] D. Yu, K. Sun, C. Cardie, and D. Yu, *Dialogue-based relation extraction*, in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 4927–4940, 2020.

[113] R. Rameshkumar and P. Bailey, *Storytelling with dialogue: A critical role dungeons and dragons dataset*, in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020.

[114] L. Cui, Y. Wu, S. Liu, Y. Zhang, and M. Zhou, *Mutual: A dataset for multi-turn dialogue reasoning*, in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 1406–1416, 2020.

[115] A. Albalak, Y.-L. Tuan, P. Jandaghi, C. Pryor, L. Yoffe, D. Ramachandran, L. Getoor, J. Pujara, and W. Y. Wang, *Feta: A benchmark for few-sample task transfer in open-domain dialogue*, in *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 10936–10953, 2022.

[116] A. Ritter, C. Cherry, and B. Dolan, *Data-driven response generation in social media*, in *Empirical Methods in Natural Language Processing (EMNLP)*, 2011.

[117] H. Wang, Z. Lu, H. Li, and E. Chen, *A dataset for research on short-text conversations*, in *Proceedings of the 2013 conference on empirical methods in natural language processing*, pp. 935–945, 2013.

[118] R. Lowe, N. Pow, I. V. Serban, and J. Pineau, *The ubuntu dialogue corpus: A large dataset for research in unstructured multi-turn dialogue systems*, in *SIGDIAL*, 2015.

[119] R. Pasunuru and M. Bansal, *Game-based video-context dialogue*, in *EMNLP*, 2018.

[120] K. Nakamura, S. Levy, Y.-L. Tuan, W. Chen, and W. Y. Wang, *Hybridialogue: An information-seeking dialogue dataset grounded on tabular and textual data*, in *Findings of the Association for Computational Linguistics: ACL 2022*, pp. 481–492, 2022.

[121] H. Rashkin, E. M. Smith, M. Li, and Y.-L. Boureau, *Towards empathetic open-domain conversation models: A new benchmark and dataset*, in *ACL*, 2019.

[122] A. Narayan-Chen, P. Jayannavar, and J. Hockenmaier, *Collaborative dialogue in minecraft*, in *ACL*, 2019.

[123] P. Ammanabrolu, J. Urbanek, M. Li, A. Szlam, T. Rocktäschel, and J. Weston, *How to motivate your dragon: Teaching goal-driven agents to speak and act in fantasy worlds*, in *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2021.

[124] F. Mairesse, M. A. Walker, M. R. Mehl, and R. K. Moore, *Using linguistic cues for the automatic recognition of personality in conversation and text*, *Journal of artificial intelligence research* **30** (2007) 457–500.

[125] Y. Goldberg and O. Levy, *word2vec explained: deriving mikolov et al.'s negative-sampling word-embedding method*, *arXiv preprint arXiv:1402.3722* (2014).

[126] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, *A simple framework for contrastive learning of visual representations*, in *International conference on machine learning*, pp. 1597–1607, PMLR, 2020.

[127] S. Welleck, I. Kulikov, S. Roller, E. Dinan, K. Cho, and J. Weston, *Neural text generation with unlikelihood training*, in *International Conference on Learning Representations*, 2019.

[128] P.-E. Mazaré, S. Humeau, M. Raison, and A. Bordes, *Training millions of personalized dialogue agents.*, in *EMNLP*, 2018.

[129] S. Hochreiter and J. Schmidhuber, *Long short-term memory*, *Neural computation* **9** (1997), no. 8 1735–1780.

[130] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, *Learning phrase representations using rnn encoder-decoder for statistical machine translation*, in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1724–1734, 2014.

[131] Ren and Malik, *Learning a classification model for segmentation*, in *Proceedings ninth IEEE international conference on computer vision*, IEEE, 2003.

[132] S. Welleck, I. Kulikov, S. Roller, E. Dinan, K. Cho, and J. Weston, *Neural text generation with unlikelihood training*, in *International Conference on Learning Representations (ICLR)*, 2020.

[133] L. Yu, W. Zhang, J. Wang, and Y. Yu, *Seqgan: Sequence generative adversarial nets with policy gradient*, in *Proceedings of the AAAI conference on artificial intelligence (AAAI)*, 2017.

[134] D. M. Ziegler, N. Stiennon, J. Wu, T. B. Brown, A. Radford, D. Amodei, P. Christiano, and G. Irving, *Fine-tuning language models from human preferences*, *arXiv preprint arXiv:1909.08593* (2019).

[135] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray, *et. al.*, *Training language models to follow instructions with human feedback*, *Advances in Neural Information Processing Systems (NeurIPS)* (2022).

[136] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, *Generative adversarial nets*, *Advances in neural information processing systems (NIPS)* (2014).

[137] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, *Proximal policy optimization algorithms*, *arXiv preprint arXiv:1707.06347* (2017).

[138] Y.-L. Tuan, J. Zhang, Y. Li, and H.-y. Lee, *Proximal policy optimization and its dynamic version for sequence generation*, *arXiv preprint arXiv:1808.07982* (2018).

[139] R. Rafailov, A. Sharma, E. Mitchell, C. D. Manning, S. Ermon, and C. Finn, *Direct preference optimization: Your language model is secretly a reward model*, *Advances in Neural Information Processing Systems (NIPS)* (2024).

[140] T. Mikolov, K. Chen, G. Corrado, and J. Dean, *Efficient estimation of word representations in vector space*, *arXiv preprint arXiv:1301.3781* (2013).

[141] M. Li, S. Roller, I. Kulikov, S. Welleck, Y.-L. Boureau, K. Cho, and J. Weston, *Don't say that! making inconsistent dialogue unlikely with unlikelihood training*, in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2020.

[142] J. Li, W. Monroe, T. Shi, S. Jean, A. Ritter, and D. Jurafsky, *Adversarial learning for neural dialogue generation*, in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pp. 2157–2169, 2017.

[143] Y.-L. Tuan, A. El-Kishky, A. Renduchintala, V. Chaudhary, F. Guzmán, and L. Specia, *Quality estimation without human-labeled data*, in *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pp. 619–625, 2021.

[144] W. Xu, Y.-L. Tuan, Y. Lu, M. Saxon, L. Li, and W. Y. Wang, *Not all errors are equal: Learning text generation metrics using stratified error synthesis*, in *Findings of the Association for Computational Linguistics: EMNLP 2022*, pp. 6559–6574, 2022.

[145] M. G. Azar, M. Rowland, B. Piot, D. Guo, D. Calandriello, M. Valko, and R. Munos, *A general theoretical paradigm to understand learning from human preferences*, *arXiv preprint arXiv:2310.12036* (2023).

[146] K. Ethayarajh, W. Xu, N. Muennighoff, D. Jurafsky, and D. Kiela, *Kto: Model alignment as prospect theoretic optimization*, *arXiv preprint arXiv:2402.01306* (2024).

[147] J. Hong, N. Lee, and J. Thorne, *Orpo: Monolithic preference optimization without reference model*, 2024.

[148] D. Bahdanau, P. Brakel, K. Xu, A. Goyal, R. Lowe, J. Pineau, A. Courville, and Y. Bengio, *An actor-critic algorithm for sequence prediction*, in *International Conference on Learning Representations (ICLR)*, 2016.

[149] K. Kandasamy, Y. Bachrach, R. Tomioka, D. Tarlow, and D. Carter, *Batch policy gradient methods for improving neural conversation models*, in *International Conference on Learning Representations (ICLR)*, 2017.

[150] K. Q. Weinberger and L. K. Saul, *Distance metric learning for large margin nearest neighbor classification.*, *Journal of machine learning research* **10** (2009), no. 2.

[151] M. Gutmann and A. Hyvärinen, *Noise-contrastive estimation: A new estimation principle for unnormalized statistical models*, in *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pp. 297–304, JMLR Workshop and Conference Proceedings, 2010.

[152] K. Sohn, *Improved deep metric learning with multi-class n-pair loss objective*, *Advances in neural information processing systems* **29** (2016).

[153] A. v. d. Oord, Y. Li, and O. Vinyals, *Representation learning with contrastive predictive coding*, *arXiv preprint arXiv:1807.03748* (2018).

[154] S. Biderman, H. Schoelkopf, Q. G. Anthony, H. Bradley, K. O'Brien, E. Hallahan, M. A. Khan, S. Purohit, U. S. Prashanth, E. Raff, *et. al.*, *Pythia: A suite for analyzing large language models across training and scaling*, in *International Conference on Machine Learning*, pp. 2397–2430, PMLR, 2023.

[155] M. L. McHugh, *Interrater reliability: the kappa statistic*, *Biochemia medica* **22** (2012), no. 3 276–282.

[156] A. Chowdhery, S. Narang, J. Devlin, M. Bosma, G. Mishra, A. Roberts, P. Barham, H. W. Chung, C. Sutton, S. Gehrmann, *et. al.*, *Palm: Scaling language modeling with pathways*, *arXiv preprint arXiv:2204.02311* (2022).

[157] OpenAI, *Gpt-4 technical report*, *ArXiv* **abs/2303.08774** (2023).

[158] J. Pearl, *Causal inference in statistics: An overview*, .

[159] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction.* MIT press, 2018.

[160] S. Dathathri, A. Madotto, J. Lan, J. Hung, E. Frank, P. Molino, J. Yosinski, and R. Liu, *Plug and play language models: A simple approach to controlled text generation*, in *International Conference on Learning Representations*, 2019.

[161] F. Petroni, T. Rocktäschel, S. Riedel, P. Lewis, A. Bakhtin, Y. Wu, and A. Miller, *Language models as knowledge bases?*, in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 2463–2473, 2019.

[162] J. Wei, X. Wang, D. Schuurmans, M. Bosma, F. Xia, E. Chi, Q. V. Le, D. Zhou, *et. al.*, *Chain-of-thought prompting elicits reasoning in large language models*, *Advances in Neural Information Processing Systems* **35** (2022) 24824–24837.

[163] N. Hossain, M. Ghazvininejad, and L. Zettlemoyer, *Simple and effective retrieve-edit-rerank text generation*, in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 2532–2538, 2020.

[164] K. Krishna, Y. Chang, J. Wieting, and M. Iyyer, *Rankgen: Improving text generation with large ranking models*, in *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 199–232, 2022.

[165] I. Loshchilov and F. Hutter, *Decoupled weight decay regularization*, in *International Conference on Learning Representations*, 2019.

[166] A. Holtzman, J. Buys, L. Du, M. Forbes, and Y. Choi, *The curious case of neural text degeneration*, in *International Conference on Learning Representations*, 2019.

[167] P. R. Wurman, S. Barrett, K. Kawamoto, J. MacGlashan, K. Subramanian, T. J. Walsh, R. Capobianco, A. Devlic, F. Eckert, F. Fuchs, *et. al.*, *Outracing champion gran turismo drivers with deep reinforcement learning*, *Nature* **602** (2022), no. 7896 223–228.

[168] J. Degrave, F. Felici, J. Buchli, M. Neunert, B. Tracey, F. Carpanese, T. Ewalds, R. Hafner, A. Abdolmaleki, D. de Las Casas, *et. al.*, *Magnetic control of tokamak plasmas through deep reinforcement learning*, *Nature* **602** (2022), no. 7897 414–419.

[169] D. Kalashnikov, A. Irpan, P. Pastor, J. Ibarz, A. Herzog, E. Jang, D. Quillen, E. Holly, M. Kalakrishnan, V. Vanhoucke, *et. al.*, *Scalable deep reinforcement learning for vision-based robotic manipulation*, in *Conference on Robot Learning*, pp. 651–673, PMLR, 2018.

[170] S. Song, Ł. Kidziński, X. B. Peng, C. Ong, J. Hicks, S. Levine, C. G. Atkeson, and S. L. Delp, *Deep reinforcement learning for modeling human locomotion control in neuromechanical simulation*, *Journal of neuroengineering and rehabilitation* **18** (2021), no. 1 1–17.

[171] R. Agarwal, M. Schwarzer, P. S. Castro, A. C. Courville, and M. Bellemare, *Reincarnating reinforcement learning: Reusing prior computation to accelerate progress*, *Advances in Neural Information Processing Systems* **35** (2022) 28955–28971.

[172] A. Bandura, *Social Learning Theory.* Prentice-Hall series in social learning theory. Prentice Hall, 1977.

[173] L. P. Kaelbling, *The foundation of efficient robot learning*, *Science* **369** (2020), no. 6506 915–916.

[174] A. Nair, B. McGrew, M. Andrychowicz, W. Zaremba, and P. Abbeel, *Overcoming exploration in reinforcement learning with demonstrations*, in *2018 IEEE international conference on robotics and automation (ICRA)*, pp. 6292–6299, IEEE, 2018.

[175] J. Rajendran, A. Lakshminarayanan, M. M. Khapra, P. P, and B. Ravindran, *Attend, adapt and transfer: Attentive deep architecture for adaptive transfer from multiple sources in the same domain*, in *International Conference on Learning Representations*, 2017.

[176] P. Zhang, J. Hao, W. Wang, H. Tang, Y. Ma, Y. Duan, and Y. Zheng, *Kogun: Accelerating deep reinforcement learning via integrating human suboptimal knowledge*, in *International Joint Conference on Artificial Intelligence*, 2020.

[177] A. H. Qureshi, J. J. Johnson, Y. Qin, T. Henderson, B. Boots, and M. C. Yip, *Composing task-agnostic policies with deep reinforcement learning*, in *International Conference on Learning Representations*, 2020.

[178] Z.-Y. Chiu, Y.-L. Tuan, H.-y. Lee, and L.-C. Fu, *Parallelized reverse curriculum generation*, *arXiv preprint arXiv:2108.02128* (2021).

[179] M. Chevalier-Boisvert, L. Willems, and S. Pal, “Minimalistic gridworld environment for openai gym.” `https://github.com/maximecb/gym-minigrid`, 2018.

[180] M. Plappert, M. Andrychowicz, A. Ray, B. McGrew, B. Baker, G. Powell, J. Schneider, J. Tobin, M. Chociej, P. Welinder, *et. al.*, *Multi-goal reinforcement learning: Challenging robotics environments and request for research*, *arXiv preprint arXiv:1802.09464* (2018).

[181] E. Jang, S. Gu, and B. Poole, *Categorical reparameterization with gumbel-softmax*, *arXiv preprint arXiv:1611.01144* (2016).

[182] B. D. Ziebart, *Modeling purposeful adaptive behavior with the principle of maximum causal entropy.* Carnegie Mellon University, 2010.

[183] T. Haarnoja, H. Tang, P. Abbeel, and S. Levine, *Reinforcement learning with deep energy-based policies*, in *International conference on machine learning*, pp. 1352–1361, PMLR, 2017.

[184] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, *Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor*, in *ICML*, 2018.

[185] D. J. MacKay, *Information theory, inference and learning algorithms.* Cambridge university press, 2003.

[186] M. Bain and C. Sammut, *A framework for behavioural cloning*, in *Machine Intelligence 15*, 1995.

[187] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, *Dropout: a simple way to prevent neural networks from overfitting*, *The journal of machine learning research* **15** (2014), no. 1 1929–1958.

[188] O. Bastani, C. Kim, and H. Bastani, *Interpreting blackbox models via model extraction*, *arXiv preprint arXiv:1705.08504* (2017).

[189] S. Jain and B. C. Wallace, *Attention is not explanation*, in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 3543–3556, 2019.

[190] A. Dosovitskiy, J. T. Springenberg, M. Riedmiller, and T. Brox, *Discriminative unsupervised feature learning with convolutional neural networks*, *Advances in neural information processing systems* **27** (2014).

[191] A. Van Den Oord, O. Vinyals, *et. al.*, *Neural discrete representation learning*, *Advances in neural information processing systems* **30** (2017).

[192] A. Albalak, V. Embar, Y.-L. Tuan, L. Getoor, and W. Y. Wang, *D-rex: Dialogue relation extraction with explanations*, in *Proceedings of the 4th Workshop on NLP for Conversational AI*, pp. 34–46, 2022.

[193] C. A. Coulomb, *Premier mémoire sur l'électricité et le magnétisme*, *Histoire de l'Academie royale des sciences* **569** (1785).

[194] T. L. Brown, *Chemistry: the central science.* Pearson Education, 2009.

[195] D. Halliday, R. Resnick, and J. Walker, *Fundamentals of physics.* John Wiley & Sons, 2013.

[196] A. Razavi, A. Van den Oord, and O. Vinyals, *Generating diverse high-fidelity images with vq-vae-2*, *Advances in neural information processing systems* **32** (2019).

[197] A. Dubey, O. Gupta, R. Raskar, and N. Naik, *Maximum-entropy fine grained classification*, *Advances in neural information processing systems* **31** (2018).

[198] P. J. Mohr, B. N. Taylor, and D. B. Newell, *Codata recommended values of the fundamental physical constants: 2006*, *Journal of Physical and Chemical Reference Data* **80** (2008), no. 3 633–1284.

[199] N. Bohr, *I. on the constitution of atoms and molecules*, *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science* **26** (1913), no. 151 1–25.

[200] J. Bromley, I. Guyon, Y. LeCun, E. Säckinger, and R. Shah, *Signature verification using a" siamese" time delay neural network*, *Advances in neural information processing systems* **6** (1993).

[201] S. Chopra, R. Hadsell, and Y. LeCun, *Learning a similarity metric discriminatively, with application to face verification*, in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, vol. 1, pp. 539–546, IEEE, 2005.

[202] Y. LeCun, S. Chopra, R. Hadsell, M. Ranzato, and F. Huang, *A tutorial on energy-based learning*, *Predicting structured data* **1** (2006), no. 0.

[203] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, *Gradient-based learning applied to document recognition*, *Proceedings of the IEEE* **86** (1998), no. 11 2278–2324.

[204] A. Krizhevsky, G. Hinton, *et. al.*, *Learning multiple layers of features from tiny images*, .

[205] T. Karras, T. Aila, S. Laine, and J. Lehtinen, *Progressive growing of GANs for improved quality, stability, and variation*, in *International Conference on Learning Representations*, 2018.

[206] A. Radford, L. Metz, and S. Chintala, *Unsupervised representation learning with deep convolutional generative adversarial networks*, *arXiv preprint arXiv:1511.06434* (2015).

[207] B. Zhang, S. Gu, B. Zhang, J. Bao, D. Chen, F. Wen, Y. Wang, and B. Guo, *Styleswin: Transformer-based gan for high-resolution image generation*, in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 11304–11314, 2022.

[208] P. Esser, R. Rombach, and B. Ommer, *Taming transformers for high-resolution image synthesis*, in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 12873–12883, 2021.

[209] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, *Gans trained by a two time-scale update rule converge to a local nash equilibrium*, *Advances in neural information processing systems* **30** (2017).

[210] A. Obukhov, M. Seitzer, P.-W. Wu, S. Zhydenko, J. Kyl, and E. Y.-J. Lin, *High-fidelity performance metrics for generative models in pytorch*, 2020. Version: 0.3.0, DOI: 10.5281/zenodo.4957738.

[211] O. M. Parkhi, A. Vedaldi, A. Zisserman, and C. Jawahar, *Cats and dogs*, in *2012 IEEE conference on computer vision and pattern recognition*, pp. 3498–3505, IEEE, 2012.

[212] M.-E. Nilsback and A. Zisserman, *Automated flower classification over a large number of classes*, in *2008 Sixth Indian Conference on Computer Vision, Graphics & Image Processing*, pp. 722–729, IEEE, 2008.

[213] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, *Imagenet: A large-scale hierarchical image database*, in *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255, Ieee, 2009.

[214] X.-S. Wei, J. Wu, and Q. Cui, *Deep learning for fine-grained image analysis: A survey*, *arXiv preprint arXiv:1907.03069* (2019).

[215] X. Wang and A. Gupta, *Unsupervised learning of visual representations using videos*, in *Proceedings of the IEEE international conference on computer vision*, pp. 2794–2802, 2015.

[216] K. He, X. Zhang, S. Ren, and J. Sun, *Deep residual learning for image recognition*, in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

[217] P. Khosla, P. Teterwak, C. Wang, A. Sarna, Y. Tian, P. Isola, A. Maschinot, C. Liu, and D. Krishnan, *Supervised contrastive learning*, *Advances in neural information processing systems* **33** (2020) 18661–18673.

[218] E. D. Cubuk, B. Zoph, D. Mane, V. Vasudevan, and Q. V. Le, *Autoaugment: Learning augmentation strategies from data*, in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 113–123, 2019.

[219] E. D. Cubuk, B. Zoph, J. Shlens, and Q. V. Le, *Randaugment: Practical automated data augmentation with a reduced search space*, in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pp. 702–703, 2020.

[220] T. Gao, X. Yao, and D. Chen, *Simcse: Simple contrastive learning of sentence embeddings*, in *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 6894–6910, 2021.

[221] Y. Zhang, R. Zhang, S. Mensah, X. Liu, and Y. Mao, *Unsupervised sentence representation via contrastive learning with mixing negatives*, in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2022.

[222] A. Wang, A. Singh, J. Michael, F. Hill, O. Levy, and S. R. Bowman, *Glue: A multi-task benchmark and analysis platform for natural language understanding*, in *International Conference on Learning Representations*, 2018.

[223] A. Warstadt, A. Singh, and S. R. Bowman, *Neural network acceptability judgments*, *Transactions of the Association for Computational Linguistics* **7** (2019) 625–641.

[224] E. Sheng and D. C. Uthus, *Investigating societal biases in a poetry composition system*, in *Proceedings of the Second Workshop on Gender Bias in Natural Language Processing*, pp. 93–106, 2020.