Interacting with Non-Cooperative User:
A New Paradigm for Proactive Dialogue Policy

Wenqiang Lei†
College of Computer Science
Sichuan University, China
wenqianglei@gmail.com

Yao Zhang∗
TKLNDST, CS
Nankai University, China
yaozhang@mail.nankai.edu.cn

Feifan Song
MOE Key Lab of Computational Linguistics
Peking University, China
songff@stu.pku.edu.cn

Hongru Liang
College of Computer Science
Sichuan University, China
lianghongru@scu.edu.cn

Jiaxin Mao
GSAI
Renmin University of China, China
maojiaxin@ruc.edu.cn

Jiancheng Lv
College of Computer Science
Sichuan University, China
lvjiancheng@scu.edu.cn

Zhenglu Yang
TKLNDST, CS
Nankai University, China
yangzl@nankai.edu.cn

Tat-Seng Chua
National University of Singapore
Singapore
chuats@comp.nus.edu.sg

ABSTRACT
Proactive dialogue system is able to lead the conversation to a goal topic and has advantaged potential in bargain, persuasion and negotiation. Current corpus-based learning manner limits its practical application in real-world scenarios. To this end, we contribute to advance the study of the proactive dialogue policy to a more natural and challenging setting, i.e., interacting dynamically with users. Further, we call attention to the non-cooperative user behavior — the user talks about off-path topics when he/she is not satisfied with the previous topics introduced by the agent. We argue that the targets of reaching the goal topic quickly and maintaining a high user satisfaction are not always converge, because the topics close to the goal and the topics user preferred may not be the same. Towards this issue, we propose a new solution named I-Pro that can learn Proactive policy in the Interactive setting. Specifically, we learn the trade-off via a learned goal weight, which consists of four factors (dialogue turn, goal completion difficulty, user satisfaction estimation, and cooperative degree). The experimental results demonstrate I-Pro significantly outperforms baselines in terms of effectiveness and interpretability.

CCS CONCEPTS
• Computing methodologies → Natural language processing; Discourse, dialogue and pragmatics; Interactive simulation.

1 INTRODUCTION

Proactive dialogue agent aims to lead the conversation with a user from the start topic (“Andy Lau”) to the goal topic (“Raging Fire”) through chatting with the user [27], as shown in Figure 1. This task has great potential in scenarios like bargain [8], persuasion [5, 25] and negotiation [12, 30]. Current solutions [2, 26, 27, 31, 35] follow the corpus-based learning setting — given a knowledge graph (KG), a goal topic and a dialogue context between two human (e.g., the leader and the follower in the DuConv corpus [27]), the agent is required to predict a topic of the next turn and generate a response based on this topic. However, turn-level policy might not align to the conversation-level policy well [6, 32]. Thus, we argue that the corpus-based learning setting is insufficient to meet the ultimate end that the agent is capable to chat with the user dynamically. In this work, we take one step further to scrutinize proactive dialogue policy in the interactive setting. This is a more natural but challenging setting where the agent is required to optimize long-term goal during the dynamic interaction.

Intuitively, the most straightforward solution for the policy would be a shortest path from the start to the goal, as presented in Figure 1 (b). However, the users may non-cooperatively introduce off-path topics. It usually happens when they are not satisfied with the current topic [20]. Take Figure 1 (c), as an example, when the user is unsatisfied with the previous topics introduced by the
agent, he/she talks about off-path topics ("Eighteen Springs" and "Leon Lai") in the 2\textsuperscript{nd} turn. Such non-cooperative user behavior has been seldom studied in prior efforts, but is crucial in the dynamic interaction. First, it can make the conversation out of the agent’s control — the agent is unable to introduce the next topic ("Benny Chan") in the currently planned shortest path and needs to find a new path to the goal topic. Second, the low satisfaction may hurt the user’s engagedness and may even lead the user to terminate the interaction session. This motivates us to pay attention to the non-cooperative user behavior and carefully manage user satisfaction during the conversation.

In this work, we aim to learn proactive dialogue policy in the interactive setting where we call attention to the non-cooperative user behavior. To advance this, we believe the agent should accomplish two targets — I) \textit{reaching the goal topic quickly} to make the conversation short and II) \textit{maintaining a high user satisfaction} to keep the user engaged during the conversation [27]. However, these targets cannot always converge. For example, in the 3\textsuperscript{rd} turn of Figure 1 (c), "Jacklyn Wu" is closer to the goal but "Almost A Love Story" may be more preferred by the user. Hence, the agent’s choice needs to take into account long-term issues — introducing the topic close to the goal is risky to trigger non-cooperative user behaviors, or, conversely, introducing the topic user preferred is likely to make the agent take longer turns to reach the goal. A good proactive agent is expected to achieve the balance between these targets.

Inspired by the aforementioned analyses, we build I-Pro, a novel proactive agent that can learn Proactive policy in the Interactive setting. Specifically, it employs the deep Q-learning algorithm [16] to train the dialogue policy function, optimizing the reward of faster goal arrival and higher user satisfaction. Towards the desired trade-off, we design a learned goal weight. It is derived from dialogue turn, goal completion difficulty, user satisfaction estimation, and cooperative degree. These four factors comprehensively decide which topic to be introduced in the next turn. In the experiment, we recur to user simulators to automatically interact with the agent because involving real users is laborious. Mimicking the user behavior, the simulator decides whether to be cooperative based on the quantitative satisfaction or to choose new topics based on their preference. We further give the simulator a personality character "tolerance" describing the variation of non-cooperative user behaviors. A high-tolerance simulator is less likely to behave non-cooperatively, while a low-tolerance simulator does the opposite. The experiments demonstrate the effectiveness and interpretability of the proposed solution over baselines. In addition, we observe two interesting insights. First, the agent distinctly tends to prioritize the target "reaching the goal topic quickly" when the number of dialogue turns is large. Second, users with different tolerance levels require different dialogue policies. In the face of low-tolerance users, the agent prefers to improve user satisfaction to keep he/she more engaged.

In summary, the main contributions of this work are as follows:

- We take the first step to scrutinize proactive dialogue policy in the natural interaction setting, where we call attention to cope with non-cooperative user behavior.
- We propose a novel model I-Pro that enables a trade-off between the targets of reaching the goal topic quickly and maintaining a high user satisfaction.
- Our grounded work can serve as a preliminary baseline, and the insightful analysis provides the potential of benefiting further research.
To investigate the proposed paradigm, we formalize a simple and practical interactive setting. We hope that this setting will serve as a good foundation for the current exploration and inspire more complex and realistic settings in the future.

Our setting refers to the scenario proposed by [27] where agent and user engage in a conversation based on topics in KG. Starting from a given topic, based on the KG, the agent introduces new topics to the user one by one, and then the user responses to the topic introduced by the agent in the current turn. The conversation ends when the goal topic is reached. Specifically, at the current turn, the agent will introduce a new topic adjacent to the one previously talked about to the user and the user will give comments on the topic. If the user is satisfied with the current conversation, he/she will behave cooperatively, giving positive comments on the topic introduced by the agent in the immediate previous turn, e.g., sounds great. However, if the user is not satisfied, then he/she will behave non-cooperatively, proactively mentioning new topics that the user prefers to increase his/her satisfaction [20], e.g., I recently watched a good movie, "Eighteen Springs" starring Leon Lai.

As this paper focuses on the dialogue policy for the goal leading and user satisfaction, we abstract the whole dialogue process as a topic-level interaction to simplify the problem, as shown in Figure 2. Specifically, we assume a third-party natural language understanding technique can perfectly detect the topics and user’s preference score corresponding to this topic. For example, in the 1st turn of the dialogue in Figure 2, the cooperative comments indeed he is in response to the topic "Andy Lau" can be extracted as \{(Andy Lau, p_1)\}. In the 2nd turn, the user utterance I recently watched a good movie, “Eighteen Springs” starring Leon Lai can be extracted as \{(Eighteen Springs, p_2), (Leon Lai, p_3)\}. p_1, p_2, p_3 are preference scores corresponds to “Andy Lau”, “Eighteen Springs”, and “Leon Lai”, respectively. In a similar way, we also assume a natural language generation component can perfectly generate an utterance given a chosen topic. For example, the topic “A Moment of Romance” can be generated as He has a movie called "A Moment of Romance" is very good. Therefore, the conversational policy only needs to focus on topic choosing. This perfectly isolates the dialogue policy from natural language understanding and generation, making the conversational agent is dedicated to policy learning.

Formally, the agent starts the conversation with topic \(e_n\), and leads the conversation to its goal topic \(e_g\). The background KG \(G = \{(e, r, e') | e, e' \in E, r \in R\}\) is defined as a set of triples with a topic (entity) set \(E\) and relation set \(R\). Given \(T\) dialogue turns, the dialogue history is defined as \(H_T = \{(e^1_t, u_1), (e^2_t, u_2), \ldots, (e^T_t, u_T)\}\), where \(e^i_t\) is the topic chosen by the agent and \(u_i\) refers to the user utterance at the \(i\)th turn. At the \(i\)th turn, given the goal topic \(e_g\) and the dialogue history \(H_{i-1}\), the agent chooses a new topic \(e_i\) from the candidate topic set \(C_t\). Following prior works [7, 17, 24], to ensure the coherence of the dialogue, the candidate topic set contains only the topics within one hop centered on the latest mentioned topic \(e_c\), i.e., \(C_t = \{e | (e, r, e_c) \in G\}\). The user utterance \(u_t = \{(e_{t,1}, \{p_{e_{t,1}}\}), \ldots, (e_{t,u_t}, \{p_{e_{t,u_t}}\})\}\) is the set of topic and preference pairs.

### 3 PROACTIVE DIALOGUE POLICY

With the proactive dialogue scenario setting, we propose a proactive dialogue policy I-Pro, which employs a learned goal weight to achieve the desired trade-off between the targets of reaching the goal topic quickly, and maintaining a high user satisfaction. At the \(t\)th turn, with the goal topic \(e_g\) and the dialogue history \(H_{t-1}\) as input, I-Pro chooses a topic \(e^i_t\) from the candidate topic set \(C_t\) as output. During the topic choosing process, I-Pro mainly tackles three issues: i) which topic is closest to the goal, ii) which topic the user prefers, and iii) how to make a trade-off between the two targets. Accordingly, I-Pro consists of a distance estimation module, a preference estimation module and a goal weight learning module.

#### 3.1 Distance Estimation

In order to lead the conversation to the goal, it is inevitable to clarify which topic is closest to the goal. The most straightforward method is to traverse the entire KG in advance to obtain the shortest distance between topics. However, KGs are usually large-scale
and continuously evolving, leading to high computational costs for global searches. We propose a soft distance estimation method. Specifically, if the distance between two topics is less than a pre-set distance limit \( D \), we perform accurate distance estimation; otherwise, we perform fuzzy distance estimation.

We take the distance estimation between topics \( e_i \) and \( e_j \) as an example to depict the soft distance estimation method, as shown in Figure 3. First, for the topic \( e_i (e_j) \), we search with it as the center within radius \( r \), where \( r = \frac{1}{2}D \), and collect the topics covered during the search as the set \( S_i \), \( S_j \). Then, if the two sets, \( S_i \) and \( S_j \), intersect, we perform accurate distance estimation: using the minimum value of the sum of the distances from the overlapping topics to the two centers. If the two sets are disjoint, we use a maximal value as the fuzzy distance estimate. The process can be formulated as:

\[
\text{ed}_{i,j} = \begin{cases} 
\min_{e_k \in S_i \cap S_j} (d_{i,k} + d_{j,k}), & S_i \cap S_j \neq \emptyset \\
\text{dMAX}, & S_i \cap S_j = \emptyset 
\end{cases} 
\]  

where \( \text{ed}_{i,j} \) is the estimated distance between \( e_i \) and \( e_j \), \( e_k \) belongs to the intersection of two sets \( S_i \) and \( S_j \); \( d_{i,k} \) and \( d_{j,k} \) are accuracy distances from the center topics, \( e_i \) and \( e_j \), to topic \( e_k \).

### 3.2 Preference Estimation

Which topic the user prefers is an inevitable consideration for agents working to improve user satisfaction. However, the agent can only obtain partially accurate preference information from the user through interaction. Therefore, we propose a preference estimation method to estimate unknown preferences based on the known ones.

According to the core assumptions in recommender systems [13, 19, 21], user preferences \( P \in \mathbb{R}^{|E|} \) can be viewed as the product of topic embeddings \( E \in \mathbb{R}^{|E| \times d} \) and the \( d \)-dimensional user vector \( u \in \mathbb{R}^d \), i.e., \( P = Eu \). We can obtain the user vector by solving this equation using the known preference information and its corresponding topics. At the \( t \)-th turn, we estimate the user vector by using the normal equation:

\[
\hat{u}_t = (E_{t-1}^T E_{t-1} + \beta I)^{-1} E_{t-1}^T P_{t-1},
\]

where \( \hat{u}_t \) is the estimated user vector \( 2 \), \( E_{t-1} \) and \( P_{t-1} \) are the topic embedding matrix and their corresponding user preference matrix for all topics that appear in the dialogue history \( H_{t-1} \). The topic embeddings are learned by the node embedding learning model \(^3\).

Consider the computational complexity for real-time interaction, we ensure that Eq. (2) is suitable for the case when the number of topics is small. When the number of topics is too large, we can use gradient descent to solve \( \hat{u}_t \).

Finally, we use the estimated user vector to estimate the unknown user preferences and obtain the user preferences:

\[
\hat{P}_t = P_{t-1} \odot (E \setminus E_{t-1}) \hat{u}_t,
\]

where the whole user preferences set \( \hat{P}_t \) contains the accurate user preferences and estimated user preferences.

### 3.3 Goal Weight Learning

This module is proposed to achieve the trade-off between the target I) reaching the goal topic quickly and target II) maintaining a high user satisfaction to keep the user engaged during the conversation. At the \( t \)-th turn, given the dialogue history \( H_{t-1} \) and the goal topic \( e_g \), the agent learns a goal weight \( g_w_t \) to denote the importance of the target I and \( 1 - g_w_t \) to denote the importance of the target II.

Then, the weighted score of each topic in the candidate topic set \( C_t \) is calculated:

\[
\text{Score}(e_{i,t}) = g_w_t \times \text{Rank}_{\hat{g}_d}(e_{i,t}) + (1 - g_w_t) \times \text{Rank}(e_{p_{t,i}}),
\]

where \( e_{i,t} \) is the \( i \)-th topic in \( C_t \), \( e_{i,t} \) is the estimated distance between \( e_{i,t} \) and the goal topic \( e_g \), \( e_{p_{t,i}} \) is the estimated user preference of \( e_{i,t} \). To eliminate the difference of magnitude between \( \text{ed}_{i,g} \) and \( \text{ep}_{t,i} \), we transform the raw \( \text{ed}_{i,g} \) value into its descending ranking and the raw \( \text{ep}_{t,i} \) value into its ascending ranking by the rank functions \( \text{Rank}_{\hat{g}_d}(\cdot) \) and \( \text{Rank}(\cdot) \), respectively. The topic with the highest score in \( C_t \) will be chosen as \( e_t^g \).

We abstract four factors from the dialogue history to blend into the goal weight \( g_w_t \). The specific factors are: i) dialogue turn and ii) goal completion difficulty, related to the target I; iii) user satisfaction estimation and iv) cooperative degree, related to the target II. Following are the details:

i) **Dialogue turn** \( t \) We expect the agent to lead the conversation to the goal as soon as possible. In that case, as the number of dialogue turns increases, so does the importance of achieving the goal. Thus, we introduce the dialogue turn \( t \) vector in the goal weight.

ii) **Goal completion difficulty** \( g_{cd} \) Another factor related to the target I is the difficulty of completing it. We approximate this difficulty by calculating the distance between the current talked topic \( e_c \) and the goal topic \( e_g \), i.e., \( g_{cd} = ed_{c,g} \). A high \( g_{cd} \) indicates that the current conversation is still far from the goal and that it will take a lot of effort to lead the conversation to the target. Conversely, the agent can easily complete the target I.

iii) **User satisfaction estimation** \( eus \) This factor directly portrays the time-realistic state of the target II. We estimate it by using information about user preferences that appears in user utterances. Specifically, we calculate the average preference value of all user utterances in the previous \( t - 1 \) turns to yield the estimated user satisfaction, i.e., \( eus_t = \frac{1}{t-1} \sum_{t'=1}^{t-1} \frac{1}{|u_t|} \sum_{j=1}^{|u_t|} ep_{e_{i,j}} \), where \( e_{i,j} \in u_t \)

\(^3\)https://github.com/thunlp/OpenNE
denotes the \( j \)-th topic mentioned by user at the \( i \)-th turn, and 
\( e_{p_{eij}} \in \tilde{P}_i \) denotes the preference of \( e_{ij} \).

iv) **Cooperative degree** \( cd \) This factor is learned from the historical user behaviors. At the \( t \)-th turn, we generate a one-hot sequence based on the user behavior in the previous \( t-1 \) turns, where \( 0 \) indicates that the user cooperate in the current turn, \( 1 \) indicates the user did not cooperate. Then we use GRU\([4]\) to encode this one-hot sequence to yield the cooperative degree \( cd \).

When \( cd \) is low, the user is easy to behave uncooperatively, so the agent should be careful to avoid triggering uncooperative user behavior.

Finally, we concatenate four factors and adopt a 2-layer policy MLP to learn the goal weight:

\[
gw_t = MLP(t @ cd_t @ eus_t @ cd_t). \tag{5}
\]

### 3.4 Deep Q-Network Learning

We model the process by which an agent chooses a new topic based on a dialogue policy as a Markov Decision Process (MDP). In details, as a RL problem, at the \( t \)-th turn, \((s_t, a_t, p_t, r_t)\) in MDP are defined as: 1) State \( s_t \) is equal to the dialogue history \( H_{t-1} \) with the goal topic \( e_{\gamma} \); 2) Action \( a_t \) is equal to the candidate topic set \( C_t \); 3) Transition \( p_t \) is the transition function with \( p(s_{t+1} | s_t, e_t^a) \) being the probability of seeing state \( s_{t+1} \) after taking action \( e_t^a \) based on \( s_t \); and 4) Reward \( r_t \) is the reward that the agent obtains from the environment, including rewards reflecting changes in goal completion and user satisfaction. Our goal is to learn a dialogue policy which can maximize the cumulative reward over the whole conversation process as:

\[
θ^* = \arg \max_{θ \in U} E \left[ \sum_{t=1}^{T} r(s_t, e_t^a) \right], \tag{6}
\]

where \( θ^* \) is the optimal parameter set of I-Pro and \( r_t = r_{as} + r_{quit} + r_{goal} + r_{suc} + r_{fail} \). Specifically, \( r_{suc} \) is the change of the user satisfaction, i.e., \( α(US_t - US_{t-1}) \), where \( α \) is the magnification factor; \( r_{quit} \) is a strongly negative reward if the user quits the conversation; \( r_{goal} \) is the change of the goal completion degree, i.e., \( e^{-\lambda}(d_t - d_{t-1}) \), where \( \lambda \) is the time decay coefficient, \( d_t \) denotes the distance between the current topic at the \( t \)-th turn and the goal topic; \( r_{suc} \) is a strongly positive reward if the conversation reach the goal topic; and \( r_{fail} \) is a strongly negative reward if the conversation do not reach the goal topic before the maximum turn limit \( T \).

We adopt Q-learning algorithm to train I-Pro. We use the score calculated by Eq. 4 as the Q-value \( Q_θ(s_t, e_t^a) \). Based on the optimal Bellman Equation [22], we can represent the optimal Q-value with the the maximum expected reward achievable is:

\[
Q_θ(s_t, e_t^a) = \mathbb{E} \left[ r_t + γ \max_{e_{t+1} \in a_{t+1}} Q_θ(s_{t+1}, e_{t+1}^a) \right]. \tag{7}
\]

We improve the value function \( Q_θ(s_t, e_t^a) \) by adjusting \( θ \) to minimize the mean-square loss function, defined as follows:

\[
l(θ) = \mathbb{E}_{(s_t, a_t, s_{t+1}, r_t) \sim \mathcal{M}} \left[ (Q_θ(s_t, e_t^a) - Q_θ(s_t, a_t))^2 \right]. \tag{8}
\]

### 4 EXPERIMENTS

The key contributions of this work are on studying the proactive dialogue policy interactively and the design of a proactive dialogue policy I-Pro. We first conduct a comparison experiment to evaluate the effectiveness of I-Pro interactively. Then we explore the impact of different types of users on the proactive dialogue policy. In addition, we perform ablation study to investigate the effect of each factor in the goal weight learning. Finally, we conduct a case study to demonstrate the superiority of I-Pro. Specifically, we have the following research questions (RQs) to guide experiments:

RQ1: How is the overall performance of I-Pro comparing with existing dialogue policies?

RQ2: How do different types of users affect the proactive dialogue policy?

RQ3: How do different factors affect the goal weight learning?

### 4.1 User Simulation

Considering the huge costs associated with involving humans in the interaction, we design user simulators to complete the process of interacting with the agents when conducting the experiments. The design of the simulator is also at the topic-level. When interacting with the agent, at the \( t \)-th turn, the simulator accepts the dialogue history and the topic introduced by the agent at the current turn, i.e., \( H_{t-1} \cup \{e_t^a\} \). Then, it responds \( u_t \) to the agent or quits the conversation.

Here, we describe in detail how to generate \( u_t \). The user utterance is based on satisfaction with the current conversation. We argue that satisfaction can be formalized as the cumulative average of users’ preferences for the topics covered by the conversation:

\[
US_t = \frac{1}{\sum_{i=1}^{T} |\{s_i\}|} \sum_{j=1}^{T} p_{e_{ij}} + p_{e_t^a}, \tag{9}
\]

where \( US_t \) denotes the user satisfaction at the \( t \)-th turn, \( e_{ij} \in u_t \) denotes the \( j \)-th topic mentioned by user at the \( i \)-th turn and \( p_{e_{ij}} \) denotes the user preference to the topic \( e_{ij} \); \( |u_t| \) is the total number of topic mentioned in the \( t \) turn utterance (without removing duplicates). Note that when \( e_{ij} = e_t^a \), there is no need to double count the latter \( e_t^a \). User preferences are personalized. For one user simulator, we first sample a user vector \( u \in \mathbb{R}^d \). Each element in the user vector is sampled from the Gaussian distribution \( N(0, 2) \). Next, following the core assumptions in recommendation techniques [13, 19, 21], we map the user vector to the topic embeddings \( E \in \mathbb{R}^{|E| \times d} \) to calculate user preferences, i.e., \( P = Eu \), where \( P = \{p_e|e \in E\} \).

At the \( t \)-th turn, based on the calculated user satisfaction, the user utterance can be generated in cases:

\[
u_t = \begin{cases} \{e_t^a, P_{e_t^a}\} & \text{if } US_t > Q_c, \\ \{e, p_e \} & \text{if } Q_c < US_t \leq Q_c, \\ \emptyset & \text{if } US_t \leq Q_c, \end{cases} \tag{10}
\]

specifically, the user behavior can be deconstructed into three types: cooperative, non-cooperative and quit. If \( US_t \) is above the cooperative threshold \( Q_c \), the simulator will only respond the topic introduced by agent at this turn and the corresponding preference. Whereas if \( US_t \) is below the cooperative threshold \( Q_c \), the simulator will respond new topics with their preferences. The new topic in
Table 1: GCR(%) and US(%) performance (± standard deviation) of compared dialogue policies (pairwise t-test at 5% significance level). All dialogue policies are evaluated on three types of user simulators with different tolerance values (k = 0.8, 1.0 and 1.2) as well as on a mixed-user simulator.

|                  | Mixed User Simulator | User Simulator (k = 0.8) | User Simulator (k = 1.0) | User Simulator (k = 1.2) |
|------------------|----------------------|--------------------------|--------------------------|--------------------------|
|                  | GCR  | US      | GCR  | US      | GCR  | US      | GCR  | US      |
| Random           | 1.29 ±0.01 | 52.86 ±0.01 | 1.67 ±0.01 | 55.29 ±0.01 | 1.18 ±0.01 | 52.65 ±0.01 | 1.02 ±0.01 | 50.65 ±0.01 |
| Pop (GCR)        | 57.00 ±0.03 | 53.52 ±0.00 | 48.18 ±0.03 | 56.05 ±0.00 | 59.18 ±0.03 | 53.24 ±0.00 | 63.63 ±0.02 | 51.27 ±0.00 |
| Pop (US)         | 1.16 ±0.01 | 62.14 ±0.00 | 1.39 ±0.01 | 64.47 ±0.00 | 1.07 ±0.01 | 61.80 ±0.00 | 1.03 ±0.00 | 60.15 ±0.00 |
| NICF             | 3.46 ±0.01 | 54.48 ±0.00 | 3.55 ±0.01 | 57.76 ±0.00 | 3.30 ±0.01 | 54.10 ±0.00 | 3.52 ±0.01 | 51.58 ±0.01 |
| DeepPath         | 3.84 ±0.01 | 54.41 ±0.00 | 4.32 ±0.01 | 57.94 ±0.00 | 3.81 ±0.01 | 54.06 ±0.00 | 3.38 ±0.01 | 51.24 ±0.00 |
| N KD             | 3.15 ±0.01 | 52.96 ±0.00 | 4.19 ±0.02 | 56.45 ±0.01 | 2.97 ±0.01 | 52.31 ±0.00 | 2.28 ±0.01 | 50.13 ±0.00 |
| I-Pro            | 29.74 ±0.07 | 59.28 ±0.01 | 22.47 ±0.06 | 61.91 ±0.01 | 32.20 ±0.07 | 58.64 ±0.01 | 34.54 ±0.08 | 57.28 ±0.01 |

$u_t$ is chosen with its user preference as probability. We design the function $Sel(e)$ to calculate whether the topic $e$ is chosen or not:

$$Sel(e) \sim \text{Bernoulli}(p_e),$$

where the $Sel(e)$ is sampled from the Bernoulli distribution: 1 indicates the topic is chosen, while 0 indicates the opposite. If $US_f$ fall below an even lower quit threshold $Q_f$, the simulator will quit the conversation in advance.

Finally, to further imitate humans, we personalize the thresholds $Q_c$ and $Q_q$ as well. We design a new characteristic “tolerance”. A high tolerance simulator is less likely to behave non-cooperatively or to quit a conversation outright, while a low tolerance simulator does the opposite. Concretely, the tolerance character $k$ is used as a shirker on the thresholds. That is, we use $kQ^c_e$ and $kQ^q_q$ as the personalized cooperative and quit thresholds, where $Q^c_e$ and $Q^q_q$ are pre-set constants.

In the evaluation process, for one start-goal topic pair, we design three user simulators with the same user vector and different tolerance characteristics, low ($k = 0.8$), medium ($k = 1.0$) and high ($k = 1.2$), respectively. We choose 0.5 as $Q^c_e$ and 0.4 as $Q^q_q$, experimentally. Considering the computational consumption caused by large-scale KG, we set the topic selection scope of the simulator to all neighbors within 3 hops of the current topic, while $|u_t|$ is limited to maximum 3.

4.2 Settings

4.2.1 Dataset. The dataset we used is collected from KdConv [33], which contains a film-related KG and a large amount of multi-turn conversation data. First, we continue to use the KG of KdConv, which has 7477 entities, 4939 relations and 89618 triples. Then, because this work focuses on studying the dialogue policy in an interactive manner, we abandon the static conversation data, and extract only the start-goal topic pairs from the multi-turn conversation data. Specifically, the start topic is extracted from the 1st turn and the goal topic is extracted from the last turn of the conversation. Statistically, the KG has 7477 film-related topics. The training and testing sets contain 353 and 89 start-goal topic pairs, respectively.

4.2.2 Evaluation Metrics. With the targets I and II, we use two metrics to automatically evaluate the all-turn performance of the dialogue policy:

Goal Completion Rate (GCR) We use GCR to evaluate the ability of the policy to lead the conversation successfully. GCR is defined as:

$$GCR = \frac{1}{N} \sum_{k=1}^{N} e^{-\lambda T_k} D_k$$

where $N$ is the total number of dialogues, $T_k$ is the total number of turns of the $k$-th dialogue, $\lambda$ is the time decay coefficient. $D_k$ is set to 1 if the $k$-th dialogue succeeded in achieving the agent’s goal, otherwise it is 0.

User Satisfaction (US) We use US to evaluate the ability of the policy to maintain the high user satisfaction. We use the average preference of all the topics involved in the whole conversation to represent US:

$$US = \frac{1}{N} \sum_{k=1}^{N} \frac{1}{T_k} \sum_{i=1}^{T_k} \frac{1}{|u_t|i+1} \left( \sum_{j=1}^{N} p_{e_{k,i,j}} + p_{e^a_{k,i,j}} \right)$$

where $p_{e_{k,i,j}} \in u_{k,i}$ is the preference of the $j$-th topic mentioned by user at the $i$-th turn in the $k$-th dialogue, and $p_{e^a_{k,i,j}}$ is the preference of the topic introduced by the agent at the $i$-th turn in the $k$-th dialogue. $T_k$ is the total number of dialogue turns for the $k$-th conversation. Note that when $e_{k,i,j} = e^a_{k,i,j}$, there is no need to double count the latter $e^a_{k,i,j}$.

4.2.3 Training Details. The dimensions of the user vector and topic embedding are all set to 50. The node embedding learning model is DeepWalk [18]. The maximum dialogue turn $T$ is set to 20. The distance limit $D$ used in the distance estimation is set to 6. The optimizer chosen is the Adam and we empirically set the learning rate as $1e^{-7}$. The maximum training epoch is set to 100. The RL algorithm used here is DQN[16], in which the memory size, batch size, decay factor $\gamma$, exploration factor $\epsilon$ for the $\epsilon$-greedy policy, time decay coefficient $\lambda$, and the magnification factor $\alpha$ are 2000, 100, 0.9, 0.2, 0.02 and 100, respectively. Rewards $r_{\text{quit}}, r_{\text{suc}}, r_{\text{fail}}$ are $-10, 20$ and $-10$ if triggered, respectively; otherwise they are all 0.
Figure 5: Comparison performance of I-Pro and the Variant I-Pro-degrade. The optimal results for I-Pro and I-Pro-degrade are depicted. Apparently, the optimal results of I-Pro completely surround the optimal results of I-Pro-degrade.

Figure 6: Performance w.r.t. different user simulator types. Performance on three user simulator types with different tolerance: low ($k = 0.8$), medium ($k = 1.0$) and high ($k = 1.2$) are visualized. GCR increases and US decreases as the tolerance level increases.

For an in-depth and extensive evaluation, we test all comparison dialogue policies for 100 rounds. Due to a certain randomness of user behavior, there is a slight variation in the results of 100 rounds. We expect to select the best result for reporting. However, under the multi-metric evaluation setting, the two metrics we use (GCR and US) are equally important, so the best results are often a set. In this set, no one result can outperform the other results on both metrics at the same time. Therefore, we report the average of all results in the optimal results set as the final performance for each policy.

4.2.4 Baselines. This work is the first attempt to study a proactive dialogue policy which has the ability to quickly achieve the goal topic as well as maintain high user satisfaction. We first present three baselines to perceive the difficulty of this task:

- **Random** At each turn, the agent randomly selects the topic.
- **Pop(GCR)** At each turn, the agent chooses the topic closest to the goal, i.e., with the smallest estimated distance, in order to get the highest GCR.
- **Pop(US)** At each turn, the agent chooses the user’s favorite topic, i.e., with the highest estimated preference, in order to get the highest US.

| $k$ | 0.4 | 0.6 | 0.8 | 1.0 | 1.2 | 1.4 | 1.6 |
|-----|-----|-----|-----|-----|-----|-----|-----|
| GCR | 15.84 | 17.04 | 22.47 | 32.20 | 34.54 | 34.1 | 34.75 |
| US  | 62.37 | 62.71 | 61.91 | 58.64 | 57.28 | 56.24 | 55.99 |
| $gw$ | 0.37 | 0.41 | 0.45 | 0.49 | 0.50 | 0.51 | 0.51 |
| ± sd | 0.17 | 0.17 | 0.17 | 0.13 | 0.11 | 0.10 | 0.09 |

Table 2: Performance (GCR and US) and the learned goal weight value (± standard deviation) at different tolerance values.

Considering that there are currently no dialogue policies that address both targets, we chose several policies that are committed to one target alone as the baselines.

- **NICF** [36] This model is originally designed to recommend items for user in an interactive setting. In our proactive dialogue scenario, it can be used to enhance US during multi-turn interactions.
- **DeepPath** [28] This model is focus on searching a path in KG through multi-hop reasoning from the start entity to the goal entity. Here, it can work on leading the conversation to the goal topic.
- **NKD** [15] This model works on multi-turn dialogue generation based on the dialogue history and KG. We degrade it to the topic level dialogue generation.

All the above baselines and our I-Pro are trained and tested in an interactive manner with our designed user simulators.

4.3 Overall Performance (RQ1)

**I-Pro vs. Baselines** Table 1 demonstrates the performance of I-Pro and baselines. As can be clearly seen, I-Pro significantly outperforms existing dialogue policies both on GCR and US. We also make the following observations: 1) All policies show a decrease in the US metric as tolerance increases, because users with low tolerance are more likely to be triggered to quit the conversation at a higher satisfaction level. 2) The Random baseline shows a decreasing trend under the GCR metric as user tolerance increases. When the user tolerance is low, there is more non-cooperative behavior from the user. And the perturbation caused by the user’s non-cooperative behavior on the conversation forward direction makes the conversation have a chance to reach the goal topic. 3) Two “Pop” baselines of our own design show the upper bounds of the effects currently achievable on both GCR and US metrics. These two upper bounds can be further improved by improving the estimation accuracy of user preferences and the distance. Interestingly, under the GCR metric, the Pop (GCR) tends to increase as tolerance increases, while the Pop (US) does the opposite. We speculate that this is because the Pop (GCR) aims to improve GCR, so the lower the user tolerance, the more likely it is to trigger non-cooperative behavior. While, Pop (US), same as Random, has a higher probability of the agent walking to the goal by chance at low user tolerance. 4) Three existing works, NICF [36], DeepPath [28] and NKD [15], all underperform, exposing the challenges in proactive dialogue policy study. We suspect that their drawbacks are the inaccurate estimation of
user preferences and distance to the goal, as well as a lack of ability to make the trade-off between target I and II.

**I-Pro vs. I-Pro-degrade** We propose one variant of I-Pro, namely I-Pro-degrade, to further explore the utility of our proposed goal weight. In this variant, the goal weight is degraded to one trainable parameter, i.e., $\text{Score}(e_{t,i}) = \beta \times \text{Rank}(ed_{i,t}) + (1 - \beta) \times \text{Rank}(ep_{t,i})$. We intuitively present the performance comparison in Figure 5. Clearly, the optimal results of I-Pro completely surround the optimal results of I-Pro-degrade. This exhibits the shallow capability of one parameter to make the trade-off between two targets. We also observe that the effect of I-Pro-degrade has many repeated values, which shows the inflexible ability of only using one parameter. Apparently, it is intractable for one parameter to perceive the progress of the conversation and the personality of the user. This suggests that our integration of multiple factors on the goal weight learning is necessary.

### 4.4 Performance w.r.t. Different User Types (RQ2)

To investigate the differences in the effects of I-Pro on different types of users, we evaluate I-Pro on three user simulator types with different tolerance: low ($k = 0.8$), medium ($k = 1.0$) and high ($k = 1.2$), and visualize the results in Figure 6. Obviously, the difference in tolerance leads to a significant difference in performance: GCR increases and US decreases as the tolerance level increases. Users with low tolerance are more likely to behave non-cooperatively when their satisfaction is low, making it difficult for the agent to achieve the goal topic. In addition, agents choose users’ preferred topics in order to try to avoid non-cooperative user behaviors. Therefore, low tolerance users will lead to a low GCR and a high US. High tolerance users are just the opposite.

We further explore more diverse tolerance values and the correlation between the tolerance values and the learned goal weight. Here, for the goal weights, we use their averages over all dialogue turns. Results are reported in Table 2. We observe that the goal weight is perceptible to changes in tolerance and there is a positive correlation between the two terms. When the goal weight perceives that the user has a high tolerance, it will spontaneously be more biased towards improving GCR. We also observe that the standard deviation of the goal weight is negatively correlated with the tolerance.

This is because the less tolerance the user has, the more drastic the change in user behavior (cooperative or non-cooperative) will be. The goal weight reacts to the perceived change as a larger standard deviation.

### 4.5 Ablation Study on Goal Weight (RQ3)

We abstract four factors from the dialogue history to learn the goal weight. Here, we first analyze the effectiveness of different factors in goal weight learning. As shown in Figure 7, we evaluate the performance on our two variants: 1) the goal weight learning without two target I-related factors: dialogue turn factor and goal completion degree factor, and 2) the goal weight learning without two target II-related factors: user satisfaction estimation factor and cooperative degree factor. We observe that, when deactivating the target I-related factors, the GCR score decreases significantly and the US score increases slightly. This confirms that when the agent is unable to perceive the progress of the conversation, its control over the conversation decreases, and the conversation is easily led to the user’s preferred topic by the user’s non-cooperative behaviors. This eventually manifests as an increase in user satisfaction, but the conversation fails to achieve the goal topic. When deactivating the target II-related factors, both the US and GCR scores decrease. This suggests that the conduct of proactive conversations is sensitive to user satisfaction and behavior. So, it is critical to estimate the satisfaction of current users as well as to predict the cooperative degree of users. When deactivating the user satisfaction estimation factor, the agent cannot perceive user dissatisfaction in time, leading to a decrease in US. In addition, when the agent cannot perceive the cooperative degree of users, more non-cooperative user behaviors are triggered, leading to a decrease in GCR.

We secondly analyze the specific correlation between each factor and the goal weight. The correlations are pulled from the dialogue data collected during the evaluation, which are visualized in Figure 8. We make the following observations:
Figure 9: Case study of four dialogue policies: Pop (GCR), Pop (US), I-Pro-degrade and I-Pro. We choose the same start and goal topics, and record the interaction process between each dialogue policy and the user simulator whose tolerance is 1. We report partial transcripts of the conversations, including turns 1, 2, 14, and 15. Specifically, at each turn, we report the conversation detail, the current user satisfaction ($U_S$) and the distance to goal ($d_t$). The finally results also are reported. Specially, we also report the goal weight values of I-Pro-degrade and I-Pro, i.e., $\beta$ and $g_{\text{wt}}$. Due to the space limitation, we show here only the English translation of the cases.

### 4.6 Case Study

In this section, a case study was conducted to better demonstrate the superiority between our proactive dialogue policy and other compared dialogue policies. Figure 9 shows four dialogue fragments of I-Pro and three baselines: Pop (GCR), Pop (US) and I-Pro-degrade, interacting with a user simulator whose tolerance is 1.0, as well as the final results containing the GCR and US scores. We obtain the following observations:

i) Pop (GCR) baseline works to improve the GCR score solely, so it chooses the topic closest to the goal in each turn. However, due to the agent ignores the user satisfaction, the user behaves non-cooperatively in almost every turn. Each non-cooperative behavior again takes the conversation off-goal, resulting in the agent’s efforts being wasted.

ii) Pop (US) baseline works to improve the US score and therefore introduces the user’s favorite topics in each turn. High user satisfaction is maintained during the conversation, resulting in fewer non-cooperative user behaviors. Due to the neglect of the target $I$, the GCR score is low at the end of the conversation, despite maintaining a high user satisfaction.

iii) In I-Pro-degrade, the goal weight is a fixed value (after training), i.e., $\beta = 0.54$. The $\beta$ value greater than 0.5 means that the agent will always be partial to the goal achievement in the conversation. As we can see, in the first turn, when the user satisfaction is significantly low, the agent still chooses one topic that is close to the goal rather than preferred by the user. Therefore, this decision triggers the non-cooperative user behavior in the
next turn. Obviously, it is difficult to intelligently balance user satisfaction and goal completion using only one parameter.

iv) I-Pro is significantly more flexible and strategic than the three baselines. It is able to choose to favor US when user satisfaction is low (e.g., $g_{US} = 0.07$), while favoring GCR at a high US and later stages of the conversation (e.g., $g_{US} = 0.55$).

5 RELATED WORK

Conventional conversational systems are dedicated to assisting users to accomplish goals or achieve users’ satisfactions. For example, task-oriented systems, always as virtual personal assistants, are designed to help users with daily tasks, such as booking accommodation and restaurant [10, 14, 29] conversational recommendation systems [9, 11] can recommend products to users that they want; and human-like chatbots can interact with users to provide reasonable responses for entertainment [1, 3].

In recent years, however, there has been growing interest in exploring on another feature of dialogue systems: proactivity. Proactive dialogue systems aims to lead the conversation to their own goals, which may not perfectly align well with the user’s goal or satisfaction. Proactive dialogue systems have great potential in various scenarios. For example, persuasion dialogue system attempts to convince the user to change his/her attitude, opinion, or behavior. Wang et al. [25] collect a large dialogue dataset in which a participant persuaded another participant to make a donation. Tian et al. [23] build a classifier to predict users’ policies for resisting donations. Negotiation dialogue system interacts with a user strategically to reach an agreement. He et al. [8] propose a modular approach based on coarse dialogue acts that decouples policy and generation to make the policy controllable. Zhou et al. [34] model both semantic and policy history to improve both dialogue policy planning and generation performance. However, the proactive policy studied in the aforementioned works lack explainability hence helping less on revealing the rationale of proactive dialogues.

To gain better explainability and empower strategic reasoning for proactive dialogue, Wu et al. [27] introduce a knowledge graph and formalize the proactive policy as a path reasoning problem over the KG. They contribute a corpus through crowdsourcing where each participant persuaded another participant to make a donation. An et al. [21] build a classifier to predict users’ policies for resisting donations. Negotiation dialogue system interacts with a user strategically to reach an agreement. He et al. [8] propose a modular approach based on coarse dialogue acts that decouples policy and generation to make the policy controllable. Zhou et al. [34] model both semantic and policy history to improve both dialogue policy planning and generation performance. However, the proactive policy studied in the aforementioned works lack explainability hence helping less on revealing the rationale of proactive dialogues.

In this paper, we study the proactive dialogue policy in an interactive manner and call attention to the non-cooperative user behavior during the conversation. We argue that the interactive proactive dialogue policy learning has two targets: leading the conversation to the goal quickly and maintaining a high user satisfaction. To advance

6 CONCLUSION

In this work, we study the proactive dialogue policy in an interactive manner and call attention to the non-cooperative user behavior during the conversation. We argue that the interactive proactive dialogue policy learning has two targets: leading the conversation to the goal quickly and maintaining a high user satisfaction.
