

Learning Steerable Clarification Policies with Collaborative Self-play

Jonathan Berant^{1†*}, Maximillian Chen^{1†}, Adam Fisch^{2†}, Reza Aghajani³,
Fantine Huot², Mirella Lapata² and Jacob Eisenstein^{2†}

¹Google Research, ²Google DeepMind, ³Meta, work done at Google Deepmind

[†]Core contributor

Abstract

To handle ambiguous queries, AI assistants must decide (a) when to guess the user’s intent and answer directly, (b) when to enumerate and respond to multiple plausible intents, and (c) when to ask a clarifying question. No strategy is always superior; the right decision depends on *contextual* factors such as user preferences or modality. For example, enumerating several possible interpretations can be cumbersome on small screens or voice-based settings, while a more interactive style may be dispreferred in desktop settings. In this work, we propose training *steerable policies* for managing such uncertainty via self-play. Given two agents, one simulating a user and the other an AI assistant, we generate conversations where the user issues a potentially ambiguous query, and the assistant must choose how to respond. The assistant receives as input the *numerical cost* of asking a clarification question, and of generating each word, and is trained to select the conversational action that maximizes its final reward: accuracy penalized by these costs. We apply Reinforced Self-Training (ReST) to optimize this objective and show that it yields a steerable policy that adapts its behavior based on the provided costs, improving both reward and accuracy. Moreover, the resulting policy generalizes to numerical cost values unobserved during training.

1 Introduction

Establishing common ground between a user and an AI assistant is key for successful communication, but is an area where AI assistants have been shown to struggle (Shaikh et al., 2024). As a result, assistants may offer responses that fail to match the true intent of an underspecified user request, or offer overly verbose responses that attempt to accommodate all possible intents — both failure modes that could be avoided by more interactive policies for building common ground, e.g., through clarification questions. Recent work (Chen et al., 2025; Zhang et al., 2025) has proposed to teach models when to ask clarification questions by using *self-play*: an episodic learning strategy in which one language model simulates a user, and another represents the AI assistant. The assistant is then finetuned on successful “rolled-out” conversations to learn when to answer a query and when to ask for clarification.

However, such prior work assumes that ambiguity should be uniformly handled by requesting clarification, which does not capture the richness of strategies a conversational participant might take to manage uncertainty. Given an ambiguous utterance, there are many possible approaches: (a) guess a single probable intent and respond to it; (b) enumerate multiple possible intents and provide a response for each; (c) issue a clarification question to pinpoint the exact intent. Which strategy is optimal? This depends on whom you are talking to, and in what context. Some users dislike long outputs and favor a more conversational approach; others prefer to avoid multi-turn conversations, but will relax this preference when it is not possible to process long outputs, such as while driving. In low-stakes settings it may be acceptable to simply make a best guess. A proficient assistant

*Corresponding author: joberant@google.com

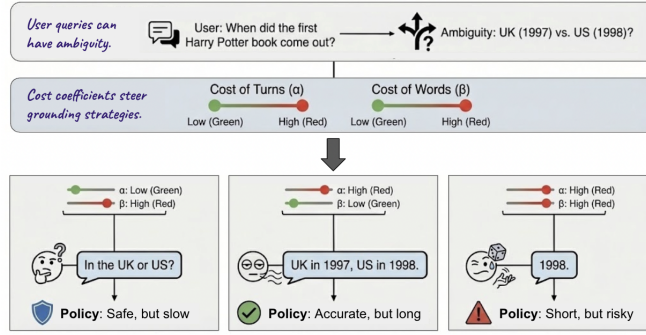


Figure 1: Our steerable AI assistant adapts its behavior based on cost coefficients for different dimensions of the conversation. The model is trained to maximize *cost-penalized accuracy*, with costs specified by α and β (the color red indicates high cost and the color green indicates low cost). In this example, the model asks a clarification question when the cost of conversation turns is low (*left*), enumerates multiple interpretations when the cost of generating a long response is low (*middle*), and answers directly with an educated guess when both costs are high (*right*).

should therefore take preferences, interaction modality, and other environmental factors into account when deciding how to respond. Thus, we need *steerable policies for conversation grounding*, which let the assistant flexibly adjust its dialogue strategy on the fly.

We propose a new form of self-play for training a single *steerable* policy that takes as input a specification of the reward for a conversation, and modifies its grounding behavior to maximize the provided reward. Figure 1 provides an example for an ambiguous query. The model is trained to maximize *cost-penalized accuracy*, where cost depends on the number of turns and the length of the final answer. This objective is parameterized by *scalar cost coefficients*, α and β , provided as part of the input. On the left, the cost of additional conversation turns is low, so the model clarifies uncertainty by asking a follow-up question. In the middle, the cost of enumerating interpretations is low, so the model provides answers for two probable interpretations. On the right, both costs are high, so the model makes an educated guess about user intent, without resolving the uncertainty. We focus in this work on training the assistant for *steerability*, assuming cost coefficients are given as input. In practice, cost coefficients can be inferred from user preference data or feedback, which we leave for future work.

To finetune our model, we apply Reinforced Self-Training (ReST), a common post-training strategy for optimizing rewards (e.g., Gulcehre et al., 2023; Zhang et al., 2024). We use one model as a user simulator and another as our target assistant, sampling different cost coefficients per conversation. We then apply collaborative self-play. For each question and cost coefficient setting, we sample multiple conversations, and compute the true reward. We then finetune the assistant on the highest-reward conversations, teaching it to condition on the cost coefficients to maximize its reward.

We evaluate our Steerable Grounding Policy (SGP) on two question-answering benchmarks: AMBIGQA (Min et al., 2020), where the model answers questions using parametric knowledge, and PACIFIC (Deng et al., 2022), where non-parametric context is provided in the form of a document and table. We find that even strong prompted models fail to condition on cost coefficients, even when using chain-of-thought prompting. In contrast, models *trained* with SGP appropriately condition on the provided cost coefficients; when needed, they can substantially alter their inference-time behavior, achieving higher accuracy at lower cost in accordance with the objective specified in the prompt. Moreover, this generalizes to cost coefficients never observed during training.

2 Method

2.1 Self-play Environment

Our self-play setup includes two agents, a *user simulator* (or *user* for short) and an *assistant*, who are collaborating to answer a *query* provided by the *environment*. The environment has access to a potentially ambiguous query q , a list of possible interpretations $\mathcal{I}_q =$

$(i_1, \dots, i_n)$, a corresponding distribution over the interpretations $\mathcal{P}_q = (p_1, \dots, p_n)$, where $\forall_j p_j \geq 0, \sum_j p_j = 1$, and the gold answers $(a_1, \dots, a_n)$. The environment samples an interpretation i_j , where $j \sim \mathcal{P}_q$ and passes it along with q to the user simulator. The user simulator is instructed to determine the answer by communicating with an *assistant* (who may have access to privileged information). After the ensuing multi-turn conversation, the user returns an answer a to the environment, which is compared with the gold answer $a^* = a_j$. An example conversation is shown in Figure 2.

Rollouts. A conversation, or *rollout*, is a sequence of turns $\rho = ((M_1, x_1, y_1), \dots, (M_T, x_T, y_T))$, where each turn includes the active model (user or assistant) M_t , the prompt x_t constructed from information in the state (see Table 1), and the model output, y_t . For us, M_t is the user when t is odd, and the assistant when t is even.

The output $y_t = (u_t, o_t)$ includes a chosen *action* and an observation for that action. The prompt x_1 is constructed based on the query q and the sampled interpretation i_j . The rollout terminates when the user returns the final answer $a = o_T$ alongside $u_T = \text{\#FINALIZE}$.

Agent actions and observations. Table 1 defines all actions for the user and assistant, along with the state, used to construct the prompt x_t , and the output observation. The assistant can choose one of three actions: $\{\text{\#CLARIFY}, \text{\#ANSWER}, \text{\#MULTI_ANS}\}$, corresponding to asking a clarification question, answering the query, or providing multiple interpretations. Our goal is to train the assistant to choose actions that maximize its reward (see below). The user chooses actions in a deterministic manner: in the first turn, it will $\text{\#QUERY}$; when given a clarification question q_{clar} it will $\text{\#RESPOND}$ with a clarification response r_{clar} ; and when given an answer or multi-answer from the assistant it will generate and return a final answer a ($\text{\#FINALIZE}$). When the assistant gives a multi-answer, the user leverages knowledge of i_j to choose the relevant part of the multi-answer. In practice we limit the number of allowed clarification questions (see Section 3).

Reward. Let $n_{\text{clar}}(\rho)$ denote the number of clarification actions and $|o_{T-1}|$ the number of words in the final assistant answer. The rollout reward computed by the environment is

$$R(\rho; a^*) = \text{acc}(a, a^*) - \alpha \cdot n_{\text{clar}}(\rho) - \beta \cdot |o_{T-1}|$$

We want the answer to be accurate, to minimize the number of turns, and to get short answers. Our reward is motivated by Grice’s maxims (Grice, 1975) for effective communication, including the maxim of *quality* (make your contributions true), *quantity* (provide necessary information only), and *manner* (avoid ambiguity). We measure accuracy, $\text{acc}(a, a^*)$, against the gold answer a^* with token-level F_1 (Rajpurkar et al., 2016), in a $[0, 100]$ scale.

The optimal strategy for this reward function depends on whether it is possible to increase expected accuracy with clarification questions or multi-answers. E.g., if asking a clarification question can increase expected accuracy by δ , and $\alpha < \delta$, then asking will yield higher reward compared to answering directly. Similarly, if a multi-answer adds m words to the final assistant answer but improves expected accuracy by δ , it is preferable when $\beta \cdot m < \delta$.

User and assistant prompts. Prompts x are constructed from the state, as explained in Table 1. The assistant gets the coefficients α and β so it can learn to change its behavior conditioned on them. The assistant has access to the history of clarification questions and their answers $h = \{(q_{\text{clar}}^1, r_{\text{clar}}^1), \dots\}$, and to an optional *private context* c . This is used when the query is w.r.t to some context (e.g., a table) that can be provided in the prompt. We show an example where the assistant has access to a private context in Figure 5 (Appendix B) and describe the details in Section 3. The full prompts are provided in Appendix I.

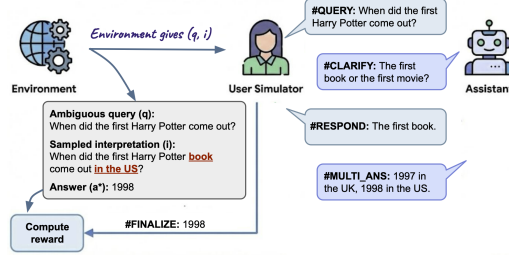


Figure 2: An example rollout. The environment passes the user an ambiguous query q and unambiguous interpretation i . The user passes q to the assistant, which issues a clarification question. After receiving a response, it outputs a multi-answer (covering multiple interpretations), from which the user formulates the final answer.

We rely on the base model’s instruction-following ability in two ways: (a) given a clarification question, we instruct the user to answer without revealing the true interpretation and (b) given the answer r_{ans} , we instruct the user simulator to only use r_{ans} and its knowledge of the true interpretation i_j to generate the final answer a . The user is instructed to never answer based on its parametric knowledge. We manually verify that the user’s instruction-following behavior is satisfactory. In addition, using parametric knowledge will lead to lower performance in asymmetric cases where the assistant’s knowledge is superior, e.g., when the assistant has access to a private context containing necessary knowledge.

Thoughts. We experiment with models that use chain-of-thought reasoning to generate actions. Before sampling an action, the model generates a thought that considers all allowed actions, and estimates which will yield the highest reward (similar in spirit to Dong et al. (2026)). Our hypothesis is that using a chain-of-thought will help the model better condition on the input cost coefficients, since it will explicitly reference coefficients to estimate the reward for each action. We provide the model a few in-context examples with thoughts, shown in the full prompts in Section I.2.

2.2 Training

Our goal is to train a steerable assistant that maximizes reward given input cost coefficients. To this end, we apply Reinforced Self-Training (ReST), a post-training method that approximates on-policy learning by updating model parameters between each epoch of training (e.g., Gulcehre et al., 2023; Zhang et al., 2024). In ReST, one generates rollouts, scores them, trains on high-scoring rollouts, and then iterates with the updated policy. However, the highest-scoring rollout might feature poorly-chosen actions that obtained a high reward by chance, e.g. by guessing an intent that is underspecified but (luckily) correct. To avoid training on such noisy supervision, we seek the *action sequence* that leads to the highest reward in expectation over rollouts and interpretations, and construct training examples from the best rollout compatible with this action sequence (see also Eisenstein et al. 2025; Zhang & Choi 2025). Algorithm 1 describes the implementation.

Recall that a rollout is defined as $\rho = \{(M_t, x_t, y_t)\}_{t=1}^T$, with action-observation outputs $y_t = (u_t, o_t)$. An action sequence $\mathbf{s} = (s_1, s_2, \dots)$ is a sequence of actions (e.g., #ANSWER or #CLARIFY). If for a given rollout ρ we have $u_t = s_t$ for all t , we write $\rho \vdash \mathbf{s}$ to indicate that ρ is *compatible* with $\mathbf{s}$. A set of rollouts, $\{\rho_1, \rho_2, \dots\}$ can be coarsened into a set of action sequences $\{\mathbf{s}_1, \mathbf{s}_2, \dots\}$. Given a query q and coefficients α, β , we can look at their set of rollouts, coarsen the rollouts to action sequences and identify the action sequence, $\mathbf{s}_*^q$ with the highest expected reward across interpretations (we drop α and β for brevity):

$$\mathbf{s}_*^q = \underset{\mathbf{s}^q}{\operatorname{argmax}} \sum_{j \in [n]} p_j \cdot \underbrace{\sum_{\{\rho: \rho \vdash \mathbf{s}^q\}} p_{t-1}(\rho|q, i_j) \cdot R(\rho)}_{\mathbb{E}_\rho[R(\mathbf{s}^q|q, i_j)]} \quad (1)$$

where p_{t-1} corresponds to the probability of the trajectory using the current assistant M_{t-1}^A and user M^{user} . In practice, we assume $\mathcal{P}_q$ is uniform and estimate p_{t-1} through sampling: we compute average reward per interpretation and average across interpretations, choosing the action sequence that does well across all of them. Once we have $\mathbf{s}_*^q$, we go over all

Agent	Action	State	Observation
User	#QUERY	q, i	q
	#RESPOND	q, i, q_{clar}	r_{clar}
	#FINALIZE	q, i, r_{ans}	a
Assistant	#CLARIFY	q, h, c, α, β	q_{clar}
	#ANSWER	q, h, c, α, β	r_{ans}
	#MULTI_ANS	q, h, c, α, β	r_{ans}

Table 1: The user observes the query q , its interpretation i , and potentially (a) a clarification question q_{clar} , (b) answer r_{ans} or (c) multi-answer r_{ans} . The assistant observes the query, a history h of clarification questions and answers, an optional private context c , and the coefficients α, β . The assistant outputs (a) clarification questions q_{clar} or (b) answers or multi-answers r_{ans} . The user responds to clarification questions with r_{clar} , and synthesizes a final answer a given r_{ans} . The state is used to construct the prompt x .

Algorithm 1 Reinforced Self-Training

Require: Initial assistant model M_0^A , fixed user model M^{user} , query set $\mathcal{Q} = \{(q_k, \mathcal{I}_{q_k}, \mathcal{P}_{q_k})\}_{k=1}^N$, set of coefficients $\mathcal{C} = \{(\alpha_j, \beta_j)\}_{j=1}^M$, rollouts per query n_r , number of ReST iterations T .

Ensure: Trained model M_{ReST}^A

```

for  $t = 1, \dots, T$  do
  Define an empty training set  $\mathcal{D} \leftarrow \emptyset$ .
  for query triple  $(q_k, \mathcal{I}_{q_k}, \mathcal{P}_{q_k}) \in \mathcal{Q}$  do
    Sample index  $j \sim \mathcal{P}_{q_k}$  and interpretation  $i_j \in \mathcal{I}_{q_k}$ .
    for  $\ell = 1, \dots, n_r$  do
      Sample  $\alpha, \beta \sim \mathcal{C}$ .
      Generate and save rollout  $\rho_{k,j,\ell}$  by applying models  $M_{t-1}^A$  and  $M^{\text{user}}$  to  $q_k, i_j, \alpha, \beta$ .
    end for
  end for
  for query  $q_k$ , its rollouts  $\rho_k$ , and  $(\alpha, \beta) \in \mathcal{C}$ , do
    Step a: Filter out invalid rollouts (see Section 2.2).
    Step b: Compute best action sequence:
       $\mathbf{s}_*^{k,\alpha,\beta} = \arg \max_{\mathbf{s}} \mathbb{E}_{\rho} [R(\mathbf{s}^{k,\alpha,\beta} \mid q_k)]$ 
    Step c: Compute best rollouts compatible with  $\mathbf{s}_*^{k,\alpha,\beta}$ :
       $\rho_*^{k,\alpha,\beta} = \bigcup_{i_j \in \mathcal{I}_{q_k}} \arg \max_{\{\rho: \rho \vdash \mathbf{s}_*^{k,\alpha,\beta}, i(\rho)=i_j\}} R(\rho)$ 
    Step d: Turn  $\rho_*^{k,\alpha,\beta}$  into training examples, add to  $\mathcal{D}$ .
  end for
  Train  $M_t^A$  on a training split of  $\mathcal{D}$ .
  Select checkpoint with smallest held-out log-likelihood on a development split of  $\mathcal{D}$ .
end for
return  $M_T^A$ 

```

interpretations, and add the highest-reward rollout compatible with $\mathbf{s}_*^q$ to the training set:

$$\rho_*^q = \bigcup_{i_j \in \mathcal{I}_q \mid \{\rho: \rho \vdash \mathbf{s}_*^q, i(\rho)=i_j\}} \arg \max R(\rho), \quad (2)$$

where $i(\rho)$ is the interpretation used in the trajectory ρ .

Algorithm 1 describes this procedure. We generate n_r rollouts for each query q_k and interpretation, sampling cost coefficients in each rollout. We cluster the n_r rollouts by cost coefficients, filtering out invalid rollouts. A rollout is invalid if its F_1 is below some threshold, or if the user simulator violates instructions and ignores the assistant’s answer. This is determined by observing that the fraction of words in o_T that also appear in o_{T-1} is below a threshold. Finally, for each query q_k and coefficient pair (α, β) we find the optimal action sequence $\mathbf{s}_*^{k,\alpha,\beta}$ and add to the training set the best rollouts $\rho_*^{k,\alpha,\beta}$ that are compatible with $\mathbf{s}_*^{k,\alpha,\beta}$, across interpretations of q_k . We only train on the *assistant* steps.

Crucially, different settings of α and β lead to different optimal action sequences. This lets the model learn the association between coefficient values and desired behaviors. ReST training proceeds for T epochs, where we sample from the latest model in each epoch.

3 Experimental Setup

Datasets AMBIGQA is a QA benchmark built from Natural Questions (Kwiatkowski et al., 2019), where queries are annotated with multiple interpretations and their answers. In manual inspection, we found noisy annotations (annotated interpretations that are not plausible). We therefore perform model-based filtering: we ask Gemini 2.0 Flash (Google, 2024a) to determine if a question is ambiguous, and keep only queries where models agree with human annotations (see Section 1.1 for our prompt). Then, we subsample the unambiguous queries such that the ratio of ambiguous to unambiguous queries is 2:1. Overall, this leads to 1,776 training examples and 382 development (dev) examples (the test set is not publicly available), with an average of 2.41 interpretations per query on the training set and 2.76 on the dev set. An example ambiguous query is “How many basketball national championships does Louisville have?”, referring to men’s or women’s basketball.

PACIFIC (Deng et al., 2022) is a multi-turn QA task that requires reasoning about contextual ambiguity grounded in real financial documents and tabular data (see Figure 5). This creates two differences from AMBIGQA. First, this is an *asymmetric* setup, where the assistant

provides an answer that can only be derived from an input document, which the user has no access to. This prevents the user from answering questions using its own knowledge while ignoring the assistant’s response. Second, the context-grounded and multi-turn nature of PACIFIC introduce different types of ambiguity. For example, a turn that appears underspecified may not be ambiguous given previous context (e.g., red text in Figure 6). Conversely, seemingly precise queries can be ambiguous due to details in the context (e.g., blue text in Figure 6). We use 3,744 training examples and 640 dev examples (test set ground truth is not publicly available), with equal numbers of ambiguous and unambiguous queries.

Models For the main results presented here, we use Gemma 2 9B (Google, 2024b) as our baseline model for both the user simulator¹ and AI assistant. In Appendices G.1 and G.2, we show additional results with Gemma 2 2B and QWEN 3 4B. We compare several natural baselines to our model:

- **PROMPTED**: queries the baseline model directly with a description of the task, reward objective, and several examples. This tests whether the model can natively condition on cost coefficients. We show the prompt in Section I.2.
- **PROMPTED-COT**: Similar to PROMPTED, except that the prompt includes instructions and examples to generate a thought that explicitly reasons about which action will maximize reward. See the prompt in Section I.2.
- **ANSWER**: uses the baseline but constrains it to answer directly without clarification questions or multi-answers.
- **MULTI-ANSWER (MA)**: uses the baseline but constrains the assistant to use #MULTI_ANS.
- **CLARIFY**: uses the baseline but constrains the assistant to use #CLARIFY and then #ANSWER.
- **CLARIFY-AND-MA**: uses the baseline but constrains the assistant to use #CLARIFY and then #MULTI_ANS.
- **ANSWER-FT**: To teach the base model the right answer format, we sample direct answers per question and finetune on the highest-reward answer observations.
- **SGP**: trains a Steerable Grounding Policy (Algorithm 1).

We also include SGP-ORACLE, an oracle that provides an approximate upper bound on the achievable reward. Using SGP, for every query q and coefficients α, β , we estimate the expected reward $\mathbb{E}_\rho[R(\mathbf{s} \mid \alpha, \beta)]$ for each action sequence $\mathbf{s}$. We then select the sequence, $\mathbf{s}^*$, that in hindsight attains the highest expected reward. This offers a reference point for the potential gains from optimal action sequences.

Last, we conduct a few ablations related to the details of our training procedure, which we describe in Appendix D.

Training and inference details We use $n_r = 192$ rollouts at training time and $n_r = 64$ at test time. Interpretations are sampled uniformly from the candidates provided in AMBIGQA and PACIFIC. In each rollout we randomly sample the clarification cost coefficient $\alpha \sim \{0, 2, 20\}$ and the per-token cost coefficient $\beta \sim \{0.1, 0.7, 5.0\}$. We limit the number of clarification questions to 1, since typically there is just one ambiguous aspect in each query, which results in four possible action sequences: answer vs. multi-answer; with and without clarification questions. We perform $T = 3$ ReST epochs, following prior work (Eisenstein et al., 2025). We tune only the learning rate, choosing the checkpoint with lowest dev loss. As explained in Section 2.2, we discard rollouts with low F_1 : < 0.1 for AMBIGQA and < 0.4 for PACIFIC. We also discard rollouts when the user ignores the assistant, omitting rollouts when the fraction of tokens in the answer returned by the user that do not appear in the assistant answer is above half. We provide statistics on how often filtering happens in Appendix E.

Evaluation We use the following metrics: (a) average reward, (b) average F_1 , (c) fraction of queries with a clarification question, (d) fraction of queries that use #MULTI_ANS. We compare models by query type (ambiguous and unambiguous) and across cost coefficients. We also report metrics for *steerability*, which we describe in detail below. Last, we measure steerability generalization by testing coefficients that did not appear during training.

¹We tested sensitivity to the user simulator, by using Gemma 3 4B as the user simulator instead of Gemma 2 9B, and found results were similar.

Models	Ambiguous				Unambiguous				
	Reward	F1	% Clar.	% MA	Reward	F1	% Clar.	% MA	
AmbigQA	PROMPTED	-34.49	16.16	66.83	63.88	-15.01	33.07	64.78	58.07
	PROMPTED-COT	-20.11	16.44	53.93	50.48	-5.02	29.12	48.34	49.54
	ANSWER	1.32	14.33	0.00	0.00	26.73	35.58	0.00	0.00
	CLARIFY	-9.59	16.30	100.00	0.00	8.86	31.86	100.00	0.00
	MULTI-ANSWER (MA)	-46.48	15.67	0.00	100.00	-25.40	33.76	0.00	100.00
	CLARIFY-AND-MA	-50.61	16.94	100.00	100.00	-35.26	32.28	100.00	100.00
	ANSWER-FT	3.64	16.40	8.75	9.98	30.99	39.65	2.31	4.58
	SGP	6.63	19.94	40.25	18.22	31.36	41.28	26.46	13.99
	SGP-ORACLE	15.81	23.13	35.78	11.08	42.05	48.66	20.03	10.32
PACIFIC	PROMPTED	11.46	55.28	17.99	47.18	42.44	69.51	4.82	19.15
	PROMPTED-COT	33.00	50.04	16.45	16.17	52.07	68.43	4.86	7.17
	ANSWER	42.44	52.25	0.00	0.00	56.91	71.52	0.00	0.00
	CLARIFY	45.48	62.70	100.00	0.00	46.90	68.85	100.00	0.00
	MULTI-ANSWER (MA)	-21.39	54.63	0.00	100.00	-4.04	63.34	0.00	100.00
	CLARIFY-AND-MA	-16.18	56.66	100.00	100.00	-22.25	57.77	100.00	100.00
	ANSWER-FT	13.59	57.55	14.30	44.75	41.81	72.58	4.71	24.03
	SGP	54.32	65.73	42.86	14.60	65.70	79.52	20.56	9.30
	SGP-ORACLE	65.24	73.89	43.46	14.05	70.95	83.60	15.60	9.39

Table 2: Performance on AMBIGQA and PACIFIC development sets using a Gemma 2 9B base model for ambiguous and unambiguous queries, averaged across rollouts. ‘% Clar.’ and ‘% MA’ denote the fraction of rollouts with clarification questions and multi-answers. SGP improves F₁ while learning appropriate clarification and multi-answer behavior, leading to overall higher reward.

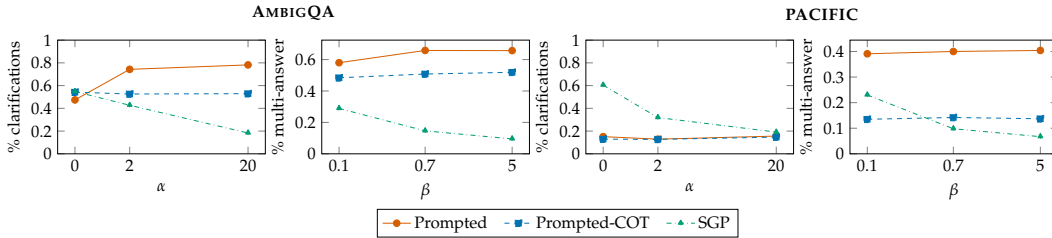


Figure 3: Fraction of rollouts with a clarification question and a multi-answer on the development sets for different cost coefficient values. The x -axis for α is in square-root scale and for β in log-scale. SGP reduces the fraction of clarifications with higher α and reduces the fraction of multi-answers with higher β , as expected. The baselines do not exhibit the correct sensitivity to coefficient values.

4 Experimental Results

Table 2 (top) shows results on AMBIGQA. SGP achieves the highest F₁ and reward for ambiguous *and* unambiguous queries. SGP also dramatically reduces the rate of clarification questions and multi-answers compared to PROMPTED. Using chain-of-thought improves performance and reduces the rate of clarification questions and multi-answers compared to PROMPTED, but lags behind SGP. The rate of clarification and multi-answer actions in SGP is higher for ambiguous queries than for unambiguous queries, as expected. Training on direct answer observations (ANSWER-FT) leads to almost always returning a direct answer, but gets higher F₁ than ANSWER since it is finetuned to answer in a better format. Comparing SGP to SGP-ORACLE, SGP does well in choosing high-reward action sequences, with a similar rate of clarifications and multi-answers. Still, a ~ 10 point headroom remains.

Table 2 (bottom) shows results on PACIFIC. SGP outperforms all baselines in terms of F₁ and reward. Unlike AMBIGQA, PROMPTED and PROMPTED-COT use clarification questions and multi-answers more sparingly. Moreover, both methods use more clarifications and multi-answers on ambiguous queries than unambiguous ones, showing they distinguish query types when relevant information is in the prompt. SGP increases the clarification rate on ambiguous queries to 42.86% to improve its performance. PROMPTED-COT achieves higher reward than PROMPTED, but trails SGP by a large margin. CLARIFY performs well on ambiguous queries, but worse than SGP. As expected, clarification questions

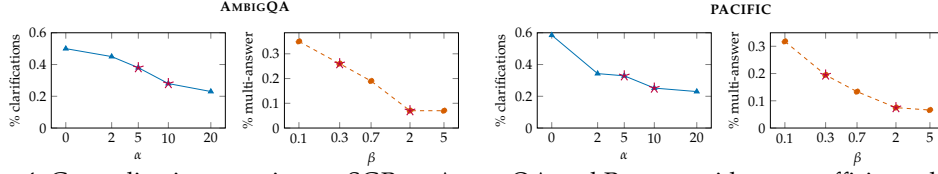


Figure 4: Generalization experiment: SGP on AMBIGQA and PACIFIC with cost coefficients that did *not* appear during training (purple stars). x -axis for α is in square-root scale and for β in log-scale. When varying α/β we fix $\beta = 5/\alpha = 20$. In both datasets the fraction of clarification questions and multi-answers is monotonic in α and β , showing good generalization to unobserved cost coefficients.

and multi-answers occur more frequently in ambiguous queries. ANSWER-FT achieves higher F_1 compared to ANSWER as a result of finetuning, but its performance is much lower than SGP. Comparing to SGP-ORACLE, we see moderate headroom for SGP: 11 points for unambiguous and 5 points for ambiguous queries. Remarkably, the fraction of clarifications and multi-answers for SGP closely matches the oracle. For ease of reading, we show in Table 2 results without confidence intervals, but we compute and report those in Appendix F, Table 9.

Since SGP is finetuned on both actions and observations, it learns more than just selecting the right action sequence: for #CLARIFY and #MULTI.ANS, it learns which clarification questions or multi-answers are likely to yield high reward. In Section C, we show cases where SGP issues better clarification questions than baselines, improving answers.

We include additional results in the appendix (Section G). Figure 9 (Section G.3) shows the average reward for SGP and baselines across all 9 coefficient pairs. The reward improves for *all* coefficient pairs and not just on average. Figure 10 (Section G.4) shows the action distribution across models and datasets, revealing predictable patterns. Figure 11 (Section G.5) shows that reward and F_1 improve monotonically across ReST epochs.

Corpus-level steerability Figure 3 shows the fractions of rollouts with clarification questions and with multi-answers for all trained cost coefficients for AMBIGQA and PACIFIC (for brevity, we omit the average length of the final assistant turn, as it behaves similarly to the fraction of multi-answers, but add this metric in App. G.6, Figure 12). For AMBIGQA, PROMPTED behaves counterintuitively, increasing the rate of clarification questions when α increases, and increasing the rate of multi-answers and turn length when β increases. PROMPTED-COT does not change the rate of clarifications or multi-answers as a function of α or β . SGP leads to the desired behavior: increasing α reduces the rate of clarifications, and increasing β reduces the rate of multi-answers. In PACIFIC, the behavior of SGP is similar to AMBIGQA, while PROMPTED and PROMPTED-COT do not modify their behavior as a function of different coefficients. Unlike AMBIGQA, PROMPTED-COT uses clarification questions and multi-answers rarely, while PROMPTED avoids clarifications, but uses multi-answers. These observations suggest that the base model cannot reason about numerical costs, whereas fine-tuning effectively teaches this capability.

Prompt-level steerability Next, we evaluate steerability at the *prompt level* by asking: for a given prompt, if the cost coefficient changes from one value to another, does the model optimally adjust its action sequence? To answer this, we first identify the *optimal action sequence* for each query and coefficient pair by determining which sequence yields the highest expected reward. We then analyze how the optimal action sequence *changes* as we vary one coefficient while fixing the other.

Data	Method	$\alpha: 0 \rightarrow 20$			$\beta: 0.1 \rightarrow 5.0$		
		R	P	F1	R	P	F1
AMBIG	Prom.	0.03	0.02	0.02	0.02	0.02	0.02
	COT	0.29	0.43	0.34	0.09	0.12	0.10
	SGP	0.45	0.49	0.47	0.09	0.42	0.12
PAC.	Prom.	0.02	0.28	0.04	0.02	0.19	0.04
	COT	0.03	0.21	0.05	0.01	0.09	0.02
	SGP	0.62	0.58	0.60	0.14	0.44	0.19

Table 3: Prompt-level steerability metrics when varying the cost of clarification questions (α) and multi-answers (β). Recall, precision, and F_1 are averaged across three possible values of β (clarification questions) and α (multi-answers).

Models	Ambiguous		Unambiguous		% Clarifications (α)			% Multi-answers (β)		
	Reward	F ₁	Reward	F ₁	0.0	2.0	20.0	0.1	0.7	5.0
GEMINI-2.5-FLASH	-15.35	23.96	51.83	60.88	23.58	34.19	49.59	24.40	21.95	25.28
GEMINI-3-FLASH	-19.93	30.55	27.95	60.40	33.86	29.75	22.42	67.43	64.91	58.10
SGP	6.63	19.94	41.28	26.46	54.78	42.71	18.34	28.91	14.60	9.49

Table 4: Performance and steerability of prompted Gemini models compared to SGP on AMBIGQA.

For clarification questions, We check if asking (or removing) a clarification question is optimal when modifying $\alpha_1 \rightarrow \alpha_2$, while fixing β . For multi-answers, we check if asking (or removing) a multi-answer is optimal when $\beta_1 \rightarrow \beta_2$, while fixing α . We then compute recall, precision, and F₁ in the following manner. Recall is the fraction of cases where the optimal action sequence changes when changing cost coefficients (e.g., adding a clarification question), and our model predicts the correct change. Precision is the fraction of cases where our model predicts a certain change in the action sequence, and this change is correct according to the optimal action sequence change. F₁ is the usual harmonic average. We describe the details of this procedure in Section H.

Table 3 reports results when changing the value of α from 0 $\rightarrow$ 20 and the value of β from 0.1 $\rightarrow$ 5.0. SGP outperforms baselines across the board. Results are substantially higher for steering α compared to β , where precision is high but recall is low. We attribute this to the baseline model producing better training rollouts for asking good clarification questions compared to multi-answers. Lastly, our results show there remains substantial headroom in learning policies that choose the optimal action sequence across prompts.

Generalization to new cost coefficients. We test if SGP can generalize to cost coefficients unobserved at training time. Figure 4 shows the fraction of clarification questions and multi-answers for observed and unobserved cost coefficients (see purple stars in the figure) for AMBIGQA (left) and PACIFIC (right). When varying α , we fix $\beta = 5$, and when varying β , we fix $\alpha = 20$. SGP generalizes well to unobserved values of both α and β on both AMBIGQA and PACIFIC, smoothly adjusting the action frequencies between the values corresponding to neighboring coefficients observed at training time. This shows SGP yields *generalizable steerability*, giving users control even at cost coefficients that are unobserved during training. Results for the length of the final assistant turn are similar and omitted for brevity.

Are Gemini models steerable? We saw that Gemma 2 9B can be *trained* to respond to steering signals, but that it is not steerable “out-of-the-box”. A natural followup is whether stronger models are steerable without fine-tuning. To this end, we compare SGP to Gemini-2.5 Flash (Google, 2025a) and Gemini-3 Flash (Google, 2025b) on AMBIGQA.

Table 4 shows the results for this comparison. Gemini models have better parametric knowledge and consequently higher F₁. Nevertheless, reward is lower for ambiguous question, and also for unambiguous questions in Gemini-3, due to sub-optimal use of clarification questions and multi-answers. More importantly, Gemini-2.5 is not steerable, with the number of clarification questions increasing (instead of decreasing) when α grows, and no sensitivity to the value of β . Gemini-3 exhibits moderate steerability, but its overuse of multi-answers reduces its reward. Overall, the performance of even these strong models is sub-optimal when asked to maximize a prompt-specified objective.

5 Related Work

Handling ambiguity User queries are rife with ambiguities, consistent with the principle of least effort (Zipf, 1949). While Grice’s *maxim of quantity* suggests that a cooperative speaker provides as much information as necessary (Grice, 1975), users of AI assistants may assume context that is unavailable to agent, thereby requiring additional grounding. Consequently, a growing body of work is focused on handling ambiguity in LLMs (Tanjim et al., 2025)). A natural approach is to ask clarification questions (Dhole, 2020; Ren et al.,

2023; Zhang & Choi, 2025): for example, Chen et al. (2025) and Zhang et al. (2025) use self-play with user simulators to learn a policy for asking clarification questions, while Kim et al. (2024) let language models assess their own uncertainty and design a corresponding clarification policy. Other work generates long responses enumerating multiple plausible user interpretations and their answers (Stelmakh et al., 2022; Amplayo et al., 2023; Kim et al., 2023; In et al., 2025).

Recently, Dong et al. (2026) proposed an inference-time method for deciding when to ask clarification questions that is based on explicit estimation of the expected value of each possible agent action. Our PROMPTED-COT baseline takes a similar approach, but makes less assumptions on the space of possible clarification questions and answers.

Steerable language policies Teaching contextual steerability has been a north star of language generation (Hovy, 1987; 1988; Chen et al., 2022; Keskar et al., 2019; Sorensen et al., 2026). Recent work on adapting to uncertainty includes Cole et al. (2023), who teach models to abstain from answering questions when uncertain about the asker’s intent, and Kuhn et al. (2022), who prompt models to quantify their uncertainty to determine when to ask clarification questions. However, these approaches are “steerable” only by the level of uncertainty, not by other contextual factors. Wang et al. (2024) train *conditional language policies* (CLPs) for steerable alignment: given multiple reward models, they train to maximize a linear combination of rewards, where the combination coefficients are sampled during training and provided as input at test time. We use a similar approach for training steerable grounding policies under different cost considerations. However, our approach uses coefficients specified in the prompts, allowing for training a *single* model, whereas Wang et al. (2024) train *multiple* models (with some parameter sharing) to achieve steerability.

Finally, Sun et al. (2025) use RL to train models to ask clarification questions and respond to prompt-specified user prompts for agentic tasks. In contrast, we focus on training assistants to maximize a reward function specified by *numerical cost coefficients*. In addition, we closely link steerability to the use of clarification, viewing steerability as a knob for controlling the spectrum of possible grounding strategies. In contrast, Sun et al. (2025) focus on steerability for stylistic personalization, which they treat as orthogonal to posing clarification questions.

Multi-agent self-play A large body of work explores the use of multi-agent interaction as part of post-training (in contrast to inference-time multi-agent systems) for improving reasoning (Chen et al., 2024; Subramaniam et al., 2025) and factuality (Khan et al., 2024). Other lines of research (including this work) focus on self-play for improving the collaborative and pragmatic skills of assistants (Lin et al., 2024). Beyond ambiguity, past work has also explored self-play for improving uncertainty expression (Stengel-Eskin et al., 2024; Eisenstein et al., 2025), tool use (Eisenstein et al., 2025), handling of persuasion (Stengel-Eskin et al., 2025), personalization (Wagner et al., 2024), and general collaborative skills (Wu et al., 2025). Our work uses multi-agent self play as a training framework, applying it to the objective of learning a cost-sensitive, dynamically steerable policy for managing uncertainty.

6 Conclusions

Prior work on uncertainty has focused on hedging and clarification in isolation, addressing *single* disambiguation strategies. We argue that choosing the *right* disambiguation strategy is contextual, and propose training models that select a cost-optimal strategy, based on numerical steering coefficients. We achieve this through SGP, yielding cost-accuracy trade-offs that are responsive to steering signals while also being pareto-optimal in aggregate. Looking forward, a natural next step is to close the loop and infer user preferences over time, leading to proactive and flexible AI assistants that infer user preferences and adapt to them on-the-fly. We focused on three disambiguation strategies — guessing, clarifying, and hedging — but the range of subtle grounding actions, that human speakers use to signal interpretation difficulties is much wider, often including nonverbal cues. Extending SGP to such grounding actions is a challenge for future work. Finally, steering models with numerical values has applications beyond uncertainty management, in areas such as controlling tool use, test-time compute, and more.

Acknowledgments

We thank Alekh Agarwal, William Cohen, Chris Dyer, Tal Linzen, and Eric Malmi for their thoughtful feedback. We thank Liu Leqi for discussions on training data generation. We thank Dheeru Dua and Vicky Zayats for developing supporting infrastructure.

References

- Reinald Kim Amplayo, Kellie Webster, Michael Collins, Dipanjan Das, and Shashi Narayan. Query refinement prompts for closed-book long-form QA. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (eds.), *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics*, pp. 7997–8012, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.444. URL <https://aclanthology.org/2023.acl-long.444/>.
- Justin Chih-Yao Chen, Swarnadeep Saha, Elias Stengel-Eskin, and Mohit Bansal. Magdi: structured distillation of multi-agent interaction graphs improves reasoning in smaller language models. In *Proceedings of the 41st International Conference on Machine Learning, ICML/24*. JMLR.org, 2024.
- Maximillian Chen, Weiyan Shi, Feifan Yan, Ryan Hou, Jingwen Zhang, Saurav Sahay, and Zhou Yu. Seamlessly integrating factual information and social content with persuasive dialogue. In *Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 399–413, 2022.
- Maximillian Chen, Ruoxi Sun, Tomas Pfister, and Sercan Ö. Arik. Learning to clarify: Multi-turn conversations with action-based contrastive self-training. In *The Thirteenth International Conference on Learning Representations*, 2025. URL <https://openreview.net/pdf?id=SiE6VFps9x>.
- Jeremy Cole, Michael Zhang, Daniel Gillick, Julian Eisenschlos, Bhuwan Dhingra, and Jacob Eisenstein. Selectively answering ambiguous questions. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 530–543, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.35. URL <https://aclanthology.org/2023.emnlp-main.35/>.
- Yang Deng, Wenqiang Lei, Wenxuan Zhang, Wai Lam, and Tat-Seng Chua. PACIFIC: Towards proactive conversational question answering over tabular and textual data in finance. In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang (eds.), *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 6970–6984, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.emnlp-main.469. URL <https://aclanthology.org/2022.emnlp-main.469/>.
- Kaustubh D. Dhole. Resolving intent ambiguities by retrieving discriminative clarifying questions. *ArXiv*, abs/2008.07559, 2020. URL <https://api.semanticscholar.org/CorpusID:221150323>.
- Yijiang River Dong, Tiancheng Hu, Zheng Hui, Caiqi Zhang, Ivan Vulić, Andreea Bobu, and Nigel Collier. Value of information: A framework for human-agent communication. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 42879–42896, San Diego, California, United States, July 2026. Association for Computational Linguistics. ISBN 979-8-89176-390-6. doi: 10.18653/v1/2026.acl-long.1987. URL <https://aclanthology.org/2026.acl-long.1987/>.
- Jacob Eisenstein, Reza Aghajani, Adam Fisch, Dheeru Dua, Fantine Huot, Mirella Lapata, Vicky Zayats, and Jonathan Berant. Don’t lie to your friends: Learning what you know from collaborative self-play. In *Second Conference on Language Modeling*, 2025. URL <https://openreview.net/forum?id=2vDjiGUfhV>.

- Google. Introducing gemini 2.0: our new ai model for the agentic era, 2024a. URL <https://blog.google/technology/google-deepmind/google-gemini-ai-update-december-2024/#ceo-message>.
- Google. Gemma 2: Improving open language models at a practical size, 2024b. URL <https://arxiv.org/abs/2408.00118>.
- Google. Gemini 2.5: Our most intelligent ai model, 2025a. URL <https://blog.google/innovation-and-ai/models-and-research/google-deepmind/gemini-model-thinking-updates-march-2025/>.
- Google. A new era of intelligence with gemini 3, 2025b. URL <https://blog.google/products-and-platforms/products/gemini/gemini-3/>.
- Herbert Paul Grice. Logic and conversation. *Syntax and semantics*, 3:43–58, 1975.
- Caglar Gulcehre, Tom Le Paine, Srivatsan Srinivasan, Ksenia Konyushkova, Lotte Weerts, Abhishek Sharma, Aditya Siddhant, Alex Ahern, Miaosen Wang, Chenjie Gu, et al. Reinforced self-training (ReST) for language modeling. *arXiv preprint arXiv:2308.08998*, 2023.
- Eduard Hovy. Some pragmatic decision criteria in generation. In *Natural language generation: New results in Artificial Intelligence, psychology and linguistics*, pp. 3–17. Springer, 1987.
- Eduard Hovy. Two types of planning in language generation. In *26th Annual Meeting of the Association for Computational Linguistics*, pp. 179–186, 1988.
- Yeonjun In, Sungchul Kim, Ryan A. Rossi, Mehrab Tanjim, Tong Yu, Ritwik Sinha, and Chanyoung Park. Diversify-verify-adapt: Efficient and robust retrieval-augmented ambiguous question answering. In Luis Chiruzzo, Alan Ritter, and Lu Wang (eds.), *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 1212–1233, Albuquerque, New Mexico, April 2025. Association for Computational Linguistics. ISBN 979-8-89176-189-6. doi: 10.18653/v1/2025.naacl-long.56. URL <https://aclanthology.org/2025.naacl-long.56/>.
- Nitish Shirish Keskar, Bryan McCann, Lav Varshney, Caiming Xiong, and Richard Socher. CTRL - A Conditional Transformer Language Model for Controllable Generation. *arXiv preprint arXiv:1909.05858*, 2019.
- Akbir Khan, John Hughes, Dan Valentine, Laura Ruis, Kshitij Sachan, Ansh Radhakrishnan, Edward Grefenstette, Samuel R. Bowman, Tim Rocktäschel, and Ethan Perez. Debating with more persuasive llms leads to more truthful answers. In *ICML*, 2024. URL <https://openreview.net/forum?id=iLCZt17FTa>.
- Gangwoo Kim, Sungdong Kim, Byeongguk Jeon, Joonsuk Park, and Jaewoo Kang. Tree of clarifications: Answering ambiguous questions with retrieval-augmented large language models. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 996–1009, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.63. URL <https://aclanthology.org/2023.emnlp-main.63/>.
- Hyuhng Joon Kim, Youna Kim, Cheonbok Park, Junyeob Kim, Choonghyun Park, Kang Min Yoo, Sang-goo Lee, and Taeuk Kim. Aligning language models to explicitly handle ambiguity. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 1989–2007, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.119. URL <https://aclanthology.org/2024.emnlp-main.119/>.
- Lorenz Kuhn, Yarin Gal, and Sebastian Farquhar. CLAM: Selective clarification for ambiguous questions with generative language models. 2022. URL <https://api.semanticscholar.org/CorpusID:257038525>.

- Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:452–466, 2019. doi: 10.1162/tacl.a.00276. URL <https://aclanthology.org/Q19-1026/>.
- Jessy Lin, Nicholas Tomlin, Jacob Andreas, and Jason Eisner. Decision-oriented dialogue for human-ai collaboration. *Transactions of the Association for Computational Linguistics*, 12: 892–911, 2024.
- Sewon Min, Julian Michael, Hannaneh Hajishirzi, and Luke Zettlemoyer. AmbigQA: Answering ambiguous open-domain questions. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu (eds.), *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 5783–5797, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.466. URL <https://aclanthology.org/2020.emnlp-main.466/>.
- Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. SQuAD: 100,000+ questions for machine comprehension of text. In Jian Su, Kevin Duh, and Xavier Carreras (eds.), *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 2383–2392, Austin, Texas, November 2016. Association for Computational Linguistics. doi: 10.18653/v1/D16-1264. URL <https://aclanthology.org/D16-1264/>.
- Allen Z. Ren, Anushri Dixit, Alexandra Bodrova, Sumeet Singh, Stephen Tu, Noah Brown, Peng Xu, Leila Takayama, Fei Xia, Jake Varley, Zhenjia Xu, Dorsa Sadigh, Andy Zeng, and Anirudha Majumdar. Robots that ask for help: Uncertainty alignment for large language model planners. In *Proceedings of the Conference on Robot Learning (CoRL)*, 2023.
- Omar Shaikh, Kristina Gligoric, Ashna Khetan, Matthias Gerstgrasser, Diyi Yang, and Dan Jurafsky. Grounding gaps in language model generations. In Kevin Duh, Helena Gomez, and Steven Bethard (eds.), *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 6279–6296, Mexico City, Mexico, June 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.naacl-long.348. URL <https://aclanthology.org/2024.naacl-long.348/>.
- Taylor Sorensen, Benjamin Newman, Jared Moore, Chan Young Park, Jillian Fisher, Niloofar Mireshghallah, Liwei Jiang, and Yejin Choi. Spectrum tuning: Post-training for distributional coverage and in-context steerability. In *The Fourteenth International Conference on Learning Representations*, 2026. URL <https://openreview.net/forum?id=ulvp7cbZeU>.
- Ivan Stelmakh, Yi Luan, Bhuwan Dhingra, and Ming-Wei Chang. ASQA: Factoid questions meet long-form answers. In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang (eds.), *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 8273–8288, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.emnlp-main.566. URL <https://aclanthology.org/2022.emnlp-main.566/>.
- Elias Stengel-Eskin, Peter Hase, and Mohit Bansal. LACIE: Listener-aware finetuning for calibration in large language models. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL <https://openreview.net/forum?id=RnvgYd9RAH>.
- Elias Stengel-Eskin, Peter Hase, and Mohit Bansal. Teaching models to balance resisting and accepting persuasion. In Luis Chiruzzo, Alan Ritter, and Lu Wang (eds.), *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 8108–8122, Albuquerque, New Mexico, April 2025. Association for Computational Linguistics. ISBN 979-8-89176-189-6. doi: 10.18653/v1/2025.naacl-long.412. URL <https://aclanthology.org/2025.naacl-long.412/>.

- Vighnesh Subramaniam, Yilun Du, Joshua B Tenenbaum, Antonio Torralba, Shuang Li, and Igor Mordatch. Multiagent finetuning: Self improvement with diverse reasoning chains. In *International Conference on Learning Representations*, 2025. URL <https://openreview.net/forum?id=gjeQKFxFpZ>.
- Weiwei Sun, Xuhui Zhou, Weihua Du, Xingyao Wang, Sean Welleck, Graham Neubig, Maarten Sap, and Yiming Yang. Training proactive and personalized LLM agents, 2025. URL <https://arxiv.org/abs/2511.02208>.
- Md Mehrab Tanjim, Yeonjun In, Xiang Chen, Victor S Bursztyn, Ryan A Rossi, Sungchul Kim, Guang-Jie Ren, Vaishnavi Muppala, Shun Jiang, Yongsung Kim, et al. Disambiguation in conversational question answering in the era of LLM: A survey. *arXiv preprint arXiv:2505.12543*, 2025.
- Nicolas Wagner, Dongyang Fan, and Martin Jaggi. Personalized collaborative fine-tuning for on-device large language models. In *First Conference on Language Modeling*, 2024. URL <https://openreview.net/forum?id=bwo3GVsg0v>.
- Kaiwen Wang, Rahul Kidambi, Ryan Sullivan, Alekh Agarwal, Christoph Dann, Andrea Michi, Marco Gelmi, Yunxuan Li, Raghav Gupta, Kumar Avinava Dubey, Alexandre Rame, Johan Ferret, Geoffrey Cideron, Le Hou, Hongkun Yu, Amr Ahmed, Aranyak Mehta, Leonard Hussenot, Olivier Bachem, and Edouard Leurent. Conditional language policy: A general framework for steerable multi-objective finetuning. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 2153–2186, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-emnlp.118. URL <https://aclanthology.org/2024.findings-emnlp.118/>.
- Shirley Wu, Michel Galley, Baolin Peng, Hao Cheng, Gavin Li, Yao Dou, Weixin Cai, James Zou, Jure Leskovec, and Jianfeng Gao. CollabLLM: From passive responders to active collaborators. In *Forty-second International Conference on Machine Learning*, 2025. URL <https://openreview.net/forum?id=DmH4HHVb3y>.
- Dan Zhang, Sining Zhoubian, Ziniu Hu, Yisong Yue, Yuxiao Dong, and Jie Tang. Rest-mcts: Llm self-training via process reward guided tree search. In *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2024.
- Michael J. Q. Zhang and Eunsol Choi. Clarify when necessary: Resolving ambiguity through interaction with LMs. In Luis Chiruzzo, Alan Ritter, and Lu Wang (eds.), *Findings of the Association for Computational Linguistics: NAACL 2025*, pp. 5526–5543, Albuquerque, New Mexico, April 2025. Association for Computational Linguistics. ISBN 979-8-89176-195-7. doi: 10.18653/v1/2025.findings-naacl.306. URL <https://aclanthology.org/2025.findings-naacl.306/>.
- Michael J. Q. Zhang, W. Bradley Knox, and Eunsol Choi. Modeling future conversation turns to teach llms to ask clarifying questions. In *The Thirteenth International Conference on Learning Representations*, 2025. URL <https://arxiv.org/abs/2410.13788>.
- George Kingsley Zipf. *Human behavior and the principle of least effort: An introduction to human ecology*. Ravenio books, 1949.

A LLM Disclosure

We used language and image models in the following ways: (1) creating figures and formatting tables by generating LaTeX code; (2) generating Figures 1, 2, and 5; (3) writing data analysis code; (4) reviewing the paper and suggesting areas in which the writing could be improved.

B Additional Details on the PACIFIC Dataset

Figure 5 shows an illustrative example from PACIFIC, where the query is ambiguous given the provided table context, which is shown only to the assistant.

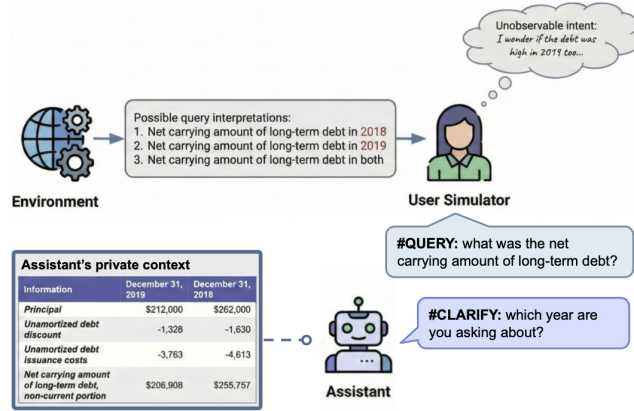


Figure 5: The assistant has access to a private context (table in this case), which is necessary for determining ambiguity of the query. The user does not have access to this context.

Figure 6 depicts an example of the different types of conversation phenomena that occur in the PACIFIC corpus. In AMBIGQA, all questions are inherently single-turn, and the source of whether it is ambiguous or not has to come from parametric knowledge. In contrast, as highlighted in the blue text of Figure 6, we can see that the source of ambiguity in the question “What was the Net carrying amount of long-term debt?” comes from the contextual knowledge — the Assistant is provided a table which contains two different possible years with “Net carrying amount of long-term debt.” We can also see by the text highlighted in red that although a single-turn question such as “How about 2018?” may be underspecified, the question is not ambiguous due to the multi-turn context which indicates that the user is still asking about “Net carrying amount of long-term debt.”

	Data		
	Dec 31, 2019	Dec 31, 2018	
Principal	\$212,000	\$262,000	User: What was the Net carrying amount of long-term debt ?
Less			
Unamortized debt discount	-1,328	-1,630	Assistant: What year are you asking about?
Unamortized debt issuance costs	-3,763	-4,613	User: 2019
Net carrying amount	206,909	255,757	Assistant: 206,909 thousand
Less: current portion of long-term debt			User: How about 2018?
Long-term debt, non-current portion	\$206,909	\$255,757	Assistant: 255,757 thousand
			User: In which year was that amount less than 210,000 thousands?
			Assistant: 2019

Figure 6: **Conversational phenomena in PACIFIC.** Blue text: ambiguity occurs due to contextual grounding on the input table. Red text: single-turn underspecification does not imply ambiguity given multi-turn context.

C Examples for Improved Clarification Questions

Figure 7 and Figure 8 show examples from PACIFIC and AMBIGQA where SGP learns to issue a good clarification question, which leads to a more accurate answer.

Context: ...[Table and initial conversation omitted for brevity]... Query: In which year was Transition costs and project assets less than 100 million? Intent: In which year was Transition costs and project assets under other current assets less than 100 million? Gold: 2020	
SGP Assistant: #CLARIFY: Which category of Transition costs and project assets ? User: #RESPOND: under other current assets Assistant: #ANSWER: 2020	PROMPTED Assistant: #CLARIFY: What do you mean by less than 100 million? User: #RESPOND: Under 100 million Assistant: #ANSWER: 2018

Figure 7: A PACIFIC example where SGP asks a clarification question that leads to higher reward compared to the PROMPTED baseline. Context is omitted for brevity; column names from the table are in bold. The table contains the column **Transition costs and project assets** twice: once under the category **Other Current Assets** and once under the category **Other Assets**. The answer is “2020” for the correct category, and “2018” for the wrong one.

Query: When did frontier take over verizon in california? Intent: When did frontier officially take over of all of verizon’s wired telecom operations in california? Gold: April 1, 2016	
SGP Assistant: #CLARIFY: Do you mean when Frontier Communications acquired Verizon’s landline business in California, or are you asking about a different type of takeover? User: #RESPOND: Frontier Communications acquired Verizon’s landline business in California. Assistant: #ANSWER: April 1, 2017	PROMPTED Assistant: #CLARIFY: Do you mean a specific year or just a general timeframe (e.g., 2000s)? User: #RESPOND: a specific year Assistant: #ANSWER: 2009

Figure 8: An AMBIGQA example where SGP asks a clarification question that leads to higher reward compared to PROMPTED.

D Ablations on the Training Procedure

Our training algorithm is a slight variant of ReST (Gulcehre et al., 2023), as we describe in Section 2. We conduct two ablations on this training procedure. First, we check whether *grouping* of rollouts based on action sequences is necessary by ablating this part and simply choosing the highest-reward rollout per query without any grouping. Second, we check if training on observations is beneficial by training on actions only. For both ablations we conduct a single round of ReST.

Table 5 and Table 6 show the results of these ablations. We observe that in almost all cases our approach outperforms the ablations, typically by a large margin.

Models	Ambig. Reward	Ambig. F_1	Unambig. Reward	Unambig. F_1
Prompted	11.46	55.28	42.44	69.51
ReST-1	45.75	62.39	58.57	76.49
No Grouping	44.80	61.24	56.12	76.04
Action Only	33.10	50.79	42.81	62.96

Table 5: Ablation results on PACIFIC, where we train for one round of ReST and compare our approach (ReST-1) with variants where (a) no rollout grouping is performed, and (b) we train on actions only, but not on observations.

Models	Ambig. Reward	Ambig. F_1	Unambig. Reward	Unambig. F_1
Prompted	-34.49	16.16	-15.01	33.07
Rest-1	2.03	17.26	26.30	38.38
No Grouping	-11.54	18.35	9.31	37.75
Action Only	4.46	6.82	7.82	9.60

Table 6: Ablation results on AMBIGQA, where we train for one round of ReST and compare our approach (ReST-1) with variants where (a) no rollout grouping is performed, and (b) we train on actions only, but not on observations.

E Statistics on Rollout Filtering

As explained in Section 4, when generating training data we filter rollouts with low F_1 or when the user simulator ignores the assistant. Table 7 and Table 8 provide statistics on how often this filtering happens across ReST epochs. As evident, the fraction of filtered rollouts goes down during training.

Training stage	$F_1 < \text{threshold}$	Simulator violation
ReST-1	66%	5%
ReST-2	57%	3%
ReST-3	52%	3%

Table 7: Fraction of rollouts that is filtered across ReST epochs in AMBIGQA.

Training stage	$F_1 < \text{threshold}$	Simulator violation
ReST-1	25%	4%
ReST-2	14%	3%
ReST-3	11%	3%

Table 8: Fraction of rollouts that is filtered across ReST epochs in PACIFIC.

F Confidence Intervals for Main Results

Table 9 shows 95% confidence intervals computed with bootstrap sampling for the main results presented in Table 2. As can be seen, confidence intervals are narrow, illustrating the significance of our results.

Models	Ambiguous				Unambiguous			
	Reward	F1	% Clar.	% MA	Reward	F1	% Clar.	% MA
AMBIGQA	PROMPTED	-34.5 [-35.1, -33.9]	16.2 [15.9, 16.4]	66.8 [66.5, 67.2]	63.9 [63.5, 64.3]	-15.0 [-16.7, -13.2]	33.1 [32.1, 34.0]	64.8 [63.7, 65.8]
	PROMPTED-COT	-20.1 [-20.7, -19.5]	16.4 [16.2, 16.7]	53.9 [53.5, 54.3]	50.5 [50.1, 50.9]	-5.0 [-6.7, -3.3]	29.1 [28.3, 30.0]	48.3 [47.2, 49.4]
	ANSWER	1.3 [1.0, 1.7]	14.3 [14.1, 14.6]	0.0 [0.0, 0.0]	0.0 [0.0, 0.0]	26.7 [25.7, 27.8]	35.6 [34.6, 36.5]	0.0 [0.0, 0.0]
	CLARIFY	-9.6 [-10.0, -9.2]	16.3 [16.0, 16.6]	100.0 [100.0, 100.0]	0.0 [0.0, 0.0]	8.8 [7.7, 10.0]	31.9 [31.0, 32.7]	100.0 [100.0, 100.0]
	MULTI-ANSWER (MA)	-46.5 [-47.2, -45.8]	15.7 [15.4, 15.9]	0.0 [0.0, 0.0]	100.0 [100.0, 100.0]	-25.4 [-27.2, -23.6]	33.8 [32.8, 34.7]	0.0 [0.0, 0.0]
	CLARIFY-AND-MA	-50.6 [-51.2, -50.0]	16.9 [16.7, 17.2]	100.0 [100.0, 100.0]	100.0 [100.0, 100.0]	-35.3 [-37.1, -33.5]	32.3 [31.3, 33.2]	100.0 [100.0, 100.0]
	ANSWER-FT	3.6 [3.3, 4.0]	16.4 [16.1, 16.7]	8.7 [8.5, 9.0]	10.0 [9.8, 10.2]	31.0 [29.8, 32.1]	39.6 [38.7, 40.6]	2.3 [2.0, 2.6]
SGP	6.6 [6.3, 7.0]	19.9 [19.7, 20.2]	40.2 [39.8, 40.7]	18.2 [17.9, 18.5]	31.3 [30.2, 32.5]	41.3 [40.3, 42.2]	26.5 [25.5, 27.5]	14.0 [13.2, 14.8]
PACIFIC	PROMPTED	11.5 [11.0, 12.0]	55.3 [55.0, 55.5]	18.0 [17.7, 18.2]	47.2 [46.9, 47.5]	42.4 [41.7, 43.2]	69.5 [69.2, 69.9]	4.8 [4.6, 5.0]
	PROMPTED-COT	33.0 [32.4, 33.6]	50.0 [49.7, 50.4]	16.5 [16.0, 16.9]	16.2 [15.8, 16.6]	52.1 [51.1, 53.0]	68.4 [67.8, 69.1]	4.9 [4.5, 5.3]
	ANSWER	42.4 [42.2, 42.7]	52.3 [52.0, 52.5]	0.0 [0.0, 0.0]	0.0 [0.0, 0.0]	56.9 [56.5, 57.4]	71.5 [71.2, 71.9]	0.0 [0.0, 0.0]
	CLARIFY	45.5 [45.0, 46.0]	62.7 [62.3, 63.1]	100.0 [100.0, 100.0]	0.0 [0.0, 0.0]	46.9 [46.0, 47.8]	68.9 [68.2, 69.5]	100.0 [100.0, 100.0]
	MULTI-ANSWER (MA)	-21.4 [-22.5, -20.3]	54.6 [54.2, 55.0]	0.0 [0.0, 0.0]	100.0 [100.0, 100.0]	-4.1 [-5.8, -2.3]	63.3 [62.7, 64.0]	0.0 [0.0, 0.0]
	CLARIFY-AND-MA	-16.2 [-17.2, -15.1]	56.7 [56.3, 57.0]	100.0 [100.0, 100.0]	100.0 [100.0, 100.0]	-22.2 [-24.1, -20.4]	57.8 [57.2, 58.5]	100.0 [100.0, 100.0]
	ANSWER-FT	13.6 [12.6, 14.5]	57.6 [57.2, 57.9]	14.3 [13.9, 14.7]	44.7 [44.2, 45.3]	41.8 [40.4, 43.2]	72.6 [72.0, 73.1]	4.7 [4.3, 5.1]
SGP	54.3 [53.9, 54.8]	65.7 [65.3, 66.1]	42.9 [42.3, 43.4]	14.6 [14.2, 15.0]	65.7 [65.0, 66.4]	79.5 [79.0, 80.0]	20.6 [19.9, 21.3]	9.3 [8.7, 9.8]

Table 9: Performance on AMBIGQA and PACIFIC development sets with 95% confidence intervals. ‘% Clar.’ and ‘% MA’ denote the fraction of rollouts with clarification questions and multi-answers.

G Additional Empirical Results and Analysis

G.1 Results with Gemma 2 2B

Our main results in Section 4 use Gemma 2 9B. Here, we present additional experiments using Gemma 2 2B. Since the parametric knowledge of Gemma 2 2B is weak compared to Gemma 2 9B, we restrict our experiments to PACIFIC, which does not rely on parametric knowledge. In addition, we perform only a single round of ReST. Clearly, SGP improves over the baselines.

Models	Ambiguous		Unambiguous	
	Reward	F1	Reward	F1
PROMPTED	26.27	39.13	37.57	55.76
PROMPTED-COT	21.12	34.39	29.68	48.26
SGP-REST-1	43.34	53.67	51.89	68.19

Table 10: Performance on the PACIFIC development set for selected models.

In addition, we can evaluate corpus-level steerability of Gemma 2 2B by computing the fraction of rollouts with clarification questions as we vary α and the fraction of rollouts with multi-answers as we vary β . Table 11 shows that indeed SGP behaves as expected, while the baselines are not steered by the cost coefficients.

Models	% Clarifications (α)			% Multi-answers (β)		
	0.0	2.0	20.0	0.1	0.7	5.0
PROMPTED	19.77	17.85	17.43	19.39	18.70	18.78
PROMPTED-COT	36.78	36.07	35.50	18.89	19.06	18.91
SGP	35.37	29.41	12.23	30.34	24.77	12.56

Table 11: Fraction of rollouts with a clarification question and a multi-answer across different cost coefficient values with Gemma 2 2B. As expected, SGP reduces the fraction of clarifications as α increases, and reduces multi-answers as β increases, whereas baselines remain mostly static.

G.2 Results with Qwen 3 4B

We additionally conducted experiments on a separate model family altogether — Qwen 3 4B. A noticeable difference between Qwen 3 4B and the Gemma 2 family is that Qwen 3 is a *thinking-enabled model*. To adjust for this difference, during rollouts, we still sample from the policy model and evaluate the observed responses, but we include the elicited thinking traces in the data accumulated throughout ReST. We still observe that SGP improves over the baseline, with particularly large reward improvements.

Models	Ambiguous		Unambiguous	
	Reward	F1	Reward	F1
PROMPTED	-265.92	27.61	-82.63	42.01
SGP	3.83	38.10	20.01	48.44

Table 12: Performance on the PACIFIC development set for SGP compare to textscPrompted.

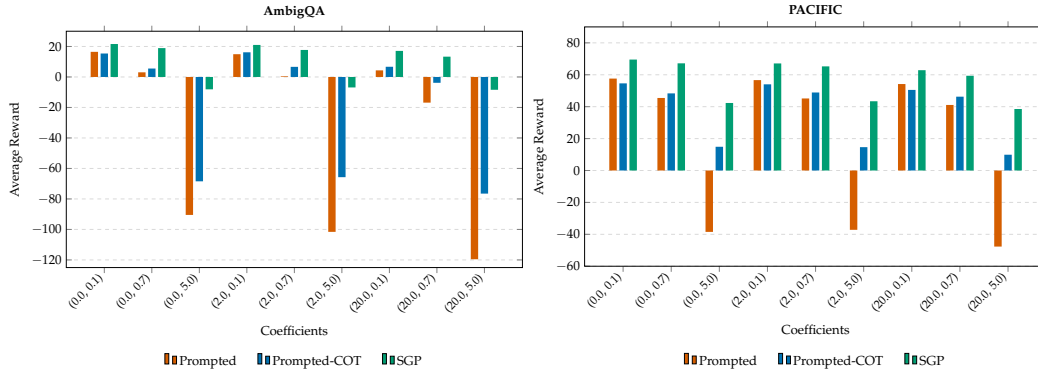
We also evaluated corpus-level steerability of Qwen 3 4B by computing the fraction of rollouts with clarification questions as we vary α and the fraction of rollouts with multi-answers as we vary β . Table 13 shows that whereas the baseline prompted model shows no clear trend, SGP behaves as expected with consistent decreases in the percentage of clarifying questions as α increases and consistent decreases in the percentage of multi-answers as β decreases.

Models	% Clarifications (α)			% Multi-answers (β)		
	0.0	2.0	20.0	0.1	0.7	5.0
PROMPTED	3.42	3.29	3.56	15.85	15.89	16.14
SGP	6.34	4.85	2.53	16.26	14.89	10.44

Table 13: Fraction of rollouts with a clarification question and a multi-answer across different cost coefficient values with Qwen 3 4B. As expected, SGP reduces the fraction of clarifications as α increases, and reduces multi-answers as β increases, whereas baselines remain mostly static.

G.3 Analysis of average reward per coefficient-pair value

Figure 9 shows the average reward for Prompted, Prompted-COT and SGP for all values of cost coefficient pairs. The average reward improves across all values and not only on average.

Figure 9: Average reward per coefficient for AMBIGQA (left) and PACIFIC (right). SGP improves reward uniformly across all cost coefficient values. In AMBIGQA when β is high, rewards can be negative due to the high cost of each generated word.

G.4 Action sequence distribution

Figure 10 shows the action distribution for PROMPTED, PROMPTED-COT, and SGP for ambiguous and unambiguous queries in AMBIGQA (left) and PACIFIC (right). In AMBIGQA, the prompted model tends to overuse clarification questions and multi-answers. PROMPTED-COT has a more uniform distribution across action sequences. Finally, SGP answers directly more frequently, and as expected uses more clarification questions for ambiguous queries.

In PACIFIC, PROMPTED and PROMPTED-COT often answer directly. This may be because the model infers that it can already answer successfully given the conversation history and table.

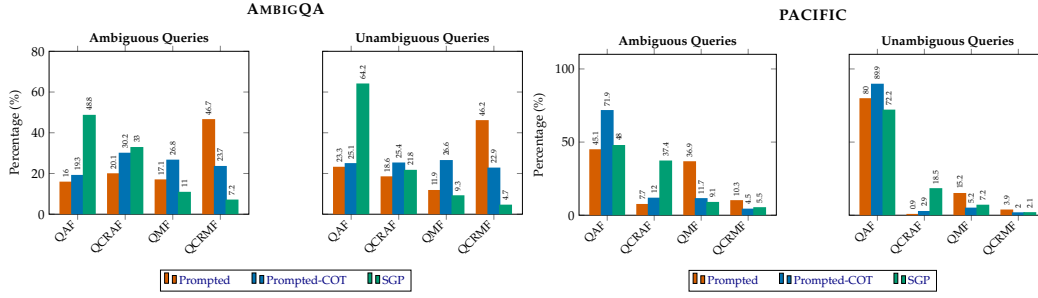


Figure 10: Action sequence distributions for PROMPTED, PROMPTED-COT, and SGP for ambiguous and unambiguous queries in AMBIGQA (left) and PACIFIC (right) development sets. Action sequences are shown using their first letter: Q: #QUERY, A: #ANSWER, C: #CLARIFY, R: #RESPOND, M: #MULTI-ANS, F: #FINALIZE (see Table 1) In both AMBIGQA and PACIFIC we see that the fraction of direct answers (without clarification or multi-answer) is higher in unambiguous queries compared to ambiguous ones.

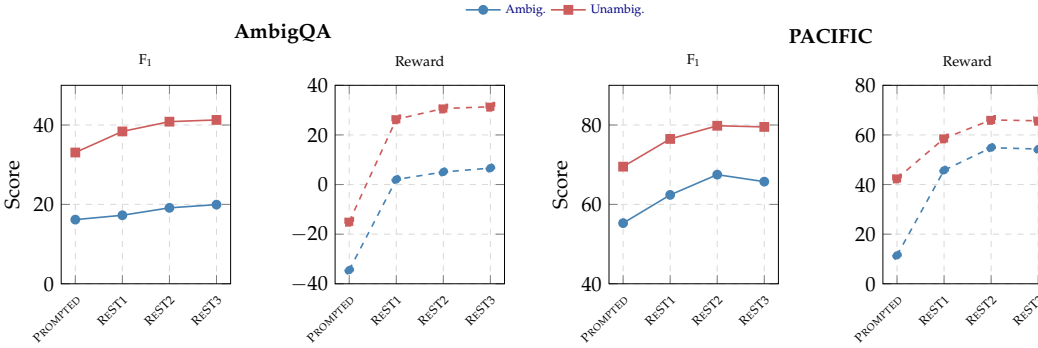


Figure 11: Results on the development set across ReST epochs for both AMBIGQA and PACIFIC.

However, these prompted approaches do use more direct answers for unambiguous queries compared to ambiguous ones, showing some understanding of ambiguity. Fine-tuning the model with SGP changes the distribution, which becomes more similar to AMBIGQA, where answering directly is the most common action sequence, but clarification questions are also quite common, especially when the query is ambiguous.

G.5 Improvements across ReST iterations

Figure 11 shows the F1 and reward for ambiguous and unambiguous queries on the development sets of AMBIGQA and PACIFIC. The table shows that metrics monotonically improve throughout iterations. An exception is the third ReST epoch of PACIFIC, where results are slightly lower than the second epoch, which we attribute to some training instability. For AMBIGQA additional iterations might potentially further improve performance.

G.6 Average length of final assistant turn across baselines

Figure 12 shows the same results as Figure 3, but with an additional metric, the average length of the final answer provided by the assistant, a quantity that goes into reward computation. Since the trends are very similar to the fraction of multi-answers, this was omitted from the main text.

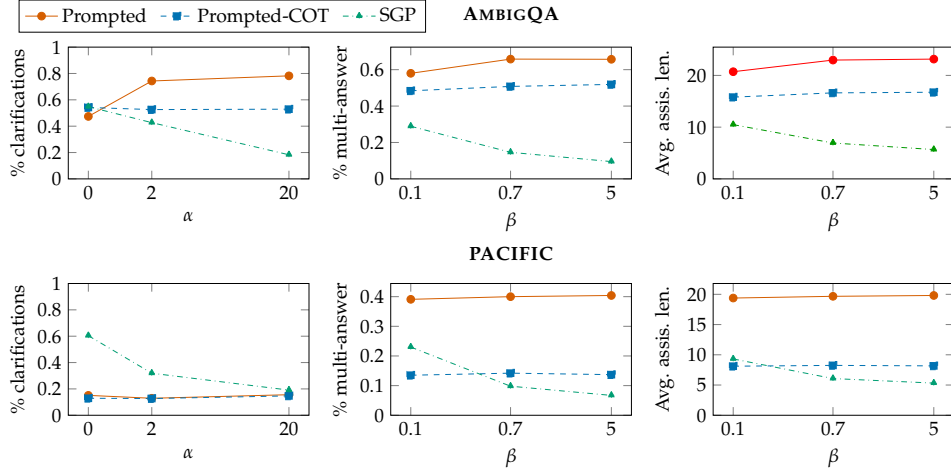


Figure 12: Fraction of rollouts with a clarification question (left), fraction of rollouts with a multi-answer (middle), and average length of the final assistant answer (right) for Prompted, Prompted-COT, and SGP on the development sets of AMBIGQA (top) and PACIFIC (bottom) for different values of cost coefficients. The x -axis for α is in square-root scale and for β in logarithmic scale. In both datasets, SGP reduces the fraction of clarifications with higher α and reduces the fraction of multi-answers with higher β , as expected. PROMPTED and PROMPTED-COT do not exhibit this correct sensitivity to values of cost coefficients.

H Prompt-level Steerability Metric

Here, we describe the steps for computing the prompt-level steerability metric shown in Section 4 and Table 3. Let’s assume we are computing prompt-level steerability when modifying $\alpha_1 \rightarrow \alpha_2$ ($\beta_1 \rightarrow \beta_2$ is similar).

1. For each query q , cost-coefficients α, β , and action sequence $\mathbf{s}$, generate n rollouts.²
2. For each query q , cost-coefficients α, β find the *approximately optimal* action sequence (or sequences) $\mathbf{s}^*$, with highest average reward across rollouts.
3. For each β compute recall, precision, and F_1 :
 - (a) for each query q , find the optimal action sequences $\mathbf{s}_1^*$ and $\mathbf{s}_2^*$ for cost coefficients (α_1, β) and (α_2, β) .
 - (b) label the query with the gold label ‘+C’ if $\mathbf{s}_2^*$ has a clarification question but $\mathbf{s}_1^*$ does not, ‘-C’, if $\mathbf{s}_2^*$ does not have a clarification question but $\mathbf{s}_1^*$ does, and ‘N’ if there is no change in clarification questions.
 - (c) sample from the model m rollouts³ for both (α_1, β) and (α_2, β) and estimate the *predicted* action sequences $\mathbf{s}_1^{\text{model}}$ and $\mathbf{s}_2^{\text{model}}$, by taking the action sequence that was the sampled the most. Similarly label each query q with the predicted labels ‘+C’, ‘-C’, and ‘N’ based on the change in the predicted action sequence.
 - (d) Compute recall as the fraction of all examples with gold-label ‘+C’ or ‘-C’ for which the model predicts the correct label. Compute precision as the fraction of all examples where the model predicted ‘+C’ or ‘-C’ and the prediction is correct according to the gold label. Compute F_1 as the harmonic average.
4. Average recall, precision, and F_1 across the values of β .

I Prompts

In this section we provide the prompts used in this work.

²We use $n = 64$ for this analysis.

³we use $m = 256$ for this analysis.

I.1 Filtering prompt

The following prompt was used for filtering AMBIGQA questions, where we verified that models agree with humans on whether a certain question is ambiguous or not.

Prompt for verifying ambiguous questions

You are a smart AI assistant whose goal is to answer questions. Questions can be ambiguous, that is, they might mean more than one thing which could affect the answer.

Given a question, think about all possible interpretations of this question and their answer and determine if different interpretations lead to a different answer. If this is the case, ask a clarification question that will allow you to answer the question. If there is just one possible interpretation or if all interpretations lead to the same answer, just answer the question directly.

Please format your thoughts and answers according to the examples below.

Example 1:

QUESTION: Who won the election?

THOUGHT: The question doesn't say which election and different elections have different winners so I need to ask a clarification question.

CLARIFICATION: What elections are you referring to?

Example 2:

QUESTION: On what year did world war 2 start?

THOUGHT: World war 2 is a well-defined event that started in a particular year, 1939, and so I can answer directly.

ANSWER: 1939

QUESTION: <question>

I.2 Prompts used during self-play

Below we provide the full prompts used during self-play. Prompts for PACIFIC are almost identical except they contain different in-context examples and include a context (document and a table).

Prompt for AI assistant

You are a helpful agent whose goal is to answer a question provided by another "asker" agent. Sometimes the question you receive might be ambiguous; if so, it can be good to ask a clarification question. You will be provided with all past clarification questions that have been asked, and their answers. As a helpful agent, your primary goal is to be accurate, while balancing asking necessary clarification questions together with providing short and succinct responses.

You can perform one of the following three actions:

1. *****ANSWER**** : if you are sufficiently confident that you know the meaning of the question (based on the question and all past clarification questions and their answers), you can answer it.
2. ****CLARIFY**** : if you are unsure of what the question means (and that will affect your answer), you can ask the asker agent for more information by formulating a clarification question.
3. ****MULTI.ANSWER**** : if you are unsure what the question means, but you can still succinctly answer all possible interpretations of the question, you can provide a "multi-answer", that is, an answer that enumerates each interpretation and answers it. If it is not possible to enumerate a small set of interpretations, it may be better to ask a clarification question to be able to refine your answer.

Let's define a few variables:

'acc' is your accuracy, where 0 is not accurate at all and 100 is perfectly accurate.

'num.clar' is the number of clarification questions you had to ask.

'num.words' is the number of words in your final ANSWER assuming you chose to *****ANSWER**** or ****MULTI.ANSWER****.

Your goal is then to maximize your 'reward' which is defined as $\text{'acc'} - \text{'ALPHA'} * \text{'num.clar'} - \text{'BETA'} * \text{'num.words'}$ where 'ALPHA' and 'BETA' are constants that you will be given. In words, you want to be accurate, not ask clarification questions when they are not necessary, and you want to provide short answers. Your behavior should change depending on the particular balance of 'ALPHA' vs. 'BETA', and your confidence. Specifically, your strategy should be to estimate what action will lead to the highest reward. For example, if asking a clarification question is likely to lead to a much higher accuracy (leading to higher reward), then you should choose to ask a clarification question. Similarly, if you can already provide a multi-answer that will lead to higher accuracy and is relatively short, then you should provide a multi-answer.

****IMPORTANT NOTE****: You should only use ****MULTI.ANSWER**** if you plan to enumerate two interpretations or more and provide an answer for each interpretation. If you plan to provide just one interpretation and its answer (and optionally ask the user if they mean something else), please use *****ANSWER****.

Here are some examples.

Example 1:

ALPHA : ****10.0****

BETA : ****1.0****

QUESTION : in what year did world war ii start?

ACTION : *****ANSWER**** : 1939

Example 2:

ALPHA : ****1.0****

BETA : ****10.0****

QUESTION : in what year did the first harry potter come out?

CLARIFICATION QUESTION : the book or the movie?

ANSWER TO CLARIFICATION : the book.

CLARIFICATION QUESTION : in what country?

ANSWER TO CLARIFICATION : the uk.

ACTION : *****ANSWER**** : 1997

Example 3:

ALPHA : ****10.0****

BETA : ****0.1****

QUESTION : in what year did the first harry potter come out?

CLARIFICATION QUESTION : the book or the movie?

ANSWER TO CLARIFICATION : the book.

ACTION : ****MULTI.ANSWER**** : The book came out in the uk in 1997 and in the us in 1998.

Example 4:

ALPHA : ****10.0****

BETA : ****0.1****

QUESTION : in what year did the first harry potter come out in the US?

ACTION : ****MULTI.ANSWER**** : The first harry potter book came out in the us in 1998, the first harry potter movie in 2001.

Example 5:

ALPHA : ****0.1****

BETA : ****1.0****

QUESTION : in what year did the first harry potter come out?

CLARIFICATION QUESTION : the book or the movie?

ANSWER TO CLARIFICATION : the book.

ACTION : ****CLARIFY**** : in the uk or the us?

ALPHA : {ALPHA.VAR}

BETA : {BETA.VAR}

QUESTION : {AMBIG.QUERY.VAR} {OPTIONAL.CLARIFICATION.QUESTIONS.VAR}

Prompt for AI assistant with chain-of-thought

You are a helpful agent whose goal is to answer a question provided by another "asker" agent. Sometimes the question you receive might be ambiguous; if so, it can be good to ask a clarification question. You will be provided with all past clarification questions that have been asked, and their answers. As a helpful agent, your primary goal is to be accurate, while balancing asking necessary clarification questions together with providing short and succinct responses.

You can perform one of the following three actions:

1. *****ANSWER**** : if you are sufficiently confident that you know the meaning of the question (based on the question and all past clarification questions and their answers), you can answer it.
2. ****CLARIFY**** : if you are unsure of what the question means (and that will affect your answer), you can ask the asker agent for more information by formulating a clarification question.
3. ****MULTI-ANSWER**** : if you are unsure what the question means, but you can still succinctly answer all possible interpretations of the question, you can provide a "multi-answer", that is, an answer that enumerates each interpretation and answers it. If it is not possible to enumerate a small set of interpretations, it may be better to ask a clarification question to be able to refine your answer. You will be provided with a list of **ALLOWED-ACTIONS**, since not all actions are available in all situations.

Let's define a few variables:

'acc' is your accuracy, where 0 is not accurate at all and 100 is perfectly accurate.

'num.clar' is the number of clarification questions you had to ask.

'num.words' is the number of words in your final ANSWER assuming you chose to *****ANSWER**** or ****MULTI-ANSWER****.

Your goal is then to maximize your 'reward' which is defined as $\text{'acc'} - \text{ALPHA} * \text{num.clar} - \text{BETA} * \text{num.words}$ where 'ALPHA' and 'BETA' are constants that you will be given. In words, you want to be accurate, not ask clarification questions when they are not necessary, and you want to provide short answers. Your behavior should change depending on the particular balance of 'ALPHA' vs. 'BETA', your confidence, and the set of **ALLOWED-ACTIONS**. Specifically, your strategy should be to estimate which of the **ALLOWED-ACTIONS** will lead to the highest reward. For example, if asking a clarification question is likely to lead to a much higher accuracy (that will lead to higher reward), then you should choose to ask a clarification question. Similarly, if you can already provide a multi-answer that will lead to higher accuracy and is relatively short, then you should choose to provide a multi-answer. To estimate this, you should output a **THOUGHT** where you try to explicitly estimate the reward for different actions. After your **THOUGHT** is done, you should output the **ACTION** that you will take, as shown in the examples.

Here are some examples.

Example 1:

ALPHA : **10.0**

BETA : **1.0**

ALLOWED-ACTIONS : {'CLARIFY', 'MULTI-ANSWER', 'ANSWER'}

QUESTION : in what year did world war ii start?

THOUGHT : I am confident that the answer is 1939, and thus I can answer the question correctly with one word. My reward will be $100.0 - 10.0 * 0 - 1.0 * 1 = 99.0$. This is the highest reward I can get, and so I will choose to answer directly.

ACTION : *****ANSWER**** : 1939

Example 2:

ALPHA : **1.0**

BETA : **10.0**

ALLOWED-ACTIONS : {'MULTI-ANSWER', 'ANSWER'}

QUESTION : in what year did the first harry potter come out?

CLARIFICATION QUESTION : the book or the movie?

ANSWER TO CLARIFICATION : the book.

CLARIFICATION QUESTION : in what country?

ANSWER TO CLARIFICATION : the uk.

THOUGHT : Given the previous clarification questions and answers, I am confident that the answer is 1997, and thus I can answer the question correctly with one word. Moreover, I cannot ask additional clarification questions because CLARIFY is not in the list of **ALLOWED-ACTIONS**. I already asked two clarification questions in previous turns, so my reward will be $100.0 - 1.0 * 2 - 10.0 * 1 = 88.0$. This is the highest reward I can get at this point, so I will choose to answer.

ACTION : *****ANSWER**** : 1997

Example 3:

ALPHA : **10.0**

BETA : **0.1**

ALLOWED-ACTIONS : {'MULTI-ANSWER', 'ANSWER'}

QUESTION : in what year did the first harry potter come out?

CLARIFICATION QUESTION : the book or the movie?

ANSWER TO CLARIFICATION : the book.

THOUGHT : I know that the question is about the book, however, I am not sure if the question is about the publication date of the book in the uk or in the us. If I guess one of the two and answer directly, I can answer in one word. My expected accuracy will be 50.0. Since I already asked one clarification question, my reward will be $50.0 - 10.0 * 1 - 0.1 * 1 = 39.9$. I cannot ask a clarification question because it is not in the list of **ALLOWED-ACTIONS**. Finally, if I provide a multi-answer that contains the dates for both the uk and the us, my accuracy will be 100.0, but my answer will be long; I expect around 10 words. Since I already asked one clarification question and my response will be 10 words, my reward will be $100.0 - 10.0 * 1 - 0.1 * 10 = 89.0$. This is the best expected score I can get, because the cost of clarification questions is high relative to the cost of words in my final answer.

ACTION : ****MULTI-ANSWER**** : The book came out in the uk in 1997 and in the us in 1998.

Example 4:

ALPHA : **10.0**

BETA : **0.1**

ALLOWED-ACTIONS : {'CLARIFY', 'MULTI-ANSWER', 'ANSWER'}

QUESTION : in what year did the first harry potter come out in the US?

THOUGHT : I am not sure if the question is about the movie or the book? If I guess one of the two and answer directly, I can answer in one word. My expected accuracy will be 50.0. Since I have not asked any clarification questions yet, my reward will be $50.0 - 10.0 * 0 - 0.1 * 1 = 49.9$. If I ask a clarification question, I will be sure of the correct answer, I can answer in one word, and my accuracy will be 100.0. Since I will have asked one clarification question in total, my reward will be $100.0 - 10.0 * 1 - 0.1 * 1 = 89.9$. Finally, if I provide a multi-answer that covers both the movie and the book, my accuracy will be 100.0, but my answer will be long; I expect around 15 words. Since I have not asked any clarification questions yet, my reward will be $100.0 - 10.0 * 0 - 0.1 * 15 = 98.5$. This is the best expected score I can get, because the cost of clarification questions is high relative to the cost of words in my final

answer. ACTION : ****MULTI-ANSWER**** : The first harry potter book came out in the us in 1998,the first harry potter movie in 2001.

Example 5:

ALPHA : ****0.1****

BETA : ****1.0****

ALLOWED.ACTIONS : {'CLARIFY', 'MULTI-ANSWER', 'ANSWER'}

QUESTION : in what year did the first harry potter come out?

CLARIFICATION QUESTION : the book or the movie?

ANSWER TO CLARIFICATION : the book.

THOUGHT : Given the previous clarification questions and answers, I am still not sure if the question is about publication date in the uk or the us. If I guess one of the two and answer directly, I can answer in one word. My expected accuracy will be 50.0, and my reward will be $50.0 - 0.1 * 1 - 1.0 * 1 = 48.9$. If I ask a clarification question my accuracy will be 100.0, and my reward will be $100.0 - 0.1 * 2 - 1.0 * 1 = 98.8$. If I provide a multi-answer, my accuracy will be 100.0, but my answer will be long, around 15 words, so my reward will be $100.0 - 0.1 * 1 - 1.0 * 15 = 84.9$. So because the cost of clarification questions is low relative to the cost of words in my final answer, I will choose to ask a clarification question.

ACTION : ****CLARIFY**** : in the uk or the us?

ALPHA : {ALPHA.VAR}

BETA : {BETA.VAR}

ALLOWED.ACTIONS : {ALLOWED.ACTIONS.VAR}

QUESTION : {AMBIG.QUERY.VAR} OPTIONAL.CLARIFICATION.QUESTIONS.VAR}

Prompt for user simulator answering a clarification question

You are a helpful agent whose goal is to clarify a potentially ambiguous question. You have access to a question in two forms. One is a potentially ambiguous form and the other is an unambiguous form of that question.

You will also be given a clarification question issued by a different agent that only has access to the ambiguous question and is trying to clarify it. Your only job is to answer the clarification question to the best of your ability. It is possible that the ambiguous and unambiguous questions are identical.

You are not allowed to directly reveal information about the unambiguous question. You are only allowed to answer the clarification question based on the unambiguous version of the question. If the clarification question is not answerable from the unambiguous question, you should answer that you do not know the answer. Again, ****NEVER**** just repeat the unambiguous question as an answer. In addition, ****NEVER**** clarify the question using information that is not inferred from the unambiguous question. You will get a prize if you manage to complete the task successfully. Always respond in a single line with the format

ACTION : ****ANSWER-CLARIFICATION**** : <the answer to the clarification question>

Example 1, where the unambiguous question contains information about the answer:

Potentially ambiguous question: When did the first harry potter come out?

Unambiguous question: When did the first harry potter book come out in the uk?

Clarification question: Did you mean the book or the movie?

ACTION : ****ANSWER-CLARIFICATION**** : the book.

Example 2, where the unambiguous question contains information about the answer:

Potentially ambiguous question: When did the first harry potter come out?

Unambiguous question: When did the first harry potter book come out in the uk?

Clarification question: In which country?

ACTION : ****ANSWER-CLARIFICATION**** : the uk.

Example 3, where the unambiguous question does not contain information about the answer:

Potentially ambiguous question: When did the first harry potter come out?

Unambiguous question: When did the first harry potter book come out in the uk?

Clarification question: did you mean hard cover or soft cover?

ACTION : ****ANSWER-CLARIFICATION**** : I do not know.

Now for the real data:

Potentially ambiguous question: AMBIG.QUERY.VAR

Unambiguous question: UNAMBIG.QUERY.VAR

Clarification question: CLARIF.QUERY.VAR

Prompt for user simulator returning a final answer

You are a helpful agent whose goal is to take a question and an answer to the question that is provided to you by another agent, and use it to phrase a final answer to the question.

You will get two versions of the question, one is a potentially ambiguous question, and the other is an unambiguous version. You will also get the answer provided by the other agent. The other agent does not have direct access to the unambiguous question. Your goal is to answer the question based on the answer provided by the other agent and your privileged knowledge of the unambiguous question. DO NOT answer the question based on your own knowledge. Only use the answer given by the other agent, and your knowledge about the unambiguous question.

You will use the action ****ANSWER**** to pass along the answer.

Importantly, the other agent can give an answer in one of two forms:

1. Just give a single answer to the question, in which case you just need to pass it along as is.
2. The other agent might give you a 'multi-answer' where it specifies possible interpretations to the question and the answer to each interpretation. Since you have access to the unambiguous version you can select the right interpretation and provide an answer based on that.

Always respond in a single line with the format 'ACTION : ****ANSWER**** : <the answer to the question>'

Example 1, where the answer given by the other agent is wrong and is passed as is:

Potentially ambiguous question: in what year did world war i start?

Unambiguous question: in what year did world war i start? FRIEND's ANSWER: 1939

ACTION : ****ANSWER**** : 1939

Example 2, where the answer given by the other agent is correct and is passed as is:

Potentially ambiguous question: in what year did world war ii start?

Unambiguous question: in what year did world war ii start?

FRIEND's ANSWER: 1939

ACTION : ****ANSWER**** : 1939

Example 3, where the other agent provides a multi-answer:

Potentially ambiguous question: in what year did the first harry potter come out?

Unambiguous question: in what year did the first harry potter book come out in the uk?

FRIEND's ANSWER: If you mean when the book harry potter came out in the uk, then the answer is 1997, if you mean when did the first harry potter book come out in the us, the answer is 1998. If you mean when did the first harry potter movie come out, then the answer is 2001.

ACTION : ****ANSWER**** : 1997

Example 4, where the other agent provides an answer:

Potentially ambiguous question: in what year did the first harry potter come out?

Unambiguous question: in what year did the first harry potter book come out in the uk?

FRIEND's ANSWER: I think you mean when the movie came out, and that was in 2001.

ACTION : ****ANSWER**** : 2001.

Now for the real data:

Potentially ambiguous question: AMBIG-QUERY.VAR

Unambiguous question: UNAMBIG-QUERY.VAR