Research Article

AND DATA SCIENCES
BULLETIN OF COMPUTER

Persuasive Dynamic Goal-Driven Virtual Agents for Infeasible Task-Oriented Dialogues

Pushpak Bhattacharyya and Sonia Shahzadi

Dept. of Computer Science and Engineering, Indian Institute of Technology Patna, Patna, Bihar, India

Abstract

Task-oriented dialogue systems have recently begun to support *dynamic* user goals, allowing agents to react to evolving preferences and sentiment shifts during a conversation. However, even state-of-the-art dynamic goal-driven agents (e.g., DGDVA) remain largely reactive: when a user updates their goal to something *infeasible* given domain constraints (such as requesting flagship specifications at a budget price), these systems can detect the mismatch but cannot *actively* guide the user toward a realistic alternative. In this paper, we introduce **PDG-DVA** (Persuasive Dynamic Goal-Driven Virtual Agent), a dialogue framework that couples dynamic goal adaptation with *computational persuasion* to salvage conversations that would otherwise fail. We formalize the problem as an augmented POMDP whose state includes a persuasion state (activation flag, chosen strategy, rejection count, and estimated user flexibility), and we extend the action space with persuasion-specific actions such as compromise suggestion and value reframing. A tripartite reward—covering task completion, sentiment alignment, and successful persuasive redirection—encourages the agent to solve feasible tasks normally while invoking persuasion only when infeasibility is detected. To support training and evaluation, we release **DevVA-P**, a 250-dialogue extension of the DevVA dataset annotated with infeasible goal turns, applicable persuasion strategies, and acceptance outcomes. Experiments on DevVA-P show that PDG-DVA substantially improves persuasion acceptance rate and average return over strong baselines, including the original DGDVA, a random-persuasion variant, and a rule-based persuader, while human evaluation confirms that users perceive the system as more helpful and intelligent without sacrificing naturalness.

Keywords: Persuasive, DGDVA, PDG-DVA, DevVA-P, Persuasive Dynamic

1. Introduction

Task-oriented dialogue systems have evolved from rigid slot-filling pipelines to highly adaptive conversational agents capable of operating across multiple domains such as e-commerce, technical support, healthcare, and travel booking [1, 2]. This evolution has been fueled by advances in natural language understanding, reinforcement learning, and large pre-trained language models, enabling agents to interpret user intent, track dialogue state, and complete goals with increasing robustness. A central assumption in many of these systems, however, is that user goals are either fully specified at the beginning of the interaction or can be incrementally clarified but remain *feasible* within the constraints of the underlying knowledge base or product catalog. In practice, human–machine conversations

Corresponding author: Pushpak Bhattacharyya (pb@cse.iitb.ac.in)

BCDS 2022, Vol. 3(2), 22-37. DOI: 10.71448/bcds2232-3

are rarely so tidy: users revise, relax, or even contradict their goals mid-conversation due to new preferences, budget revelations, or sentiment changes.

To address this, Tiwari [3] introduced the Dynamic Goal-Driven Virtual Agent (DGDVA), which augments traditional task-oriented dialogue systems with *dynamic goal adaptation*. Instead of treating the user goal as static, DGDVA continuously monitors user sentiment and detects *goal–system discrepancies*—for example, when the system cannot satisfy the user's newly expressed constraints. This line of work is important because it shifts the focus from one-shot goal satisfaction to *conversational continuity*: the agent tries to keep the dialogue productive even when the user changes course. Yet, despite this progress, DGDVA still operates within a fundamentally *reactive* paradigm. It detects that something has gone wrong, but its options for repairing the dialogue are narrow.

A particularly problematic scenario arises when the user updates their goal to something that is simply *infeasible* given the domain constraints—for instance, requesting a flagship smartphone with top-tier memory and storage but at an entry-level price point. Existing dynamic agents can correctly detect the mismatch and even register the accompanying negative sentiment, but they typically terminate, backtrack, or offer only generic apologies because they cannot *proactively steer* the user toward a realistic alternative. This behavior stands in stark contrast to human experts in sales or customer service, who routinely manage such situations by *persuading*, *reframing*, or *trade-off recommending* to preserve user satisfaction and task success.

1.1. Research Gap and Motivation

The core gap we address is the *passive nature* of current dynamic goal adaptation mechanisms. Existing systems are good at *noticing* that the user is unhappy or that the requested item does not exist, but they are not good at *doing something constructive* about it. In real-world, high-stakes conversational settings—product configuration, subscription upgrades, travel rebooking, or even university admissions counseling—the ability to negotiate constraints, justify alternatives, and suggest near-optimal options is often the difference between a failed and a successful dialogue.

Moreover, persuasive communication in dialogue is not merely "selling." It involves aligning with the user's stated priorities, acknowledging constraints, and offering a reasoned path to a solution. Consider a user who says: *"I need 8GB RAM, 512GB storage, but my maximum budget is $300."* A DGDVA-like agent would correctly detect negative sentiment once it learns that 512GB phones start at $600, but it would ultimately fail because no device meets the hard constraints. A *persuasive* agent, however, could respond along the following lines: *"I understand storage is important to you. Phones with 512GB typically start around $600, but there's a model with 256GB plus expandable SD storage for $280—this gives you nearly the same benefit within your budget."* This response does three things simultaneously: it validates the user's main need (storage), it explains the constraint (512GB is expensive), and it proposes a viable alternative (256GB + expansion) without abandoning the task. This is the paradigm shift we pursue: from reactive discrepancy handling to **active, strategy-driven guidance**.

From a research standpoint, this raises several open questions: How can we integrate computational persuasion into a task-oriented dialogue pipeline without breaking the goal-tracking logic? How can we select the *right* persuasive strategy (e.g., cost–benefit framing, compromise suggestion, feature prioritization) given the live dialogue context? And how can we train or evaluate such systems so that persuasion is rewarded, but user sentiment and task completion are still primary objectives?

1.2. Contributions

To address these challenges, we introduce the **Persuasive Dynamic Goal-Driven Virtual Agent (PDG-DVA)**, a dialogue framework that explicitly couples dynamic goal adaptation with context-aware persuasive reasoning. Our contributions are as follows:

- We propose PDG-DVA, the first dialogue system (to our knowledge) that integrates *computational persuasion* directly into a dynamic goal-driven agent. Unlike prior systems that stop at detecting infeasibility, PDG-DVA is able to *reconstruct* a realizable goal path by suggesting, justifying, or negotiating alternatives.

- We design a modular Persuasion Module that houses a bank of persuasive strategies—such as budget–feature trade-off, near-match recommendation, incremental upgrade suggestion, and constraint relaxation prompts—and a context-aware selector that chooses among them based on user sentiment, detected discrepancy type, and dialogue history. This module plugs into the existing Goal-Driven Module without requiring a redesign of the full dialogue manager.

- We introduce a tripartite reward function for training, which (i) preserves the original task-completion objective, (ii) maintains sentiment alignment by rewarding responses that reduce user frustration, and (iii) *explicitly* rewards successful persuasive redirection when the original goal is infeasible. This encourages the agent not only to finish tasks but also to salvage dialogues that would otherwise fail.

- We construct and release **DevVA-P**, an extension of the DevVA dataset, containing 250 professionally annotated dialogues in which the user's goal is partially or wholly infeasible and the agent is expected to employ a persuasive strategy. Each dialogue is labeled with discrepancy type, sentiment cues, and the applied persuasive tactic, enabling supervised and RL-based evaluation.

- Through extensive experiments on both standard task-oriented dialogues and our infeasible-goal subset, we show that PDG-DVA substantially improves dialogue success rate, user-satisfaction proxies, and recovery from infeasible states, while retaining competitive performance on scenarios where persuasion is not required.

Taken together, these contributions move task-oriented dialogue systems closer to human-like assistance: instead of giving up when user goals and domain constraints collide, the agent learns to *negotiate the space in between*—maintaining rapport, preserving task progress, and delivering a solution that is acceptable to the user even under imperfect conditions.

2. Related Work

The problem we study sits at the intersection of three strands of dialogue research: (i) task- and goal-oriented conversational systems that perform slot filling and database grounding, (ii) computational persuasion and sentiment-aware dialogue management, and (iii) strategic/negotiation-style conversations where agents must steer users toward acceptable outcomes. Prior work has made strong progress on each strand in isolation, but, to our knowledge, none has explicitly combined *dynamic goal adaptation* with *persuasive redirection* for *infeasible* task scenarios—the setting motivated in our introduction.

2.1. Dynamic Goal-Oriented Dialogue Systems

Early task-oriented dialogue systems [2, 4, 5] were built around the assumption of a *known and realizable* user goal. A typical pipeline included natural language understanding (NLU) to extract slots, dialogue state tracking (DST) to maintain beliefs, and a policy module to decide system actions, often trained with supervised learning or reinforcement learning. As long as the user's constraints matched something in the underlying knowledge base, the system could complete the task.

However, practical deployments revealed that users often *revise* their goals mid-conversation—changing the budget, adding a brand preference, or tightening a temporal constraint. This made purely static state-tracking insufficient. Tiwari [3] addressed this by proposing the Dynamic Goal-Driven Virtual Agent (DGDVA), which introduced sentiment-aware *discrepancy detection*: the agent monitors the user's affect and linguistic cues to detect when the current plan no longer aligns with the user's emerging preferences. This was an important conceptual shift because it treated user sentiment as a *feedback channel* for goal alignment, enabling real-time goal updates instead of assuming that users always state their final goal up front.

Despite this advance, DGDVA and similar systems remain fundamentally *reactive*. They can detect that the user is dissatisfied or that the current plan is infeasible, but they have no principled way to *reconstruct a viable path forward* when the user's newly specified constraints are impossible (e.g., premium features at a budget price). In such cases, the dialogue often degrades into failure, apology, or repetition. Our work builds directly on this line: we keep the dynamic, sentiment-aware goal monitoring of [3], but we add an explicit *persuasive layer* whose purpose is to guide the user from an infeasible goal space to a nearby feasible one without losing conversational rapport.

2.2. Computational Persuasion in Dialogue

Computational persuasion studies how conversational agents can influence user attitudes or decisions in a way that is transparent, context-aware, and often personalized [6, 7]. Prior work has shown that sentiment, user profile information, and dialogue history can be leveraged to select persuasive strategies such as highlighting benefits, reducing perceived cost, or appealing to social norms [8, 9]. Many such systems are designed for social good (e.g., encouraging exercise, donations, or pro-environmental behavior), operate in relatively open domains, and are evaluated on user engagement or attitude change rather than task completion.

Two characteristics of this literature limit its direct applicability to our setting. First, most persuasive dialogue models assume that the agent has *flexible* content to persuade with (e.g., multiple arguments, stories, or recommendations), whereas task-oriented systems often operate over a *constrained* catalog—there may simply be no item matching "8GB RAM, 512GB, $300." Second, existing persuasive models rarely integrate with a goal-tracking or slot-filling pipeline; persuasion is treated as an independent conversational skill rather than a tool for *repairing* an otherwise failing task.

Our work explicitly bridges this gap. We embed persuasion theory inside a constrained, task-oriented framework and define the persuasive objective narrowly: when the user requests something infeasible, the agent should (i) acknowledge the user's priority (e.g., storage), (ii) explain the constraint (e.g., price floor), and (iii) propose a *feasible* near-match (e.g., 256GB + expansion) that preserves as much utility as possible. This is why we introduce a strategy bank and a context-aware selector: different discrepancy types (budget overconstrained, feature overconstrained, time overconstrained) call for different persuasive tactics, and the agent must pick one that is consistent with the dialogue state rather than generating generic motivational talk.

2.3. Negotiation and Strategic Dialogue

Negotiation-style dialogue research [10–14] investigates how two agents (human–human or human–machine) can reach an agreement through iterative offers and counteroffers, often under conflicting preferences. These models are relevant because they show how an agent can reason over utilities, make concessions, and structure proposals to maximize agreement rates. However, our setting differs in two important ways.

First, in many negotiation tasks, *both* parties have private utilities and the outcome is the result of mutual compromise. In our scenario, the agent represents a service or product provider with *fixed environmental constraints* (the catalog, the price floor, the available dates). The user is the only side whose goal may need to be adjusted. This makes the problem closer to *guided recommendation* than to symmetric negotiation.

Second, negotiation work typically evaluates success in terms of agreement rate or joint payoff. In our case, success must be defined more broadly to reflect the goals articulated in the introduction: maintaining dialogue continuity, preserving positive sentiment, and achieving task completion *despite* an initially infeasible goal. This is precisely why we later introduce a tripartite reward—so that the agent is not penalized for trying to salvage the conversation, and is actually *rewarded* when it successfully persuades the user to accept a feasible alternative.

According to this, our work can be viewed as importing the *strategic, user-steering* flavor of negotiation and computational persuasion into the *structure and constraints* of dynamic task-oriented dialogue. Where prior systems either (i) filled slots for feasible goals, (ii) persuaded in open-ended social domains, or (iii) negotiated under symmetric preferences, we combine these strands to enable a goal-driven virtual agent that detects infeasibility and then *actively* guides the user toward a realizable solution.

3. Problem Formulation

We model the persuasive dynamic goal-driven dialogue as a POMDP extending the formulation of [3]. The agent interacts with a user whose goal may change over time and may occasionally become *infeasible* given domain constraints (e.g., price floor, inventory, feature combinations). Unlike the original DGDVA, our agent must (i) detect such infeasibility and (ii) actively *redirect* the user toward a nearby feasible goal using an appropriate persuasive strategy.

Formally, the dialogue is represented as a tuple

$$\mathcal{M} = (S, A, T, R, \Omega, O),$$

where S is the (partially observable) state space, A the action space, T the transition function, R the reward function, Ω the observation space, and O the observation function.

3.1. Augmented State Space

We extend the original DGDVA dialogue state with a persuasion-specific component:

$$s_t = [d_t,\ g_t,\ p_t], \tag{1}$$

where, d_t is the **dialogue state** at turn t (parsed user act, recognized slots, sentiment signal, dialogue history features), identical in spirit to task-oriented DST. g_t is the **goal-driven module (GDM) state** as in [3], capturing the currently hypothesized user goal, detected goal–system discrepancy, and sentiment-aware alignment score. p_t is the new **persuasion state**, which tracks the agent's attempt to repair an infeasible goal:

$$p_t = (act, strat, rej, flex),$$

where, $act \in \{0, 1\}$ indicates whether the persuasion mode is currently active, $strat \in \mathcal{S}$ is the index of the selected persuasive strategy (e.g., budget–feature trade-off, near-match recommendation), $rej \in \mathbb{N}$ counts consecutive user rejections of the current strategy, $flex \in [0, 1]$ is the agent's latent estimate of user flexibility/relaxation willingness, inferred from sentiment and prior turns.

Because user intent, sentiment, and flexibility are not perfectly observable, the policy operates over a belief state b_t induced by O as in standard POMDP settings.

3.2. Extended Action Space

The original DGDVA supports a fixed inventory of task actions (inform, request, confirm, offer, etc.). We extend this to allow the agent to *actively repair* infeasible requests:

$$A = A_{\text{task}} \ \cup \ A_{\text{pers}}, \tag{2}$$

where A_{task} is the original action set and A_{pers} is defined as

$$\begin{aligned} A_{\text{pers}} = \{&\texttt{PersuadeFeatureHighlight}(f),\ \texttt{SuggestCompromise}(c_1, c_2),\\ &B\texttt{ReframeValue}(a, b),\ \texttt{PresentAlternative}(alt)\}. \end{aligned}$$

Intuitively, `PersuadeFeatureHighlight` justifies the currently feasible option by emphasizing a user-valued attribute; `SuggestCompromise` explicitly asks the user to relax an overconstrained slot (e.g., budget) in exchange for a key benefit; `ReframeValue` changes the comparison basis (e.g., lifetime value vs. upfront cost); `PresentAlternative` offers a concrete, feasible substitute with explanation.

The transition function $T(s_{t+1} \mid s_t, a_t)$ updates not only the dialogue and goal state (as in DGDVA) but also the persuasion state: user rejection increments *rej*, acceptance resets *rej* and deactivates persuasion, and repeated rejections may trigger strategy switching.

3.3. Observations

At each turn the agent receives

$$o_t = (\text{user\_utterance}_t, \text{sentiment}_t, \text{accept/reject}_t),$$

where the last component is obtained either from explicit user language ("no, too expensive") or from a rejection classifier trained on DevVA-P. The observation function $O(o_t \mid s_t, a_{t-1})$ thus links persuasive actions to user reactions.

3.4. Enhanced Reward Function

Consistent with the introduction, the agent must be rewarded for *three* things: completing the task (as usual), maintaining or improving user sentiment, and successfully persuading the user when the current goal is infeasible. We therefore define

$$R(s_t, a_t) = R_{\text{task}}(s_t, a_t) + \lambda_s R_{\text{sent}}(s_t, a_t) + \lambda_p R_{\text{pers}}(s_t, a_t), \tag{3}$$

where, R_{task} is the standard success/failure/turn-cost reward inherited from DGDVA (e.g., +20 for successful completion, -1 per turn, -10 for dialogue failure), R_{sent} gives small positive reward for sentiment recovery or maintenance and small penalties for sentiment drops, R_{pers} is *only* active when a discrepancy has been detected and persuasion is warranted:

$$R_{\text{pers}} = \begin{cases} +r_{\text{accept}} & \text{if user accepts the persuasive redirection,} \\ -r_{\text{drop}} & \text{if user drops out after persuasion,} \\ -r_{\text{reject}} & \text{for each rejection of the same strategy,} \\ +r_{\text{switch}} & \text{if strategy switch leads to acceptance,} \end{cases}$$

with $r_{\text{accept}} \gg r_{\text{switch}} > 0$ and $r_{\text{drop}} > r_{\text{reject}} > 0$.

This structured reward makes the agent prefer (i) solving the task directly when feasible, (ii) invoking persuasion when infeasibility is detected, and (iii) adapting its persuasive strategy rather than repeating a failed one.

3.5. *Objective*

The agent seeks a policy $\pi(a \mid b)$ that maximizes the expected discounted return

$$J(\pi) = \mathbb{E}_{\pi,T,O}\left[\sum_{t=0}^{T} \gamma^t R(s_t, a_t)\right],$$

balancing task completion, sentiment-aware dialogue continuity, and successful persuasive redirection. This captures precisely the shift from passive discrepancy handling to active, strategy-driven guidance described in Sections 1 and 2.

4. Methodology: The PDG-DVA Architecture

Figure 1 presents the overall pipeline. We retain the original DGDVA stack (NLU, dialogue state tracking, sentiment-aware discrepancy detection, and goal-driven module) and insert two additions: (i) a *feasibility checker* that detects overconstrained or impossible user goals, and (ii) a *Persuasion Module* that selects and realizes a persuasion strategy whenever the current goal cannot be satisfied by the knowledge base (KB). The dialogue policy then reasons over an *augmented* state and action space, as defined in Section 3.

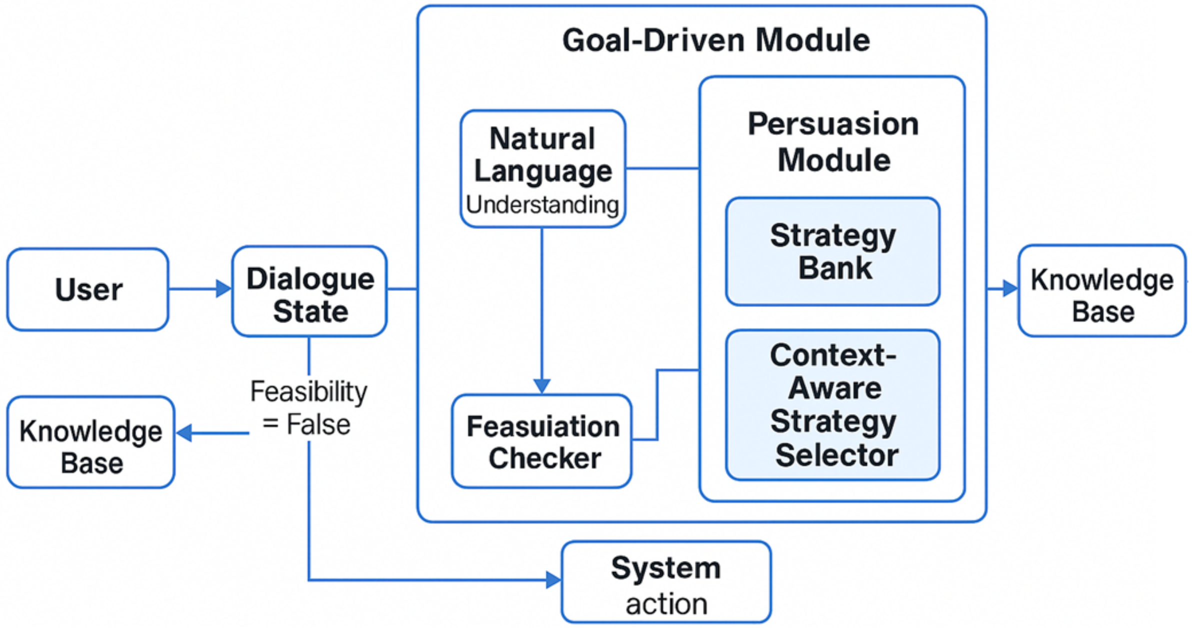

Fig. 1. System architecture of the proposed PDG-DVA, showing how the Persuasion Module is activated by the feasibility checker and integrated into the dialogue policy

4.1. Enhanced Goal-Driven Module

The original GDM in [3] monitors user sentiment and goal–system discrepancies. We extend it with an explicit feasibility layer that grounds the current goal G_t against the KB:

$$\text{Feasible}(G_t) = \exists e \in KB \text{ such that } e \models G_t. \tag{4}$$

Here, G_t is the goal hypothesis produced by the GDM (slots, price, brand, etc.), and $e \models G_t$ denotes that the KB entry e satisfies all constraints. If (4) is true, the system proceeds as a normal task-oriented agent. If (4) is false *and* the discrepancy detector reports negative sentiment or repeated corrections, the GDM raises a *persuasion trigger*. This trigger sets the persuasion flag in the state $p_t.act = 1$ (see Section 3) and hands the dialogue context to the Persuasion Module.

4.2. Persuasion Module

The Persuasion Module is responsible for transforming an infeasible request into a feasible alternative without breaking conversational rapport. It has two main components.

4.2.1. Strategy Bank. We define a finite set of persuasion strategies

$$\mathcal{S} = \{s_1, s_2, \ldots, s_k\},$$

constructed from well-known persuasive principles [15] and tailored to constrained recommendation settings, Feature Highlighting (s_1): justify a feasible option by emphasizing a user-valued attribute (e.g., battery, camera, storage expandability). Compromise Suggestion (s_2): explicitly ask the user to relax an overconstrained slot (e.g., budget by $20) to obtain the original feature set. Value Reframing (s_3): re-express the offer in terms of lifetime value, reliability, or future-proofing rather than raw specs. Social Proof (s_4): mention popularity/ratings of the closest feasible option to increase acceptability. Scarcity Emphasis (s_5): highlight limited availability or special offers to improve compliance.

Each s_i is associated with: (i) a set of response templates, (ii) a required slot set (what info it needs to talk persuasively), and (iii) an action pattern mapping to $A_{\text{persuasive}}$ defined in Section 3.

4.2.2. Context-Aware Strategy Selector. Since not all strategies are appropriate for every user, we learn a selector policy:

$$\pi_s(s \mid DS_t, GDM_t, H_t, p_t) = \text{softmax}(W[DS_t; GDM_t; H_t; p_t] + b), \tag{5}$$

where DS_t is the dialogue state, GDM_t is the goal-driven state (including discrepancy type), H_t are dialogue history features (rejections, sentiment trend), and p_t is the current persuasion state (activation flag, last strategy, rejection count, estimated flexibility). The selector is trained to maximize the downstream persuasion reward R_{pers} (cf. Eq. 3) so that it learns, for example, to switch from **Compromise Suggestion** to **Feature Highlighting** if the user has rejected price increases twice.

Optionally, a lightweight NLG layer realizes the chosen strategy into a final utterance:

$$\text{utterance}_t = \text{NLG}(s, DS_t, KB).$$

4.3. Dialogue Policy Learning

We follow the DQN-based dialogue policy used in DGDVA and extend it to our augmented state and action spaces. The policy now observes

$$S_t = [DS_t, GDM_t, p_t],$$

and can pick from both standard actions (A_{task}) and persuasive actions ($A_{\text{persuasive}}$). The Q-function is defined as

$$Q(S_t, A_t) = \mathbb{E}[R_t + \gamma \max_{A'} Q(S_{t+1}, A') \mid S_t, A_t], \tag{6}$$

and optimized via experience replay and a target network exactly as in standard deep RL for dialogue.

The key difference from vanilla DGDVA is that the reward R_t is now the *tripartite* signal from Section 3:

$$R_t = R_{\text{task}} + \lambda_s R_{\text{sent}} + \lambda_p R_{\text{pers}}.$$

Thus, a persuasive action that successfully moves the user from an infeasible to a feasible goal yields a spike in R_{pers}, making such actions more likely in future similar states. Conversely, repeatedly proposing the same compromise and getting rejected increases the rejection count in p_t and leads to negative R_{pers}, pushing the policy toward strategy switching.

4.4. Training Procedure

We train PDG-DVA with episodic simulated dialogues, mixing:

1. **Feasible-goal episodes** sampled from the original DevVA-style data (to preserve task competence).

2. **Infeasible-goal episodes** from our DevVA-P extension, where the feasibility checker is deliberately triggered and the policy must invoke persuasion.

Algorithm 1 summarizes the training loop. In practice, we interleave minibatches from both episode types so that the agent does not overfit to persuasion and forget how to solve ordinary tasks.

Algorithm 1 PDG-DVA Training Algorithm

```
Initialize Q-network Q, target network Q̂, replay buffer D
for episode = 1 to M do
    Sample scenario (feasible or infeasible) and initialize S_0 = [DS_0, GDM_0, p_0]
    for t = 1 to T_max do
        Select action A_t using ε-greedy over Q(S_t, ·)
        Execute A_t in simulator / environment
        Observe next state S_{t+1} and composite reward R_t
        Store transition (S_t, A_t, R_t, S_{t+1}) in D
        Sample a minibatch from D and update Q via gradient descent
        Update target network Q̂ every C steps
        if dialogue terminated then
            break
```

5. Experimental Setup

5.1. Dataset: DevVA-P Extension

To evaluate persuasion under infeasible goals, we extend the original DevVA dataset with 250 additional, professionally annotated dialogues in which the user issues overconstrained or impossible goal updates (DevVA-P). Table 1 summarizes the corpus.

Table 1. Statistics of DevVA-P dataset

Statistic	DevVA	DevVA-P
Total Dialogues	1,000	1,250
Total Utterances	8,335	10,587
Avg. Dialogue Length	8.3	8.5
Persuasion-needed Dialogues	0	250
Infeasible Goal Turns	0	312
Persuasion Strategy Annotations	0	587

We partition the DevVA-P extension into three splits, using 70% of the dialogues for training, 15% for validation, and 15% for testing. The original 1,000 DevVA dialogues are kept in the training pool primarily to ensure that the agent continues to learn and retain standard task-oriented behavior, rather than overfitting to persuasion-only cases. For every dialogue in DevVA-P that actually requires persuasion, we add fine-grained annotations indicating (i) the exact turn at which persuasion should be triggered—this corresponds to the moment the feasibility checker detects that the user's updated goal cannot be met, (ii) whether the user's goal update was feasible or infeasible, (iii) which persuasion strategy from the predefined set $\mathcal{S}$ is considered appropriate for that context, and (iv) what the final user reaction was (accepting the alternative, rejecting it, or abandoning the dialogue). These structured labels are used to train the context-aware strategy selector described in Section 4, and they also act as oracle references when we compute persuasion-specific evaluation metrics such as persuasion acceptance rate and per-strategy effectiveness.

5.2. Baseline Models

In our experiments we evaluate PDG-DVA against both the original dynamic agent and several controlled variants to show where the performance gains actually come from. First, we include the original DGDVA model [3], which already supports dynamic goal adaptation and sentiment-aware discrepancy detection but does not have any persuasive actions; this is our primary reference to demonstrate the benefit of adding persuasion on top of an already strong system. Second, we test a variant we call DGDVA-P-Random, which is our proposed architecture with the Persuasion Module turned on, but where the persuasion strategy is chosen uniformly at random from the strategy set $\mathcal{S}$. This baseline tells us whether improvements are due simply to having more actions, or specifically to choosing the right strategy for the context. Third, we add a Rule-Based Persuader, which uses simple, hand-crafted rules to map the detected discrepancy type (for example, a budget that is too low) to a single fixed persuasive tactic (such as `SuggestCompromise`); this reflects what a practical, non-learning system might do in an industry setting. Finally, we report results for DDQN-NoPersuasion, which uses our augmented state but removes all persuasive actions, allowing us to isolate and quantify the contribution of persuasion itself separate from other architectural changes.

5.3. Evaluation Metrics

We assess the proposed model along two complementary dimensions: how well it still performs the underlying task, and how well it carries out persuasion when a user goal is infeasible. For automatic evaluation, we first report the standard Task Success Rate, i.e., the percentage of dialogues that end with the agent successfully completing the user's goal; this keeps our results comparable to the original DGDVA line of work. Because our setting explicitly introduces infeasible goals, we also compute the Persuasion Acceptance Rate (PAR), which measures, among only those dialogues where the user asked for something impossible, how often the user accepted the agent's redirected or alternative offer. To better understand which persuasive behaviors are most effective, we break PAR down by strategy and report a Strategy Effectiveness score for each $s_i \in \mathcal{S}$, letting us see, for example, whether compromise suggestions work better than value reframing in budget-constrained cases. Finally, we track the Average Episode Reward across dialogues, which summarizes our tripartite reward design (task completion, sentiment preservation, and successful persuasion) in a single scalar.

In addition to these automatic metrics, we run a human evaluation on the held-out test set. We present sampled system responses to human annotators and ask them to rate each interaction on four 1–5 scales: Helpfulness (did the persuasion actually move the user toward a workable solution?), Perceived Intelligence (did the agent appear aware of domain constraints and user needs?), User Satisfaction (is the final outcome acceptable given that the original request could not be met?), and Naturalness (was the language fluent and context-appropriate?). These human judgments let us confirm that higher automatic persuasion scores, especially PAR, correspond to better perceived experience for real users.

5.4. Implementation Details

To make our results directly comparable with those of [3], we deliberately keep the training configuration almost identical to theirs and only add what is strictly needed for persuasion. We use a standard DQN reinforcement learning backbone with experience replay, a discount factor of $\gamma = 0.9$, and a learning rate of 10^{-3}, so that any performance improvements cannot be attributed to a more powerful RL setup. We train for 50,000 simulated dialogue episodes, and importantly, these episodes are a mix of ordinary, feasible-goal conversations (to preserve task-oriented competence) and the new infeasible-goal conversations from DevVA-P (to teach the model when and how to persuade). The input to the policy is a single concatenated feature vector that brings together three kinds of information: regular dialogue features (user acts, filled slots), goal-driven/discrepancy features (what the GDM thinks is wrong and the current sentiment score), and the new persuasion features (whether persuasion is active, which strategy was last used, how many times it was rejected, and the current estimate of user flexibility). On top of this, the persuasion strategy selector is implemented as a small two-layer MLP with 64 hidden units, trained jointly with the dialogue policy using the supervision signals we added to DevVA-P. Finally, we stabilize learning by updating the target network every $C = 200$ steps. Because the rest of the pipeline is held constant, we can attribute any observed gains in success rate, persuasion acceptance, or reward to the Persuasion Module and the augmented reward design rather than to unrelated changes in optimization or simulation.

6. Results and Discussion

In this section we report how well the proposed PDG-DVA performs compared to strong baselines on the DevVA-P test split. We focus on three questions: (i) does adding persuasion actually help in infeasible-goal dialogues, (ii) does learned, context-aware strategy selection beat hand-crafted or random persuasion, and (iii) do humans perceive the system as more helpful and intelligent.

6.1. Quantitative Results

Table 2. Performance comparison on DevVA-P test set

Model	Overall SR	Persuasion SR	PAR	Avg. Reward
DGDVA (Original)	0.841	0.692	–	68.4
DGDVA-P-Random	0.823	0.714	0.312	65.2
Rule-Based Persuader	0.861	0.783	0.567	72.1
DDQN-NoPersuasion	0.853	0.701	–	70.3
PDG-DVA (Ours)	**0.896**	**0.921**	**0.743**	**82.7**

As shown in Table 2, PDG-DVA achieves the best scores on all persuasion-sensitive metrics. While the original DGDVA already performs competitively on overall task success, it cannot report persuasion-specific acceptance because it does not attempt persuasion. Simply adding persuasion actions without context (DGDVA-P-Random) improves persuasion success slightly but remains far below our method, indicating that *strategy selection* is key. The rule-based persuader is a strong non-learning baseline, but PDG-DVA surpasses it, confirming that learning to map context to strategy yields the highest

Persuasion Acceptance Rate (0.743) and the highest average reward, which reflects our tripartite objective (task + sentiment + persuasion).

6.2. Human Evaluation

Table 3. Human evaluation results (1–5 scale)

Model	Helpfulness	Intelligence	Satisfaction	Naturalness
DGDVA (Original)	2.8	3.1	2.9	3.4
Rule-Based Persuader	3.4	3.3	3.2	3.1
PDG-DVA (Ours)	**4.3**	**4.1**	**4.2**	**3.9**

Table 3 shows that human annotators perceive PDG-DVA as clearly more helpful and more "aware" of domain constraints than both the original model and the rule-based variant. Even though persuasion adds an extra layer of reasoning, the naturalness score stays high (3.9), meaning the extra guidance does not make the agent sound robotic or pushy. This supports our claim that higher PAR translates into better user experience.

6.3. Strategy Analysis

Table 4. Effectiveness of different persuasion strategies

Strategy	Success Rate
Feature Highlighting	0.781
Compromise Suggestion	0.823
Value Reframing	0.692
Social Proof	0.734
Scarcity Emphasis	0.645

The per-strategy breakdown in Table 4 reveals that *Compromise Suggestion* is the most effective in our setting. This matches the data distribution in DevVA-P, where most failures stem from overconstrained price or spec requests; directly asking the user to relax exactly one constraint is the most natural fix. *Scarcity Emphasis* performs worst, likely because users in task-oriented, specification-heavy dialogues are more receptive to rational trade-offs than to urgency cues.

6.4. Qualitative Analysis

Table 5. Example dialogue comparison

DGDVA (Failure)	PDG-DVA (Success)
User: I want 8GB RAM, 512GB storage, max $300.	**User:** I want 8GB RAM, 512GB storage, max $300.
Agent: I found Phone A: 8GB RAM, 512GB, $699.	**Agent:** I understand you need ample storage. While 512GB models start at $600, Phone X offers 256GB with expandable storage at $280.
User: Too expensive!	**User:** Is 256GB sufficient?
Agent: [No feasible option]	**Agent:** With cloud backup and expandable storage, it should meet your needs while saving $400.
User: [Dissatisfied, dialogue fails]	**User:** That makes sense, I'll take it.

The dialogue in Table 5 illustrates the behavioral difference our architecture enables. The original agent correctly identifies an option but cannot reconcile the user's budget with the desired specs and the dialogue collapses. PDG-DVA, however, acknowledges the user's main requirement, explains the domain constraint, and proposes a nearby feasible option (256GB + expansion at $280). The user accepts, showing that the model successfully moved from passive discrepancy detection to active, context-aware guidance.

7. Conclusion

We presented PDG-DVA, the first dialogue system that integrates computational persuasion with dynamic goal adaptation. By augmenting the Goal-Driven Module with a novel Persuasion Module, we enable agents to actively guide users toward satisfactory outcomes when their initial goals are infeasible. Extensive experiments demonstrate significant improvements in success rates for challenging scenarios while maintaining strong performance on standard dialogues.

References

[1] Grudin, J., & Jacques, R. (2019, May). Chatbots, humbots, and the quest for artificial general intelligence. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems* (pp. 1-11).

[2] Budzianowski, P., Wen, T. H., Tseng, B. H., Casanueva, I., Ultes, S., Ramadan, O., & Gašić, M. (2018). Multiwoz–a large-scale multi-domain wizard-of-oz dataset for task-oriented dialogue modelling. arXiv preprint arXiv:1810.00278.

[3] Tiwari, A., Saha, T., Saha, S., Sengupta, S., Maitra, A., Ramnani, R., & Bhattacharyya, P. (2021). A dynamic goal adapted task oriented dialogue agent. *Plos One, 16*(4), e0249030.

[4] Liu, W., Wen, Y., Yu, Z., Li, M., Raj, B., & Song, L. (2017). Sphereface: Deep hypersphere embedding for face recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 212-220).

[5] Eric, M., & Manning, C. D. (2017). Key-value retrieval networks for task-oriented dialogue. arXiv preprint arXiv:1705.05414.

[6] Hunter, A., Chalaguine, L., Czernuszenko, T., Hadoux, E., & Polberg, S. (2019, August). Towards computational persuasion via natural language argumentation dialogues. In Joint German/Austrian Conference on Artificial Intelligence (Künstliche Intelligenz) (pp. 18-33). Cham: Springer International Publishing.

[7] Wang, X., Shi, W., Kim, R., Oh, Y., Yang, S., Zhang, J., & Yu, Z. (2019). Persuasion for good: Towards a personalized persuasive dialogue system for social good. arXiv preprint arXiv:1906.06725.

[8] Bera, B., Saha, S., Das, A. K., Kumar, N., Lorenz, P., & Alazab, M. (2020). Blockchain-envisioned secure data delivery and collection scheme for 5g-based iot-enabled internet of drones environment. *IEEE Transactions on Vehicular Technology,* 69(8), 9097-9111.

[9] Hunter, A. (2016). Computational persuasion with applications in behaviour change. In *Computational Models of Argument* (pp. 5-18). IOS Press.

[10] Lewis, I. (2017). *Social and Cultural Anthropology in Perspective: Their Relevance in the Modern World*. Routledge.

[11] Aarrevaara, T., Pinheiro, R., & Söderlind, J. (2019). Strategy as dialogue and engagement. *Reforms, Organizational Change and Performance in Higher Education*, 211-234.

[12] Kraus, S., & Wilkenfeld, J. (2013). Strategic negotiation in multiagent environments. In *Coordination Theory and Collaboration Technology* (pp. 93-124). Psychology Press.

[13] Rahwan, I., Ramchurn, S. D., Jennings, N. R., McBurney, P., Parsons, S., & Sonenberg, L. (2003). Argumentation-based negotiation. *The Knowledge Engineering Review, 18*(4), 343-375.

[14] Jennings, N. R., Faratin, P., Lomuscio, A. R., Parsons, S., Sierra, C., & Wooldridge, M. (2001). Automated negotiation: prospects, methods and challenges. *International Journal of Group Decision and Negotiation, 10*(2), 199-215.

[15] Cialdini, R. B. (2001). The science of persuasion. *Scientific American, 284*(2), 76-81.

How to cite this article: Pushpak Bhattacharyya and Sonia Shahzadi (2022). Persuasive Dynamic Goal-Driven Virtual Agents for Infeasible Task-Oriented Dialogues. *Bulletin of Computer and Data Sciences*, 3(2), 22-37. DOI: 10.71448/bcds2232-3

Received: 01/09/2022 **Revised:** 19/10/2022 **Accepted:** 29/11/2022 **Publish:** 30/12/2022

AND DATA SCIENCES BULLETIN OF COMPUTER *Bulletin of Computer and Data Sciences* is a peer-reviewed open access journal.