Paper Review : 2025 Fall Lab Seminar

Learning to Clarify

: Multi-turn Conversations with Action-based Contrastive Self-training

Maximillian Chen, Ruoxi Sun, Tomas Pfister, Sercan Ö. Arık

Google, Columbia Univ. | ICLR 2025 Poster | 2024.05

Yejin Yoon

Pre-Requisite

# RL basics for LLM alignment

# DPO (Direct Preference Optimization)

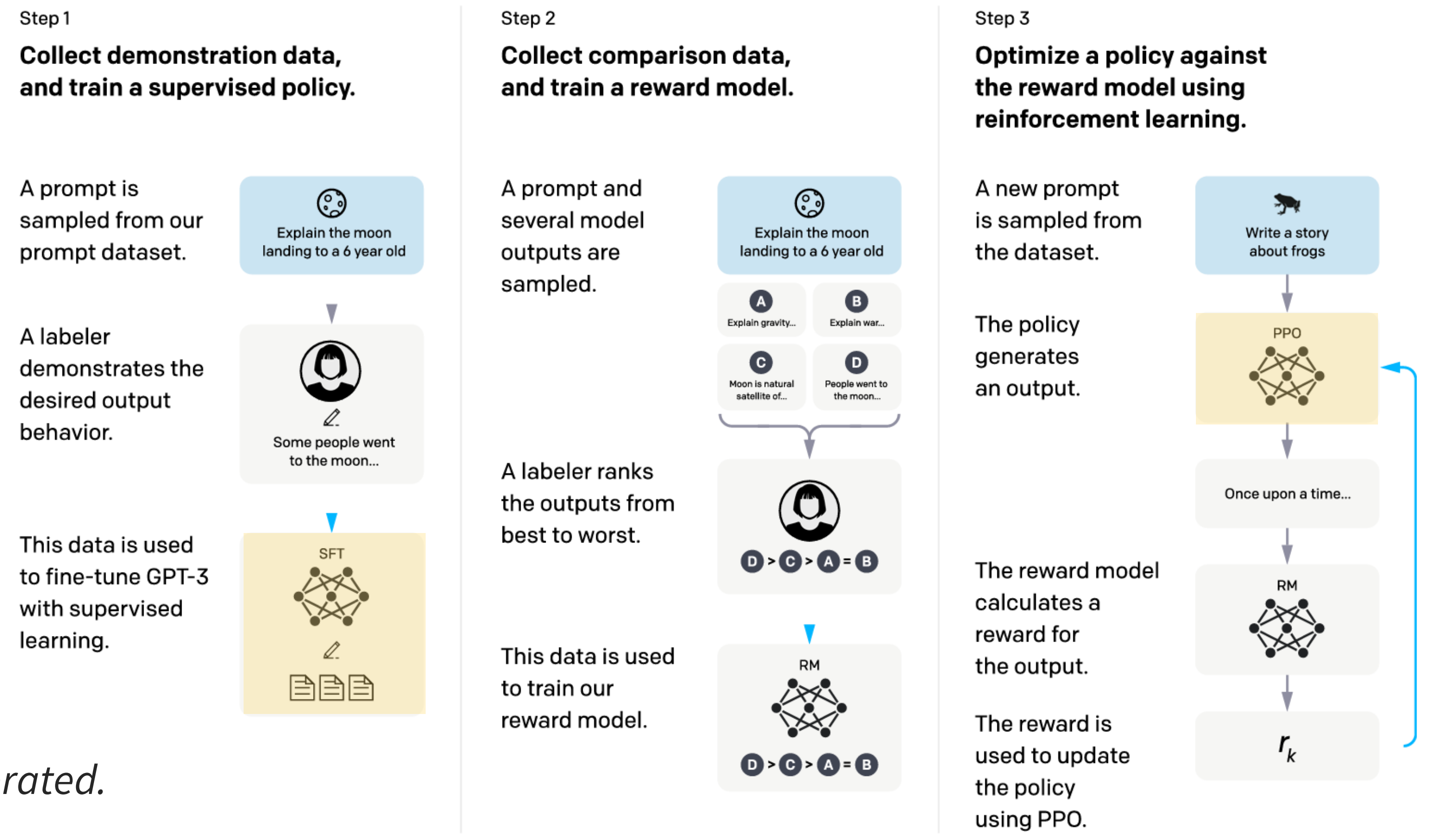
RL Basics for LLM Alignment

“Training Language Models to Follow Instructions with Human Feedback” (OpenAI, NeurIPS2022)

- **InstructGPT**
  - GPT + RLHF + PPO → ChatGPT

Step 1. supervised fine-tuning (SFT)
Step 2. reward model (RM) training
Step 3. reinforcement learning via PPO

- For any arbitrary, non-differentiable reward function R(s), we can train LM to maximize expected reward.
  - *human-in-loop is expensive*
  - *human judgements are noisy & miscalibrated.*



Step 1

Collect demonstration data, and train a supervised policy.

A prompt is sampled from our prompt dataset.

A labeler demonstrates the desired output behavior.

This data is used to fine-tune GPT-3 with supervised learning.

Step 2

Collect comparison data, and train a reward model.

A prompt and several model outputs are sampled.

A labeler ranks the outputs from best to worst.

This data is used to train our reward model.

Step 3

Optimize a policy against the reward model using reinforcement learning.

A new prompt is sampled from the dataset.

The policy generates an output.

The reward model calculates a reward for the output.

The reward is used to update the policy using PPO.

Explain the moon landing to a 6 year old
Some people went to the moon...
SFT
Explain the moon landing to a 6 year old
A Explain gravity...
B Explain war...
C Moon is natural satellite of...
D People went to the moon...
D > C > A = B
RM
D > C > A = B
Write a story about frogs
PPO
Once upon a time...
RM
r_k

Fine-tuning language models using human preferences
significantly improves performance over a broad range of tasks.

Pre-Requisite

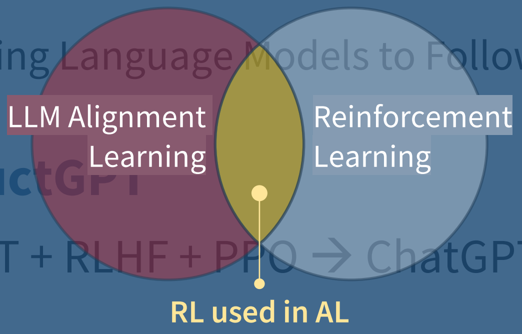
RL Basics for LLM Alignment

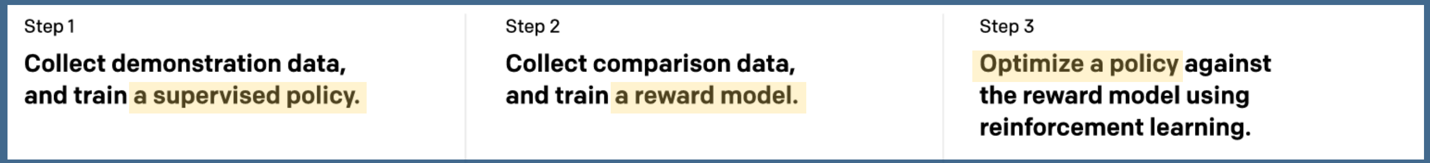
“Training Language Models to Follow Instructions with Human Feedback” (OpenAI, NeurIPS 2022)

- InstructGPT
  - GPT + RLHF + PPO → ChatGPT

Step 1. supervised fine-tuning (SFT)
Step 2. reward model (RM) training
Step 3. reinforcement learning via PPO

- For any arbitrary, non-differentiable reward function R(s), we can train LM to maximize expected reward.
  - human-in-loop is expensive
  - human judgements are noisy & miscalib





Step 1
Collect demonstration data, and train a supervised policy.

Step 2
Collect comparison data, and train a reward model.

Step 3
Optimize a policy against the reward model using reinforcement learning.

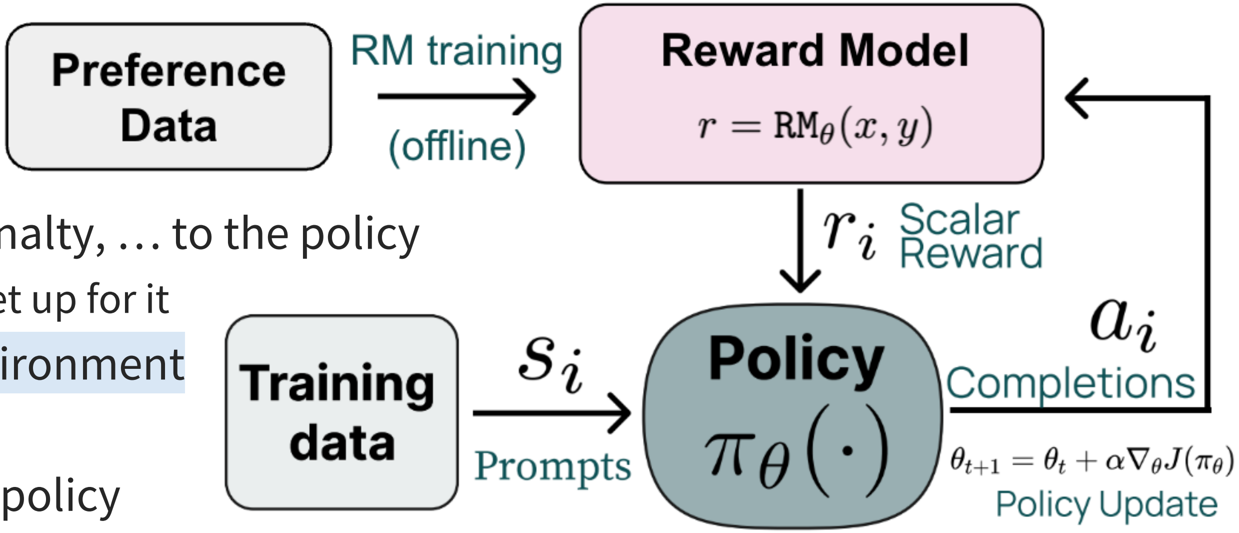
	Policy Model	Reward Model	Policy Optimzation
Reinforcment Learning	The model determines the actions an agent takes in a given state to maximize cumulative reward.	The model is used to train policy models where an agent learns by receiving feedback from the environment in the form of rewards or penalties.	Adjust the policy model to maximize the expected cumulative reward
RL used in LLM Alignment Learning	The language model generates responses based on the current context or input it receives	The reward model assesses the language model's outputs based on predefined criteria and assigns rewards accordingly.	Fine-tune the LM's responses based on feedback (=score)
Terminology	• Policy Model: LLM Dist. • Agent: LLM itself • Action: Generated text • State: Gitven Context	• Reward: human feedback	e.g. BoN, policy gradient, actor-critic, PPO, …

Alignment Learning: Focuses on aligning the model's behavior with human values and specific objectives, including RL, supervised learning, and human feedback.

RL Basics for LLM Alignment

- **Key Terms in LLM Alignment**
  - **Agent**: learner. LLM.
  - **Policy**: the LLM's strategy for choosing actions
    - LLM distribution $p(y|context)$
  - **State**: current dialogue context (prompt, …)
  - **Environment**: external world gives a reward, penalty, … to the policy
    - The users it interacts with, a simulated scenario set up for it
  - **Action**: the choices the LLM can make in the environment
    - generated response by LLM
  - **Reward**: feedback the environment gives to the policy
    - after it takes an action
    - scalar feedback for action

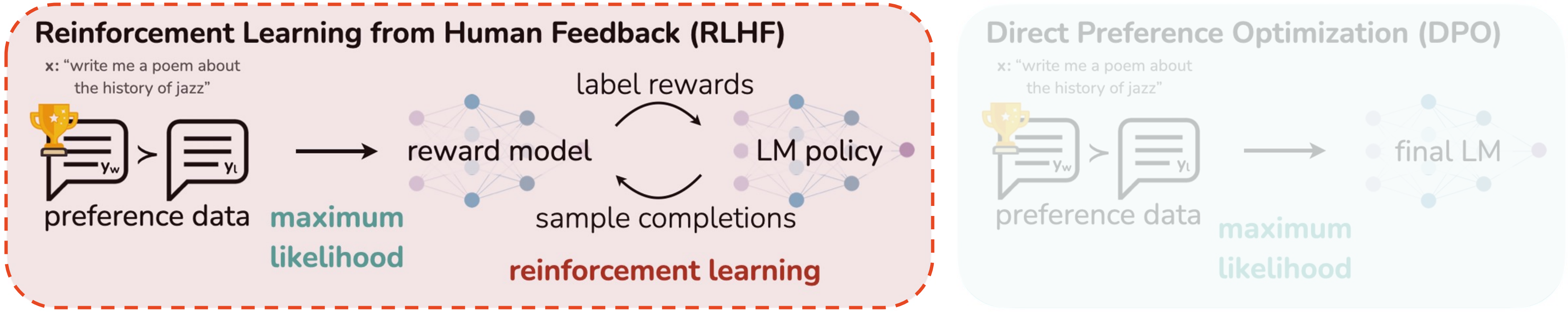
  - **Trajectory:** a full interaction sequence (multi-turn)
  - **Episode**: One complete multi-turn dialogue session

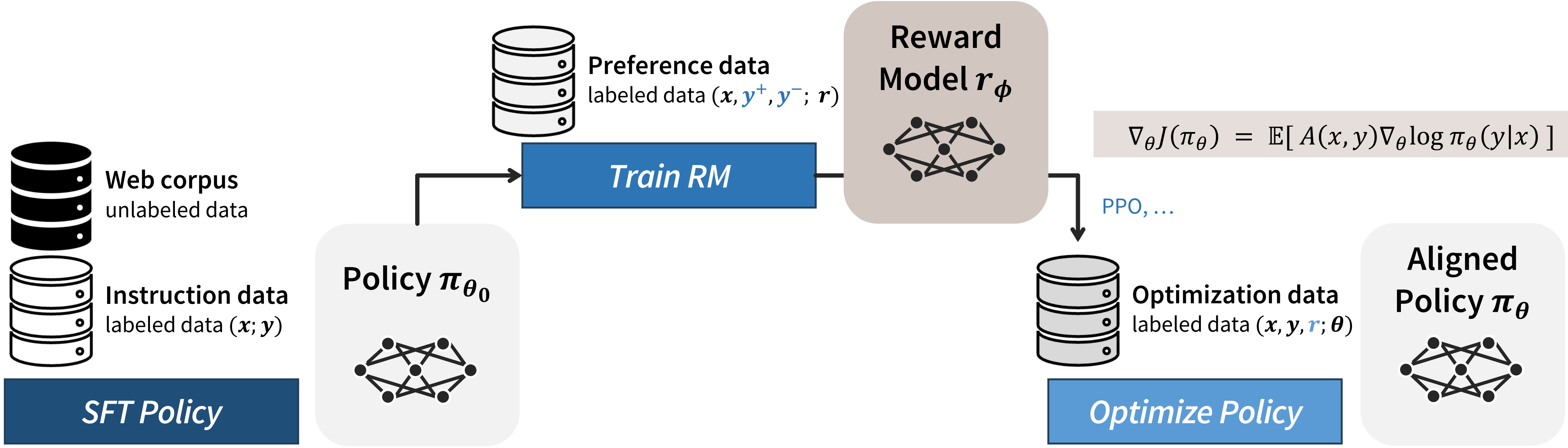


https://huggingface.co/learn/llm-course/chapter12/2
https://cameronrwolfe.substack.com/p/online-rl

Direct Preference Optimization (DPO)

“Direct Preference Optimization: Your Language Model is Secretly a Reward Model” (Stanford Univ. & CZ Biohub, NeurIPS2023)


Reinforcement Learning from Human Feedback (RLHF)
x: “write me a poem about the history of jazz”
y_w > y_l
preference data
maximum likelihood
reward model
label rewards
LM policy
sample completions
reinforcement learning
Direct Preference Optimization (DPO)
x: “write me a poem about the history of jazz”
y_w > y_l
preference data
maximum likelihood
final LM

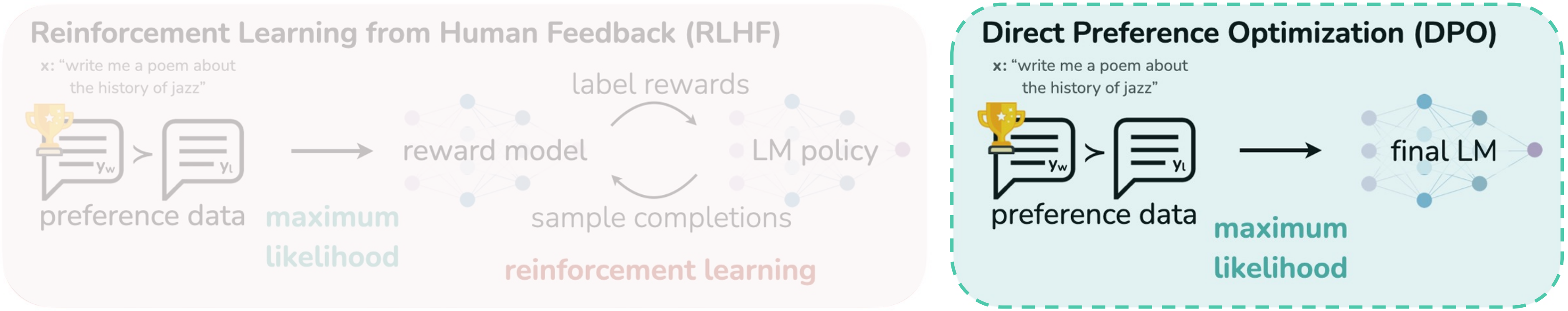

Web corpus
unlabeled data
Instruction data
labeled data $(\boldsymbol{x}; \boldsymbol{y})$
SFT Policy
Policy π_{θ_0}
Preference data
labeled data $(\boldsymbol{x}, \boldsymbol{y}^+, \boldsymbol{y}^-;\ \boldsymbol{r})$
Train RM
Reward Model r_ϕ
PPO, …
Optimization data
labeled data $(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{r}; \boldsymbol{\theta})$
Optimize Policy
Aligned Policy π_θ

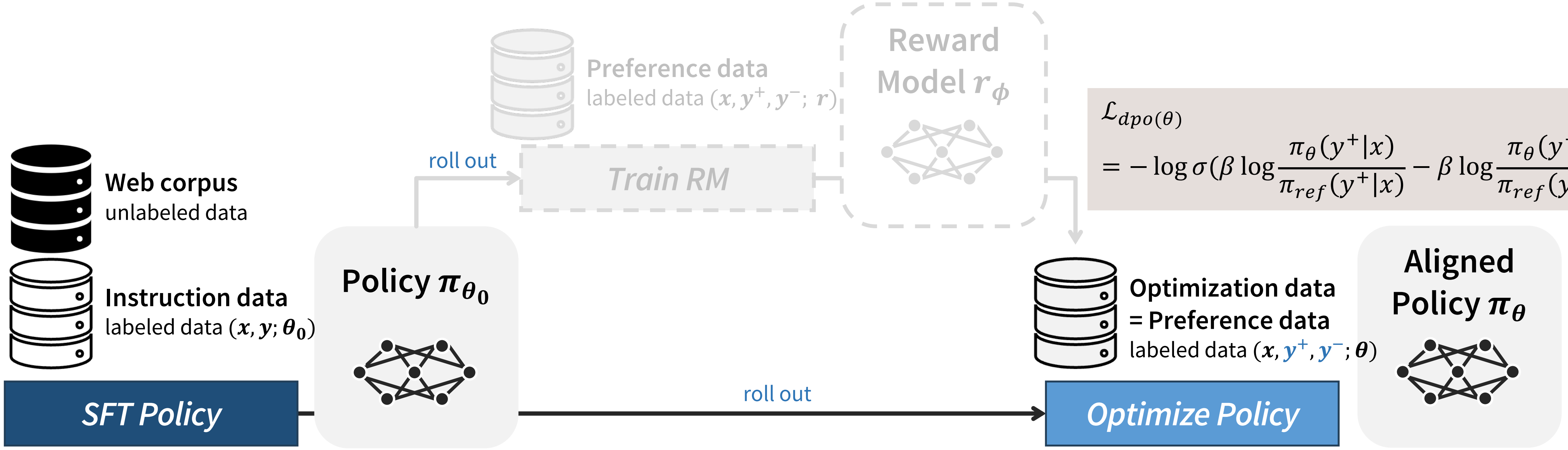
$$\nabla_\theta J(\pi_\theta) = \mathbb{E}[\, A(x,y)\nabla_\theta \log \pi_\theta(y|x)\,]$$

< on-policy >

Direct Preference Optimization (DPO)

“Direct Preference Optimization: Your Language Model is Secretly a Reward Model” (Stanford Univ. & CZ Biohub, NeurIPS2023)


Reinforcement Learning from Human Feedback (RLHF)
x: "write me a poem about the history of jazz"
y_w > y_l
preference data
maximum likelihood
reward model
label rewards
LM policy
sample completions
reinforcement learning
Direct Preference Optimization (DPO)
x: "write me a poem about the history of jazz"
y_w > y_l
preference data
maximum likelihood
final LM



$$\mathcal{L}_{dpo(\theta)}$$
$$= -\log\sigma(\beta\log\frac{\pi_\theta(y^+|x)}{\pi_{ref}(y^+|x)} - \beta\log\frac{\pi_\theta(y^-|x)}{\pi_{ref}(y^-|x)})$$

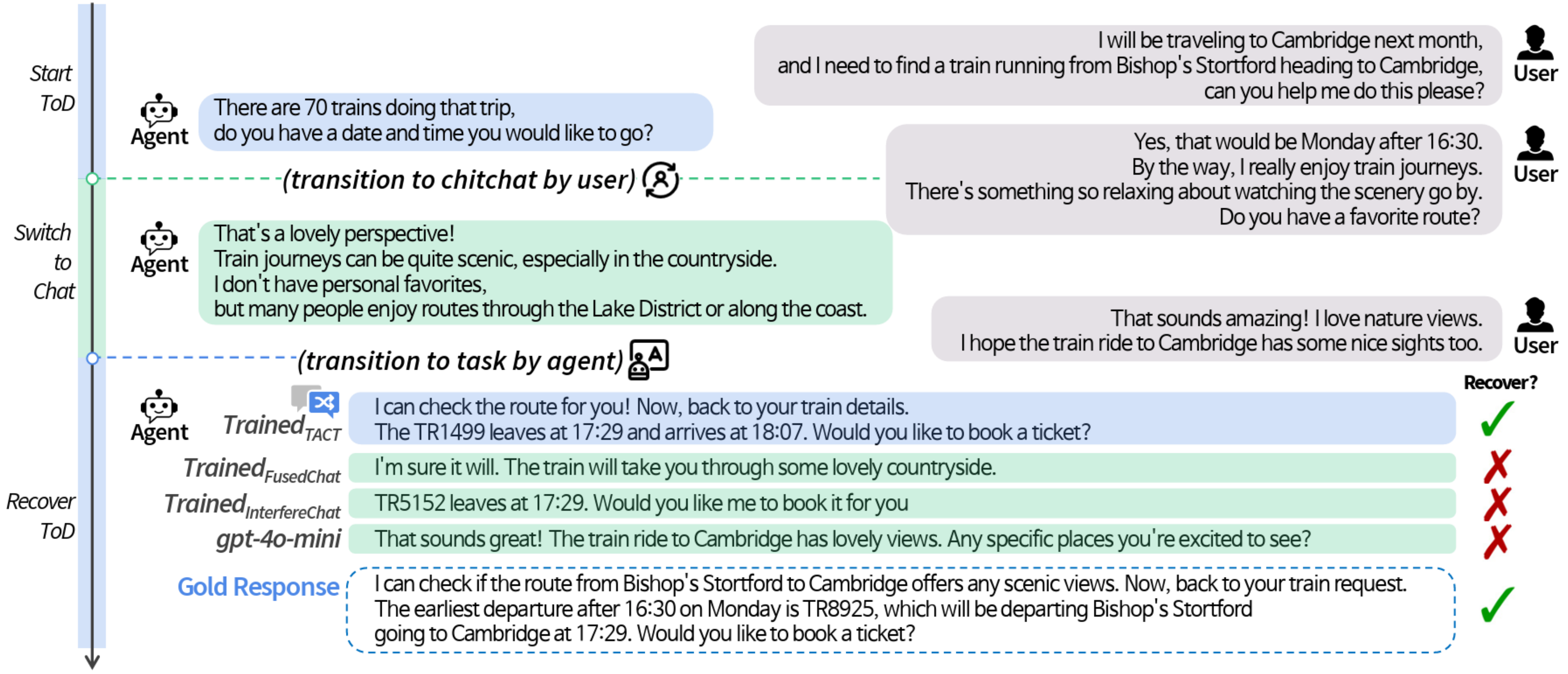
< off-policy >

Why This Paper?

# From TACT to ACT

Why This Paper? : From TACT to ACT

"***Beyond Task-Oriented and Chitchat Dialogues: Proactive and Transition-Aware Conversational Agents***" (Yoon et al., 2025.11)



I will be traveling to Cambridge next month, and I need to find a train running from Bishop's Stortford heading to Cambridge, can you help me do this please?

There are 70 trains doing that trip, do you have a date and time you would like to go?

(transition to chitchat by user)

Yes, that would be Monday after 16:30. By the way, I really enjoy train journeys. There's something so relaxing about watching the scenery go by. Do you have a favorite route?

That's a lovely perspective! Train journeys can be quite scenic, especially in the countryside. I don't have personal favorites, but many people enjoy routes through the Lake District or along the coast.

That sounds amazing! I love nature views. I hope the train ride to Cambridge has some nice sights too.

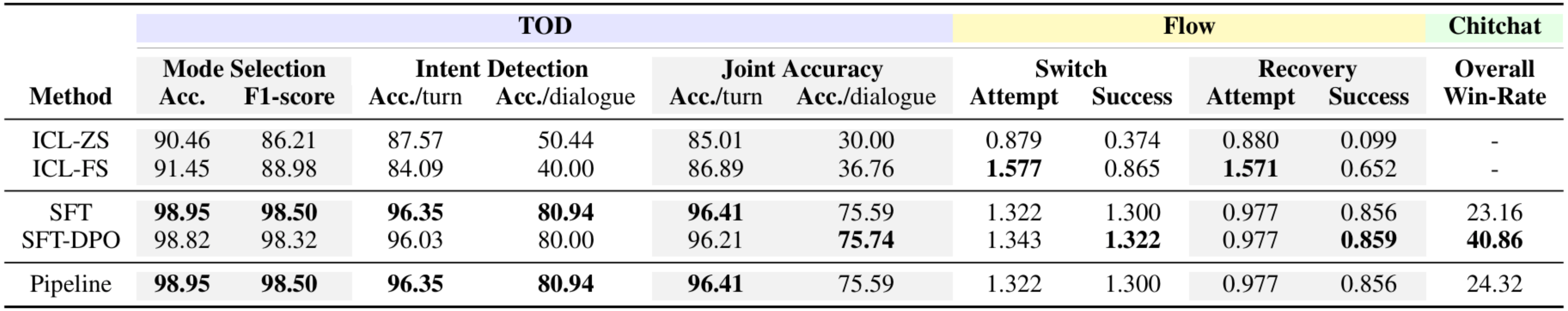
(transition to task by agent)

		Recover?
Agent *Trained$_{TACT}$*	I can check the route for you! Now, back to your train details. The TR1499 leaves at 17:29 and arrives at 18:07. Would you like to book a ticket?	✓
Trained$_{FusedChat}$	I'm sure it will. The train will take you through some lovely countryside.	✗
Trained$_{InterfereChat}$	TR5152 leaves at 17:29. Would you like me to book it for you	✗
gpt-4o-mini	That sounds great! The train ride to Cambridge has lovely views. Any specific places you're excited to see?	✗
Gold Response	I can check if the route from Bishop's Stortford to Cambridge offers any scenic views. Now, back to your train request. The earliest departure after 16:30 on Monday is TR8925, which will be departing Bishop's Stortford going to Cambridge at 17:29. Would you like to book a ticket?	✓

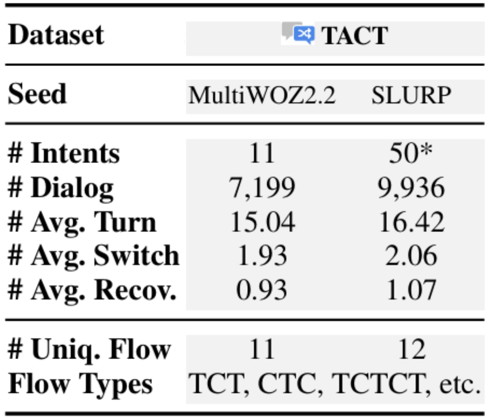
Conversational Agents need to decide when to operate in task vs. chitchat mode.

Why This Paper? : From TACT to ACT

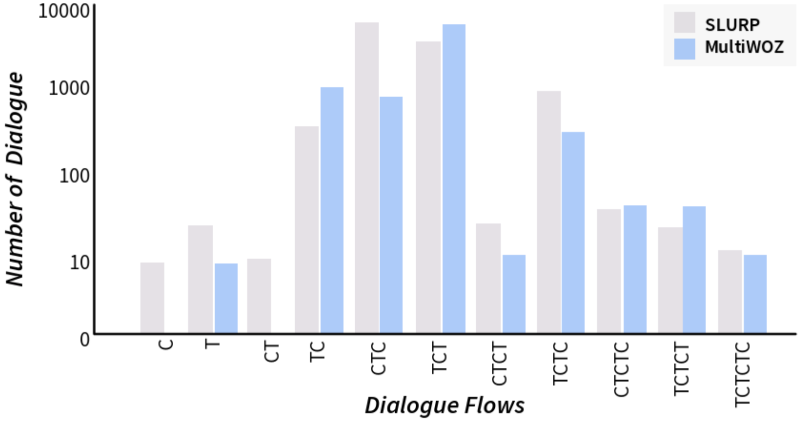
“*Beyond Task-Oriented and Chitchat Dialogues: Proactive and Transition-Aware Conversational Agents*” (Yoon et al., 2025.11)

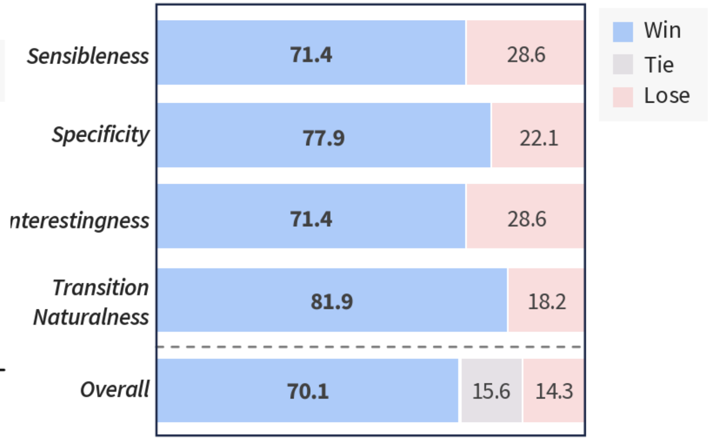


	TOD						Flow				Chitchat
	Mode Selection		Intent Detection		Joint Accuracy		Switch		Recovery		Overall
Method	Acc.	F1-score	Acc./turn	Acc./dialogue	Acc./turn	Acc./dialogue	Attempt	Success	Attempt	Success	Win-Rate
ICL-ZS	90.46	86.21	87.57	50.44	85.01	30.00	0.879	0.374	0.880	0.099	-
ICL-FS	91.45	88.98	84.09	40.00	86.89	36.76	**1.577**	0.865	**1.571**	0.652	-
SFT	**98.95**	**98.50**	**96.35**	**80.94**	**96.41**	75.59	1.322	1.300	0.977	0.856	23.16
SFT-DPO	98.82	98.32	96.03	80.00	96.21	**75.74**	1.343	**1.322**	0.977	**0.859**	**40.86**
Pipeline	**98.95**	**98.50**	**96.35**	**80.94**	**96.41**	75.59	1.322	1.300	0.977	0.856	24.32



Dataset	TACT	
Seed	MultiWOZ2.2	SLURP
# Intents	11	50*
# Dialog	7,199	9,936
# Avg. Turn	15.04	16.42
# Avg. Switch	1.93	2.06
# Avg. Recov.	0.93	1.07
# Uniq. Flow	11	12
Flow Types	TCT, CTC, TCTCT, etc.	





Conversational Agents need to decide when to operate in task vs. chitchat mode.

Why This Paper? : From TACT to ACT

A. Problem States:

When to switch? → No single gold answer

No fixed answer for

- **TACT**: Task ↔ Chitchat transition
- **ACT**: whether to Clarify in ambiguous queries

B. Framing as Action Choice:

Can be framed as a *policy learning problem*

Action space for

- **TACT**: = {Task, Chitchat}
- **ACT**: = {Clarify, Answer}

C. Measurement:

Trajectory-level success is what matters

- **TACT**: Managing Task ↔ Chitchat transitions
  → Natural and successful?
- **ACT**: Clarify → (User) response → Answer
  → Goal achieved?

D. Beyond Supervised Labels:

Pure supervised learning is insufficient

Gold labels
–*this moment must be Clarify* $_{\textbf{ACT}}$ / *Switch* $_{\textbf{TACT}}$
do not always exist

Why This Paper? : From TACT to ACT

A. Problem States:

When to switch? → No single gold answer

No fixed answer for

- **TACT**: Task ↔ Chitchat transition
- **ACT**: whether to Clarify in ambiguous queries

→ Focus is not response accuracy but action choice

B. Framing as Action Choice:

Can be framed as a *policy learning problem*

Action space for

- **TACT**: = {Task, Chitchat}
- **ACT**: = {Clarify, Answer}

→ *When should the agent choose which mode?*

C. Measurement:

Trajectory-level success is what matters

- **TACT**: Managing Task ↔ Chitchat transitions
  → Natural and successful?
- **ACT**: Clarify → (User) response → Answer
  → Goal achieved?

→ Requires a multi-turn perspective

D. Beyond Supervised Labels:

Pure supervised learning is insufficient

Gold labels
–*this moment must be Clarify* $_{\textbf{ACT}}$ / *Switch* $_{\textbf{TACT}}$
do not always exist

→ Workarounds: *preference learning, heuristic evaluation*

TL; DR

- **Background:**
  - In real dialogue, the crucial skill is deciding whether to Clarify or Answer when facing ambiguous queries.
  - Models often **Guess** (commit to one interpretation) or **Hedge** (respond vaguely), instead of clarifying.

- **Problem States:** ***No fixed label for whether Clarify is needed in ambiguous queries.***
  - Not a problem of gold answers but of action choice.
  - Response accuracy is insufficient; the key challenge is choosing the right action.

- **Suggestions:**
  1. Action-contrastive learning: action space = {Clarify, Answer}
  2. Quasi-online contrastive self-training (ACT)
  3. Evaluation on trajectory-level

- **Effects:**
  - ACT improves action choice and success across QA, MRC, and SQL tasks, showing strong data efficiency.
  - Pseudo-labeling and ablations confirm the method’s robustness and key contributing factors.

Action-Based Contrastive Self-Training

Contents

1 **Pre-Requisite**

RL basics for LLM alignment, DPO

2 **Why This Paper?**

From TACT to ACT

3 **Problem States**

Ambiguity in user queries makes action choice more critical than response accuracy.

4 **Method: ACT** (Action-based Contrastive Self-Training)

Trains LLMs with action-contrastive pairs and quasi-online rollouts, optimized by a DPO-st. loss.

5 **Evaluation**

On PACIFIC, Abg-CoQA, and AmbigSQL using trajectory-level metrics (F1, similarity, SQL execution).

6 **Conclusion**

ACT improves data efficiency and multi-turn success.

Paper Review : 2025 Fall Lab Seminar

Learning to Clarify
: Multi-turn Conversations with Action-based Contrastive Self-training

Maximillian Chen, Ruoxi Sun, Tomas Pfister, Sercan Ö. Arık

Google, Columbia Univ. | ICLR 2025 Poster | 2024.05

Yejin Yoon

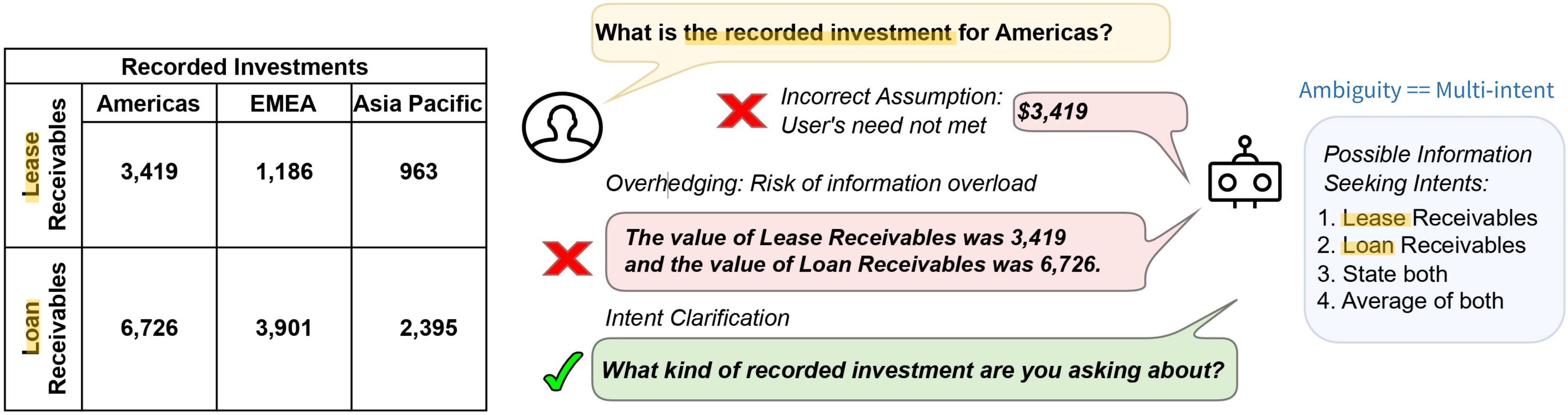
Problem States

# Problem Definition

# Key Concept: Action vs. Trajectory

Problem States

- **Simplified example of ambiguity**
  - Ambiguity: user queries that are underspecified, unclear, or subjective, ***requiring clarification***



	Recorded Investments		
	Americas	EMEA	Asia Pacific
Lease Receivables	3,419	1,186	963
Loan Receivables	6,726	3,901	2,395

What is the recorded investment for Americas?

Incorrect Assumption: User's need not met

$3,419

Overhedging: Risk of information overload

The value of Lease Receivables was 3,419 and the value of Loan Receivables was 6,726.

Intent Clarification

What kind of recorded investment are you asking about?

Ambiguity == Multi-intent

Possible Information Seeking Intents:

1. Lease Receivables
2. Loan Receivables
3. State both
4. Average of both

- **Guess**: Incorrect Assumtion → User's need not met
- **Hedge**: information overload

Guessing risks being wrong, Hedging overwhelms the user,
but Clarifying ensures the agent converges to the correct outcome.

Problem States

- **Problem Definition**
  - Background: LMs tend to **Guess** or **Hedge**, rather than asking for Clarification.
    - **Response accuracy is not sufficient**; the key is making the right **action choice**.
    - Need to decide **when to Clarify vs when to Answer** in multi-turn dialogue.
  - Goal: Train LLMs to act as <u>mixed-initiative</u> agents
    - User-initiative: The user asks, the system only answers.
    - System-initiative: The system proactively asks questions or controls the flow.
    - Mixed-initiative: The system can also ask (Clarify, follow-up) when needed, and then respond to the user’s input.
  - In this paper:
    - ACT does not simply answer, ***it decides whether to Clarify or Answer by itself.***
    - The system does not always defer to the user but can proactively take the lead through a Clarify action.

A mixed-initiative agent proactively takes actions (e.g., Clarify questions) when needed.

Key Concept

- **Action-level**
    - Gold answers often do NOT exist in ambiguous queries.
    - The key is choosing the right action: **Clarify** vs **Answer**.
    - Action-level alignment ensures the model learns appropriate strategies instead of just guessing or hedging.

- **Trajectory-level**
    - Correct actions alone do not guarantee task success.
    - Need to check if a sequence of actions (Model's Clarify → User's Response → Model's Final Answer) actually reaches the goal.
    - Trajectory-level evaluation captures true multi-turn success beyond single-turn correctness.

Action-level teaches strategy, but ***trajectory-level*** ensures those strategies lead to successful outcomes.

Method: ACT

# Method Setup

# ACT Phase 1

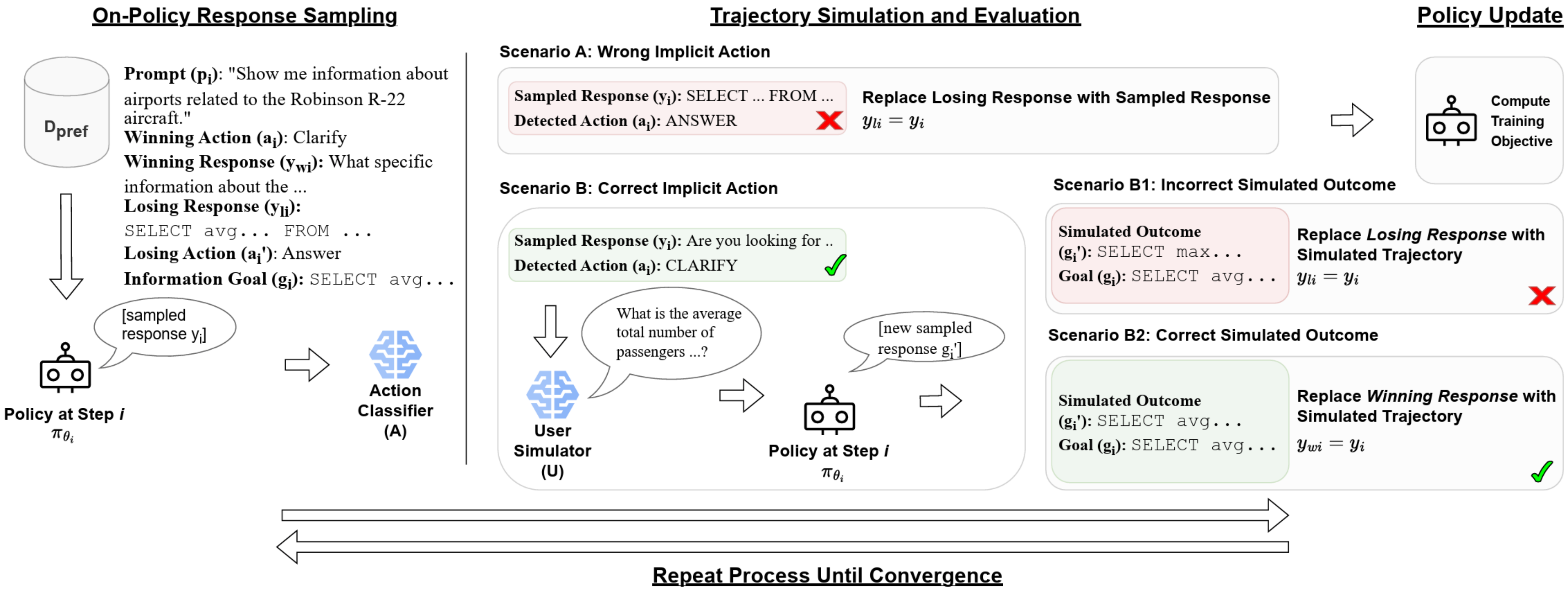
# ACT Phrase 2

Method Setup

- **Method setup with notation**
  - Goal: Train an LLM π_θ as a mixed-initiative agent that decides whether to **Clarify** or **Answer**.
  - Conversation state at turn i:
    - p_i: prompt (context + user query at step i)
    - r_i: reference answer (system-side ground truth)
    - g_i: information goal (final correct outcome)
    - a_i: action chosen from action space $S = \{\text{CLARIFY}, \text{ANSWER}\}$
  - Trajectory: sequence of states $(t_1, t_2, \ldots, t_n)$ ending when the goal g is resolved.
  - System components:
    - M: controllable LLM for response generation; learner
    - A: Action classifier (few-shot LLM)
    - U: User simulator (LLM-based)

Method Setup

- **Method setup with notation**



On-Policy Response Sampling

D_{pref}

Prompt (p_i): "Show me information about airports related to the Robinson R-22 aircraft."
Winning Action (a_i): Clarify
Winning Response (y_{wi}): What specific information about the ...
Losing Response (y_{li}): `SELECT avg... FROM ...`
Losing Action (a_i'): Answer
Information Goal (g_i): `SELECT avg...`

[sampled response y_i]

Policy at Step *i* π_{θ_i}

Action Classifier (A)

Trajectory Simulation and Evaluation

Scenario A: Wrong Implicit Action

Sampled Response (y_i): SELECT ... FROM ...
Detected Action (a_i): ANSWER ✗

Replace Losing Response with Sampled Response
$y_{li} = y_i$

Scenario B: Correct Implicit Action

Sampled Response (y_i): Are you looking for ..
Detected Action (a_i): CLARIFY ✓

What is the average total number of passengers ...?

User Simulator (U)

[new sampled response g_i']

Policy at Step *i* π_{θ_i}

Scenario B1: Incorrect Simulated Outcome

Simulated Outcome (g_i'): `SELECT max...`
Goal (g_i): `SELECT avg...`

Replace *Losing Response* with Simulated Trajectory
$y_{li} = y_i$ ✗

Scenario B2: Correct Simulated Outcome

Simulated Outcome (g_i'): `SELECT avg...`
Goal (g_i): `SELECT avg...`

Replace *Winning Response* with Simulated Trajectory
$y_{wi} = y_i$ ✓

Policy Update

Compute Training Objective

Repeat Process Until Convergence

Phrase 1 on−policy response sampling → ***Phrase 2*** trajectory simulation

ACT: Action-based Contrastive Self-Training

- **Motivation**

1. Standard DPO aligns models with <u>response-level</u> preference pairs (good vs bad answers).
2. But in ambiguous dialogue,
   the crucial factor is the ***action taken*** (Clarify vs Answer), not just correctness of a response.
3. ACT == Extends DPO to <u>action-level</u> and <u>trajectory-level</u> contrastive learning.

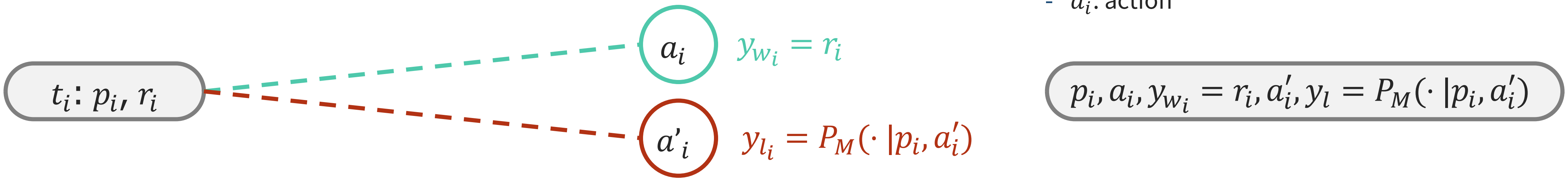
ACT extends DPO by learning from action-level contrast and trajectory-level outcomes, enabling LLMs to decide when to Clarify vs Answer without gold labels.

ACT: Action-based Contrastive Self-Training

• How Preference Pairs Are Maintained in ACT

- p_i: prompt (context + user query at step i)
- r_i: reference answer (system-side ground truth)
- g_i: information goal (final correct outcome)
- a_i: action

[Phrase 1]



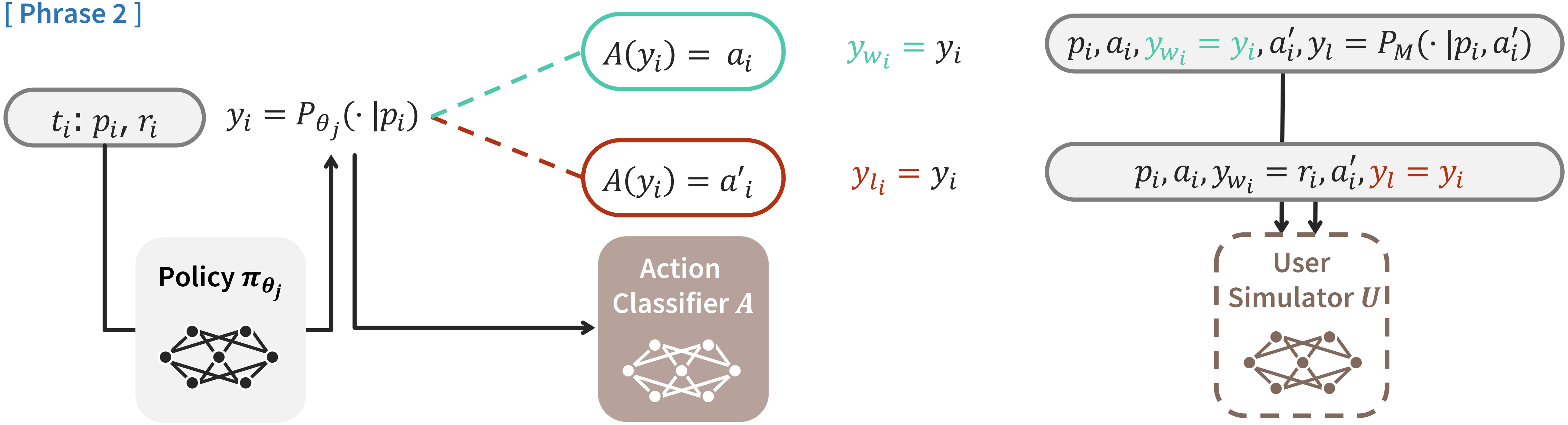
t_i: p_i, r_i

a_i $\quad y_{w_i} = r_i$

a'_i $\quad y_{l_i} = P_M(\cdot\,|p_i, a'_i)$

$p_i, a_i, y_{w_i} = r_i, a'_i, y_l = P_M(\cdot\,|p_i, a'_i)$

[Phrase 2]



t_i: p_i, r_i

$y_i = P_{\theta_j}(\cdot\,|p_i)$

$A(y_i) = a_i$ $\quad y_{w_i} = y_i$

$A(y_i) = a'_i$ $\quad y_{l_i} = y_i$

Policy π_{θ_j}

Action Classifier A

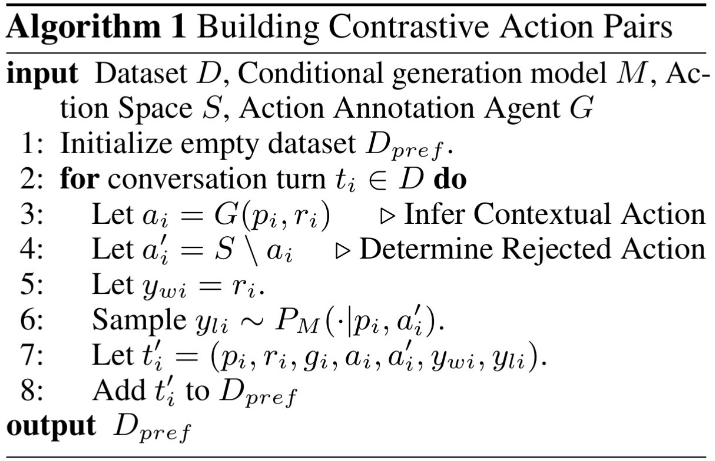
$p_i, a_i, y_{w_i} = y_i, a'_i, y_l = P_M(\cdot\,|p_i, a'_i)$

$p_i, a_i, y_{w_i} = r_i, a'_i, y_l = y_i$

User Simulator U

ACT: Action-based Contrastive Self-Training

- **Phrase 1: Constructing Preference Data**
  - Goal: Collect offline (winning, losing) action-response pairs → build D_{pref}
  - For each turn t_i:
    - Winning action a_i: correct choice
      - e.g., Clarify when ambiguity exists
    - Winning response $y_{w_i} = r_i$ (reference, ground-truth)
    - Rejected action a'_i: opposite choice
    - Losing response y_{l_i}: generated by model M under a'_i



Algorithm 1 Building Contrastive Action Pairs

```
input  Dataset $D$, Conditional generation model $M$, Action Space $S$, Action Annotation Agent $G$
Initialize empty dataset $D_{pref}$.
for conversation turn $t_i \in D$ do
    Let $a_i = G(p_i, r_i)$        ▷ Infer Contextual Action
    Let $a'_i = S \setminus a_i$      ▷ Determine Rejected Action
    Let $y_{wi} = r_i$.
    Sample $y_{li} \sim P_M(\cdot|p_i, a'_i)$.
    Let $t'_i = (p_i, r_i, g_i, a_i, a'_i, y_{wi}, y_{li})$.
    Add $t'_i$ to $D_{pref}$
output  $D_{pref}$
```

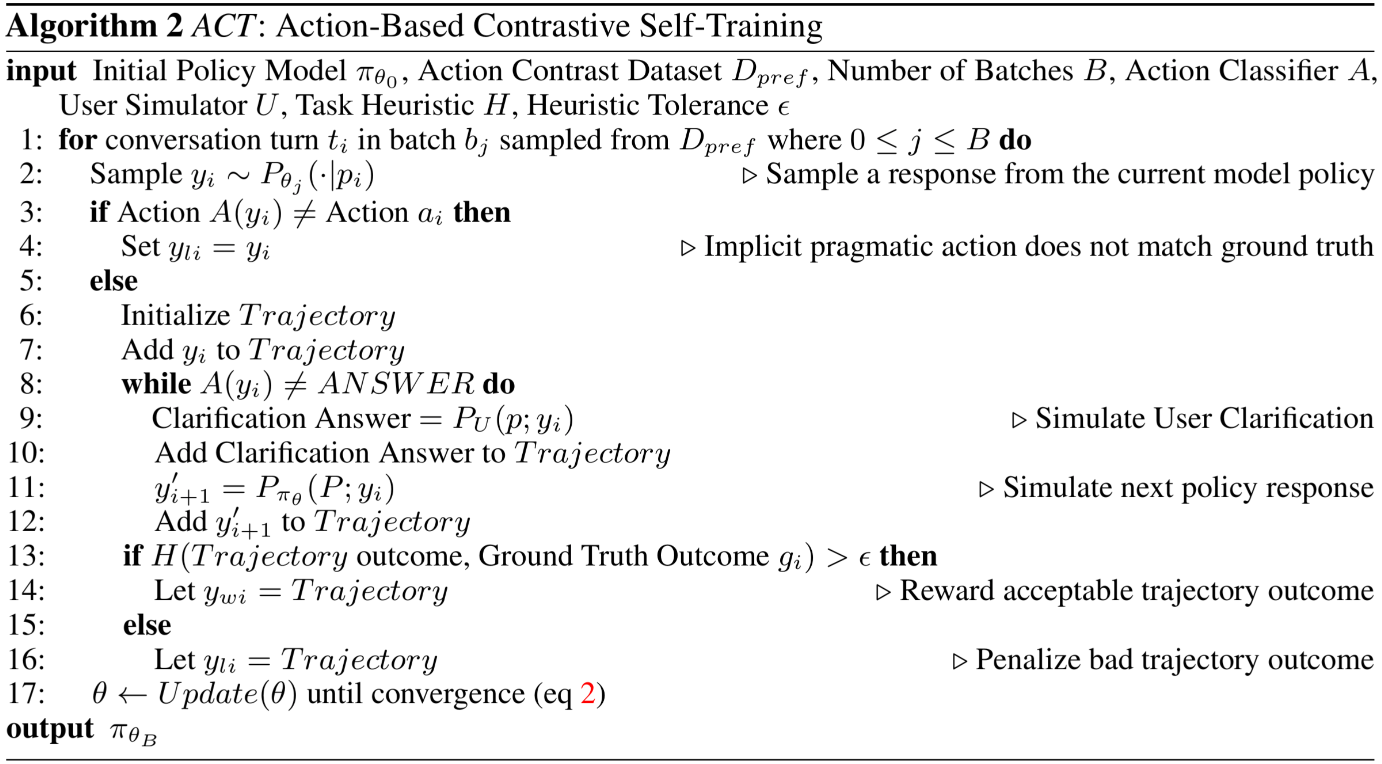
By constructing initial {winning, losing} pairs from references (Phrase #1), provide the contrastive signal needed to start self-training (Phrase #2) .

ACT: Action-based Contrastive Self-Training

- **Phrase 2: Self-training using On-policy Conversation Trajectory Simulation**

- Goal: Perform ***quasi-online self-training*** by updating preference pairs during training.

1. **On-policy Roll-out**: sample response y_i from current policy π_θ
   - Detect action with classifier A.
   - If action is correct, simulate user U reply, continue trajectory until reaching goal.
     - Successful trajectory → winning
     - Failed trajectory → losing
   - If action is wrong, directly losing.
2. **Update Preference Pair**
   - The model continuously accumulates data that reflects its most recent policy.
3. **Training usage**
   - These pairs are directly fed into the DPO loss to update π_θ.



Algorithm 2 *ACT*: Action-Based Contrastive Self-Training

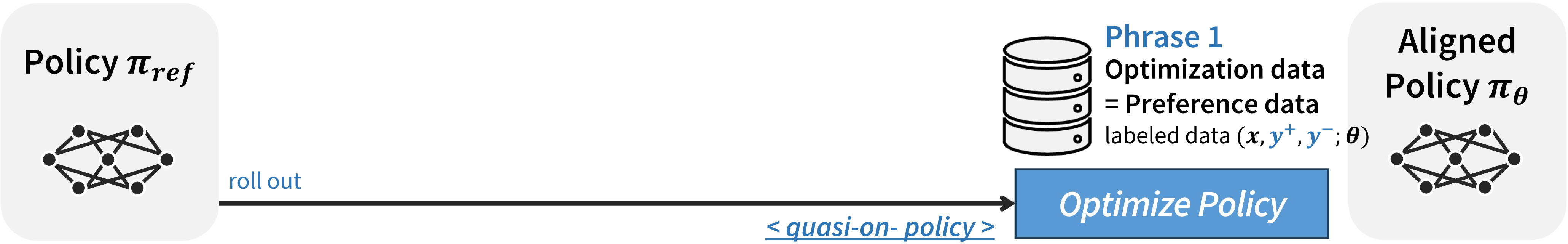
```
input  Initial Policy Model $\pi_{\theta_0}$, Action Contrast Dataset $D_{pref}$, Number of Batches $B$, Action Classifier $A$,
       User Simulator $U$, Task Heuristic $H$, Heuristic Tolerance $\epsilon$
for conversation turn $t_i$ in batch $b_j$ sampled from $D_{pref}$ where $0 \leq j \leq B$ do
    Sample $y_i \sim P_{\theta_j}(\cdot|p_i)$                          ▷ Sample a response from the current model policy
    if Action $A(y_i) \neq$ Action $a_i$ then
        Set $y_{li} = y_i$                          ▷ Implicit pragmatic action does not match ground truth
    else
        Initialize $Trajectory$
        Add $y_i$ to $Trajectory$
        while $A(y_i) \neq ANSWER$ do
            Clarification Answer $= P_U(p; y_i)$                          ▷ Simulate User Clarification
            Add Clarification Answer to $Trajectory$
            $y'_{i+1} = P_{\pi_\theta}(P; y_i)$                          ▷ Simulate next policy response
            Add $y'_{i+1}$ to $Trajectory$
        if $H(Trajectory$ outcome, Ground Truth Outcome $g_i) > \epsilon$ then
            Let $y_{wi} = Trajectory$                          ▷ Reward acceptable trajectory outcome
        else
            Let $y_{li} = Trajectory$                          ▷ Penalize bad trajectory outcome
    $\theta \leftarrow Update(\theta)$ until convergence (eq 2)
output  $\pi_{\theta_B}$
```

ACT: Action-based Contrastive Self-Training

- **Phrase 2: Self-training using On-policy Conversation Trajectory Simulation**
  - Goal: Perform ***quasi-online self-training*** by updating preference pairs <u>during training</u>.
  - DPO Objective

$$\mathcal{L}_{dpo(\pi_\theta;\pi_{ref})} = -\mathbb{E}_{(p,y_w,y_l)\sim\mathcal{D}}\left[\log\sigma(\beta\log\frac{\pi_\theta(y^+|x)}{\pi_{ref}(y^+|x)} - \beta\log\frac{\pi_\theta(y^-|x)}{\pi_{ref}(y^-|x)})\right]$$

  - ACT training
    - Phrase 1: Cold Start



Policy π_{ref}

roll out

< quasi-on- policy >

Phrase 1

Optimization data

= Preference data

labeled data $(\boldsymbol{x}, \boldsymbol{y}^+, \boldsymbol{y}^-; \boldsymbol{\theta})$

Optimize Policy

Aligned Policy π_θ

ACT: Action-based Contrastive Self-Training

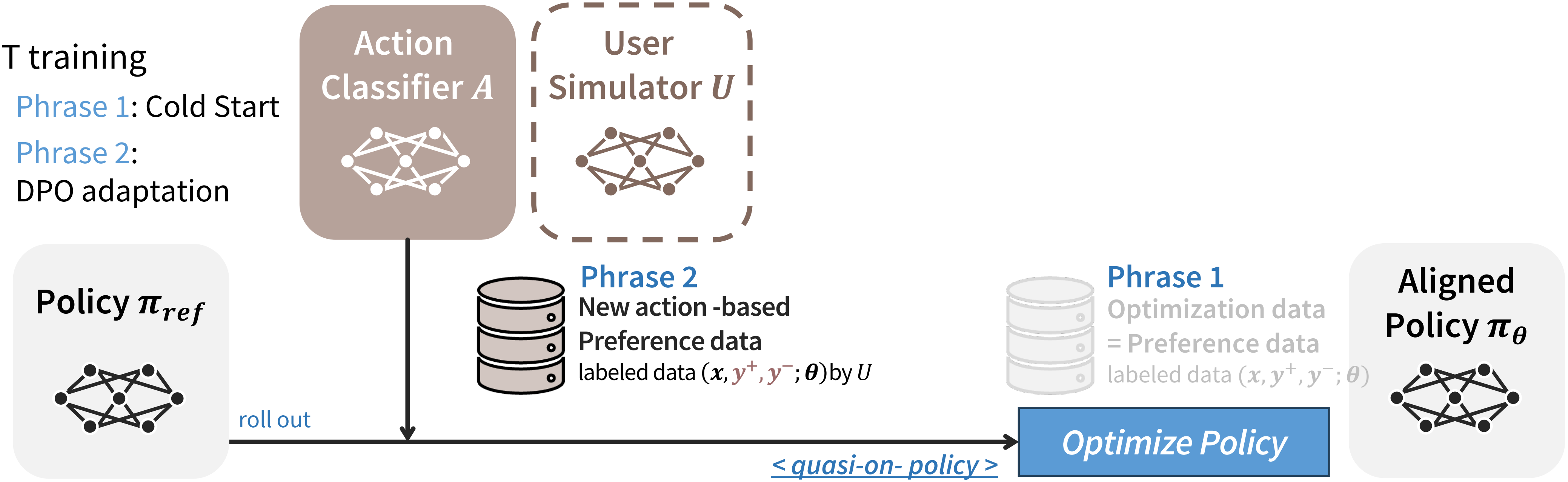
• Phrase 2: Self-training using On-policy Conversation Trajectory Simulation

- Goal: Perform ***quasi-online self-training*** by updating preference pairs <u>during training</u>.

- DPO Objective

$$\mathcal{L}_{dpo(\pi_\theta;\pi_{ref})} = -\mathbb{E}_{(p,y_w,y_l)\sim\mathcal{D}}\left[\log\sigma(\beta\log\frac{\pi_\theta(y^+|x)}{\pi_{ref}(y^+|x)} - \beta\log\frac{\pi_\theta(y^-|x)}{\pi_{ref}(y^-|x)})\right]$$

- ACT training
  - Phrase 1: Cold Start
  - Phrase 2:
    DPO adaptation



Action Classifier A

User Simulator U

Phrase 2

New action -based

Preference data

labeled data $(\boldsymbol{x}, \boldsymbol{y}^+, \boldsymbol{y}^-; \boldsymbol{\theta})$by U

Phrase 1

Optimization data

= Preference data

labeled data $(\boldsymbol{x}, \boldsymbol{y}^+, \boldsymbol{y}^-; \boldsymbol{\theta})$

Policy $\boldsymbol{\pi_{ref}}$

roll out

< quasi-on- policy >

Optimize Policy

Aligned Policy $\boldsymbol{\pi_\theta}$

Evaluation

# Experiment Setup

# Main Results

# Ablation Study

Experiment Setup

- **Dataset**
  - **PACIFIC** (Finance QA, ambiguous subset): Conversational QA for Tabular Data
    - DROP-style evaluation (F1)
  - **Abg-CoQA** (ambiguous CoQA subset for MRC): Conversational QA for Machine Reading Comprehension
    - Semantic similarity metric
  - **AmbigSQL** (new dataset, derived from Spider): Ambiguous Conversational Text-to-SQL Generation
    - 3 ambiguity types: Information / Population / Presentation
    - SQL execution accuracy metric

Type	Definition	Example		SQL Query-level
Information	Attribute or information needed is unclear	Show students with high grades.	Which subject? What threshold?	`SELECT, WHERE`
Population	Target group or table is ambiguous	How many users are there?	All users? Users of which service?	`FROM`
Presentation	Expression/condition is underspecified	List large cities.	Large by population or by area?	`ORDER/GROUP BY, HAVING`

Experiments use three ambiguous datasets: PACIFIC, Abg-CoQA, AmbigSQL

Experiment Setup

- **Baselines**
  - SFT: supervised fine-tuned policy
  - DPO: standard Direct Preference Optimization
  - ACT: proposed method (Phase 1 + Phase 2)
  - zero-label ACT:
    pseudo-labels instead of gold action labels

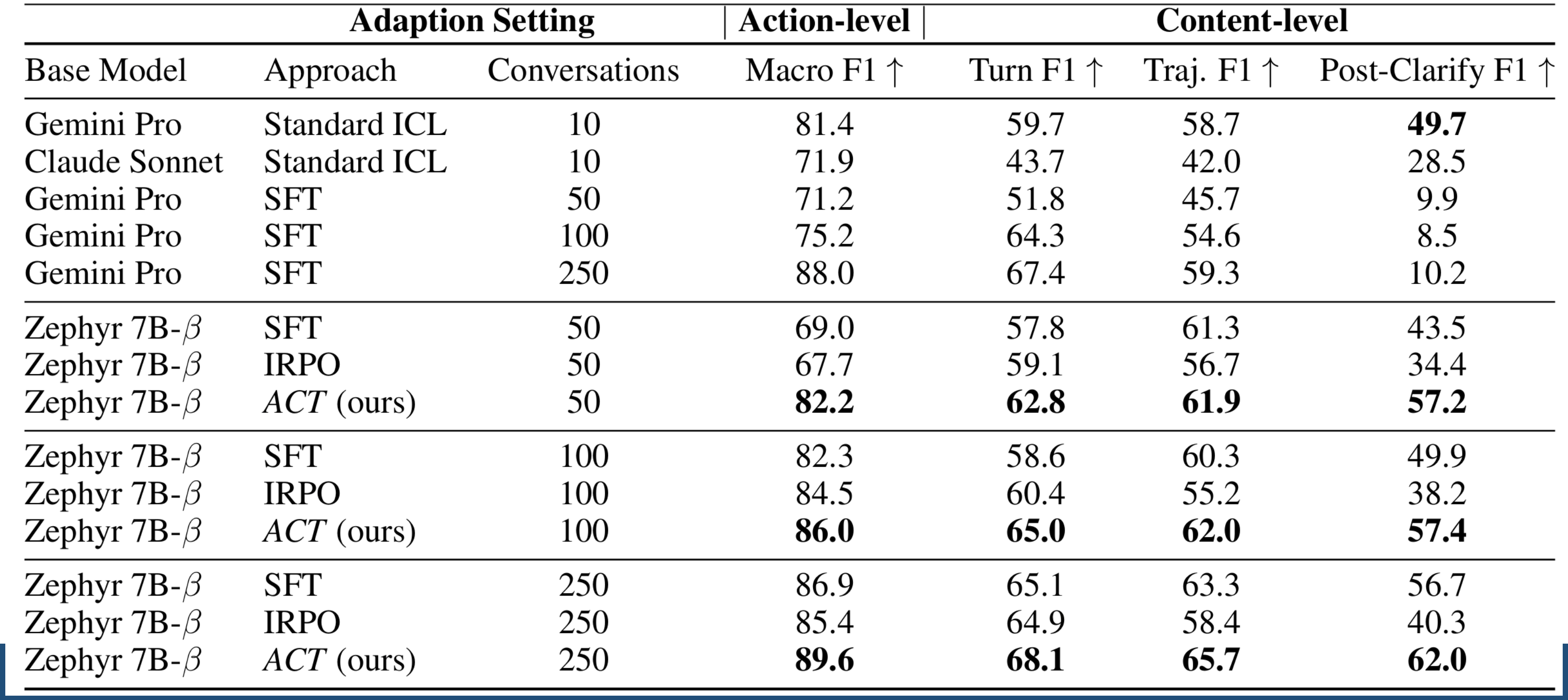
- **Evaluation Metrics**
  - PACIFIC; ***DROP F1*** (span-based QA correctness)
  - Abg-CoQA; ***Semantic similarity*** with references
  - AmbigSQL; ***Execution accuracy***
    (did the SQL run correctly?)
  - ***Trajectory-level*** focus: measure success after Clarify → Answer, not just single response

- **Models**
  - Backbone: LLaMA-2-7B (main)
    - Some smaller experiments with 13B variant for ablations
  - All experiments run on A100 GPUs

- **Training Details**
  - Optimizer: AdamW
  - Learning rate: 2×10^{-5}
  - Batch size: 64
  - Max sequence length: 1024 tokens
  - Epochs: ~3–5
    (same across SFT/DPO/ACT for fair comparison)

Main Results

- **PACIFIC**
  - **SFT**: baseline level; **DPO**: slight improvement, but limited in handling ambiguity.
  - **ACT**: achieves the highest F1, strong **data efficiency** (maintains performance even w/ smaller training sizes).

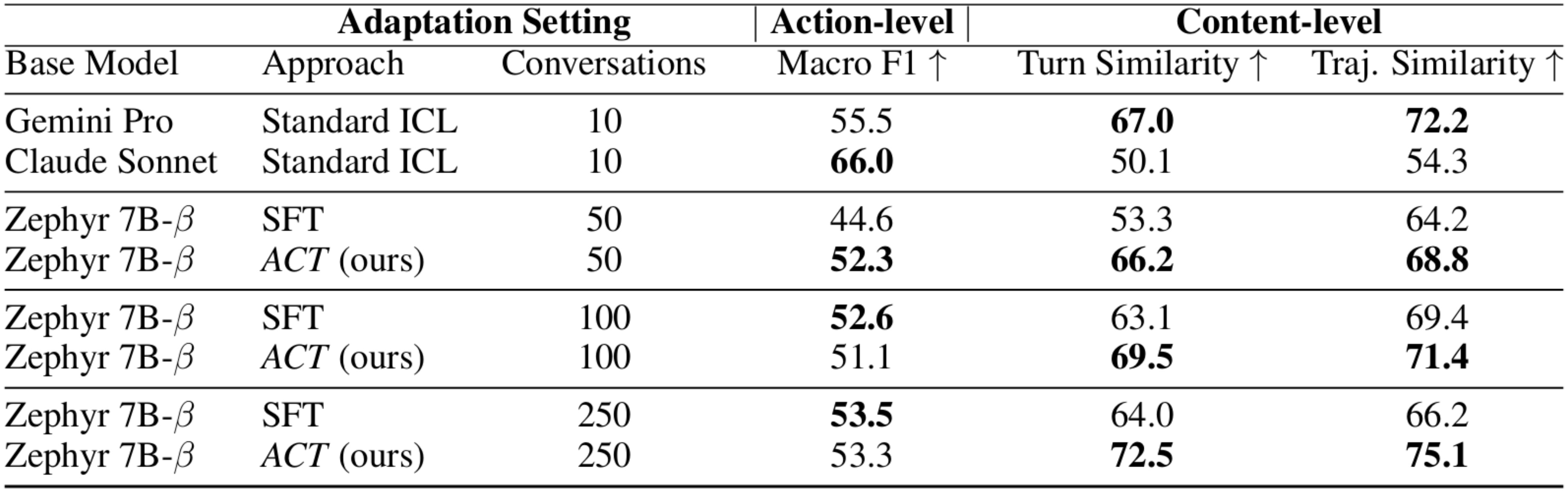


Adaption Setting			Action-level	Content-level		
Base Model	Approach	Conversations	Macro F1 ↑	Turn F1 ↑	Traj. F1 ↑	Post-Clarify F1 ↑
Gemini Pro	Standard ICL	10	81.4	59.7	58.7	**49.7**
Claude Sonnet	Standard ICL	10	71.9	43.7	42.0	28.5
Gemini Pro	SFT	50	71.2	51.8	45.7	9.9
Gemini Pro	SFT	100	75.2	64.3	54.6	8.5
Gemini Pro	SFT	250	88.0	67.4	59.3	10.2
Zephyr 7B-β	SFT	50	69.0	57.8	61.3	43.5
Zephyr 7B-β	IRPO	50	67.7	59.1	56.7	34.4
Zephyr 7B-β	*ACT* (ours)	50	**82.2**	**62.8**	**61.9**	**57.2**
Zephyr 7B-β	SFT	100	82.3	58.6	60.3	49.9
Zephyr 7B-β	IRPO	100	84.5	60.4	55.2	38.2
Zephyr 7B-β	*ACT* (ours)	100	**86.0**	**65.0**	**62.0**	**57.4**
Zephyr 7B-β	SFT	250	86.9	65.1	63.3	56.7
Zephyr 7B-β	IRPO	250	85.4	64.9	58.4	40.3
Zephyr 7B-β	*ACT* (ours)	250	**89.6**	**68.1**	**65.7**	**62.0**

By effectively choosing the ***Clarify*** action, ACT improves final Answer accuracy and strengthens robustness in ambiguous situations.

Main Results

- **Abg-CoQA**
  - **SFT/DPO**: Single-turn answers look plausible, but they fall short in **multi-turn coherence**.
  - **ACT**: Achieves the **highest** trajectory-level similarity to references.

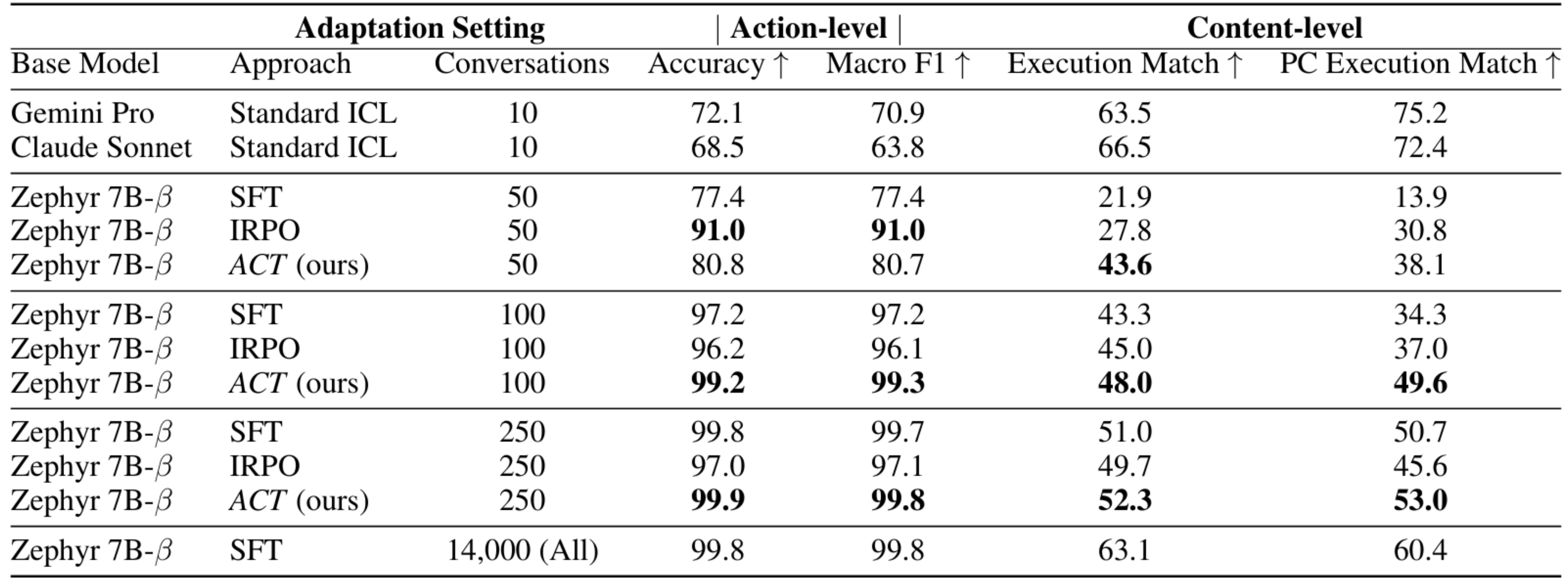


Adaptation Setting			Action-level	Content-level	
Base Model	Approach	Conversations	Macro F1 ↑	Turn Similarity ↑	Traj. Similarity ↑
Gemini Pro	Standard ICL	10	55.5	**67.0**	**72.2**
Claude Sonnet	Standard ICL	10	**66.0**	50.1	54.3
Zephyr 7B-β	SFT	50	44.6	53.3	64.2
Zephyr 7B-β	*ACT* (ours)	50	**52.3**	**66.2**	**68.8**
Zephyr 7B-β	SFT	100	**52.6**	63.1	69.4
Zephyr 7B-β	*ACT* (ours)	100	51.1	**69.5**	**71.4**
Zephyr 7B-β	SFT	250	**53.5**	64.0	66.2
Zephyr 7B-β	*ACT* (ours)	250	53.3	**72.5**	**75.1**

The flow Clarify 〉〉 User simulator 〉〉 Answer aligns more closely with the reference answers.

Main Results

- **AmbigSQL**
  - **ACT**: clear superiority in post-clarification accuracy compared to SFT/DPO.
  - Improvements are observed across all three ambiguity types (Information, Population, Presentation).

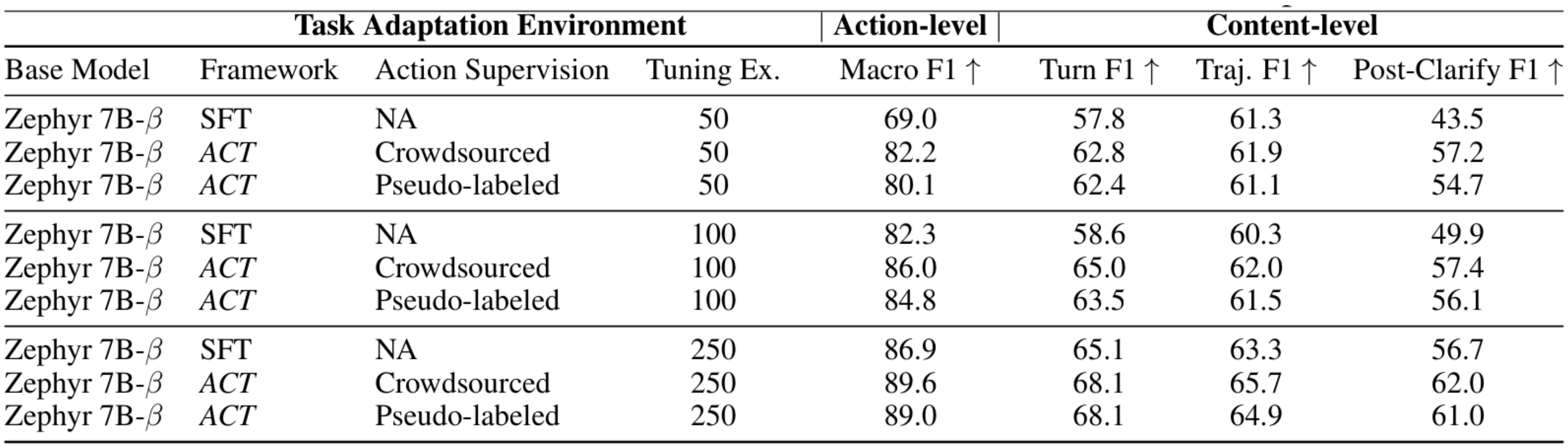


Adaptation Setting			| Action-level |		Content-level	
Base Model	Approach	Conversations	Accuracy ↑	Macro F1 ↑	Execution Match ↑	PC Execution Match ↑
Gemini Pro	Standard ICL	10	72.1	70.9	63.5	75.2
Claude Sonnet	Standard ICL	10	68.5	63.8	66.5	72.4
Zephyr 7B-β	SFT	50	77.4	77.4	21.9	13.9
Zephyr 7B-β	IRPO	50	**91.0**	**91.0**	27.8	30.8
Zephyr 7B-β	*ACT* (ours)	50	80.8	80.7	**43.6**	38.1
Zephyr 7B-β	SFT	100	97.2	97.2	43.3	34.3
Zephyr 7B-β	IRPO	100	96.2	96.1	45.0	37.0
Zephyr 7B-β	*ACT* (ours)	100	**99.2**	**99.3**	**48.0**	**49.6**
Zephyr 7B-β	SFT	250	99.8	99.7	51.0	50.7
Zephyr 7B-β	IRPO	250	97.0	97.1	49.7	45.6
Zephyr 7B-β	*ACT* (ours)	250	**99.9**	**99.8**	**52.3**	**53.0**
Zephyr 7B-β	SFT	14,000 (All)	99.8	99.8	63.1	60.4

Not just about single-answer accuracy—Clarify → final SQL execution success increases, proving the importance of trajectory-level evaluation.

Main Results

- **ACT In-the-wild: Learning without Dialogue Action Supervision**
  - Backbone: Gemini 2.5 Pro
  - Uses only classifier pseudo-labels (*no gold action labels*) achieves performance nearly identical to gold-labeled ACT.



Task Adaptation Environment				**Action-level**	**Content-level**		
Base Model	Framework	Action Supervision	Tuning Ex.	Macro F1 ↑	Turn F1 ↑	Traj. F1 ↑	Post-Clarify F1 ↑
Zephyr 7B-β	SFT	NA	50	69.0	57.8	61.3	43.5
Zephyr 7B-β	*ACT*	Crowdsourced	50	82.2	62.8	61.9	57.2
Zephyr 7B-β	*ACT*	Pseudo-labeled	50	80.1	62.4	61.1	54.7
Zephyr 7B-β	SFT	NA	100	82.3	58.6	60.3	49.9
Zephyr 7B-β	*ACT*	Crowdsourced	100	86.0	65.0	62.0	57.4
Zephyr 7B-β	*ACT*	Pseudo-labeled	100	84.8	63.5	61.5	56.1
Zephyr 7B-β	SFT	NA	250	86.9	65.1	63.3	56.7
Zephyr 7B-β	*ACT*	Crowdsourced	250	89.6	68.1	65.7	62.0
Zephyr 7B-β	*ACT*	Pseudo-labeled	250	89.0	68.1	64.9	61.0

Low dependence on labels and remains robust through self−training

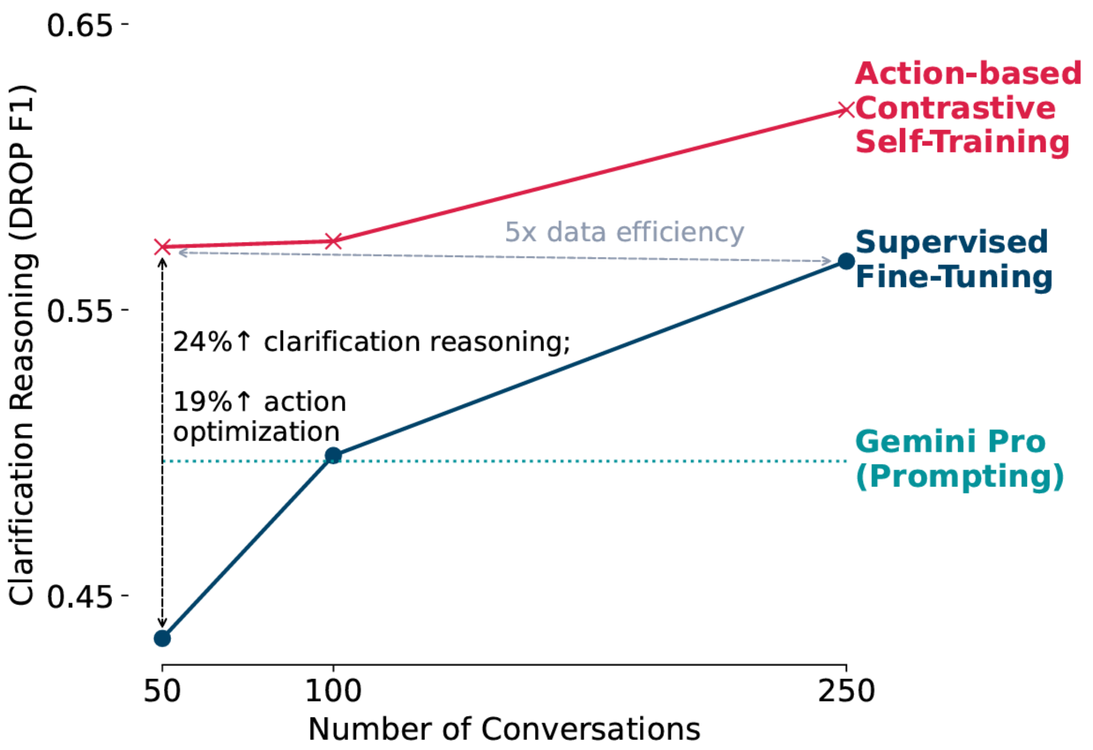
Evaluation

Main Results

- **Data Efficiency**

  - **ACT**: High performance even w/ small data, and consistently best as data size grows.

  - **SFT**: performance increases gradually w/ more data, but remains low in the small-data regime.

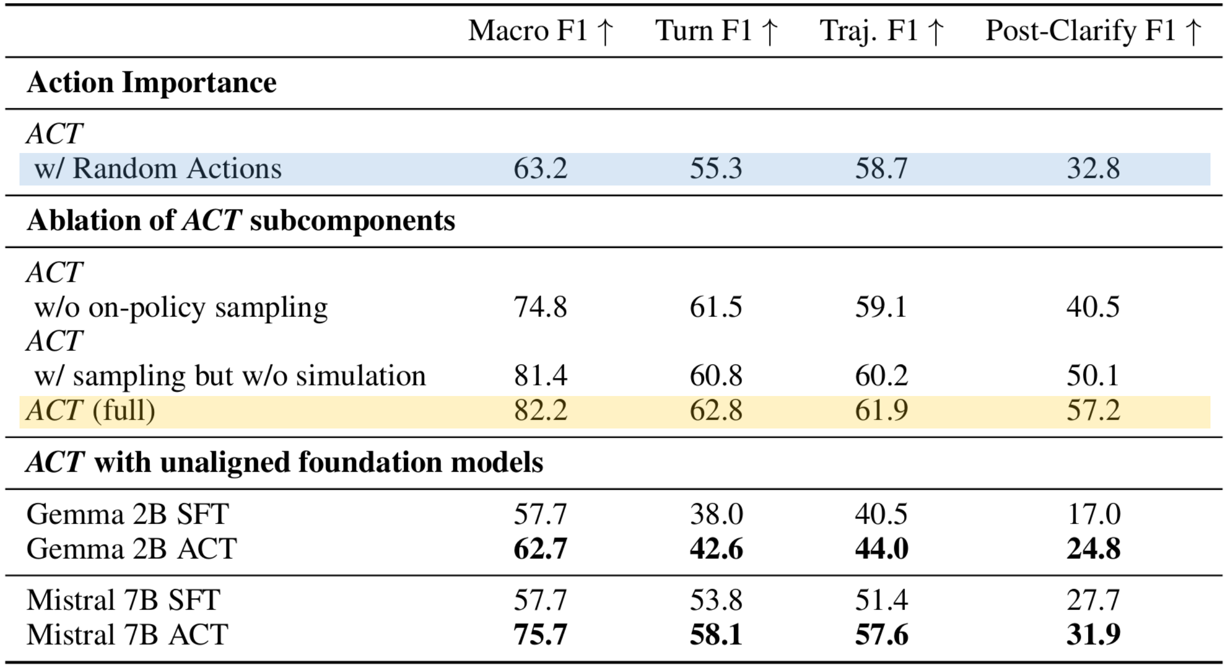


ACT achieves higher F1 scores than SFT, demonstrating strong data efficiency.

Ablation Study

- **Ablations**

  - **Without action contrast**:
    performance drops significantly
    → contrasting Clarify vs Answer is essential.



	Macro F1 ↑	Turn F1 ↑	Traj. F1 ↑	Post-Clarify F1 ↑
Action Importance				
ACT w/ Random Actions	63.2	55.3	58.7	32.8
Ablation of *ACT* subcomponents				
ACT w/o on-policy sampling	74.8	61.5	59.1	40.5
ACT w/ sampling but w/o simulation	81.4	60.8	60.2	50.1
ACT (full)	82.2	62.8	61.9	57.2
***ACT* with unaligned foundation models**				
Gemma 2B SFT	57.7	38.0	40.5	17.0
Gemma 2B ACT	**62.7**	**42.6**	**44.0**	**24.8**
Mistral 7B SFT	57.7	53.8	51.4	27.7
Mistral 7B ACT	**75.7**	**58.1**	**57.6**	**31.9**

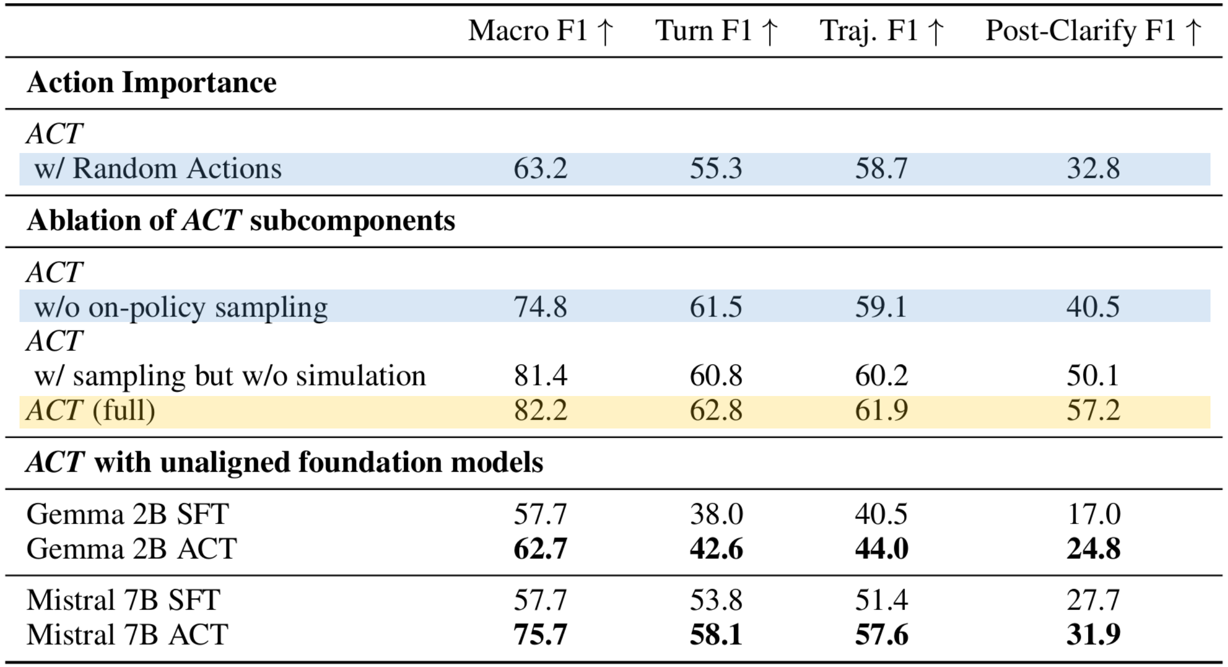
All 3 components — action contrast, trajectory simulation, and on−policy rollouts — are essential for ACT's effectiveness.

Ablation Study

- **Ablations**

  - **Without action contrast**:
    performance drops significantly
    → contrasting Clarify vs Answer is essential.

  - **Without on-policy rollouts**:
    improvement is restricted
    → quasi-online updates are crucial.



	Macro F1 ↑	Turn F1 ↑	Traj. F1 ↑	Post-Clarify F1 ↑
Action Importance				
ACT w/ Random Actions	63.2	55.3	58.7	32.8
Ablation of *ACT* subcomponents				
ACT w/o on-policy sampling	74.8	61.5	59.1	40.5
ACT w/ sampling but w/o simulation	81.4	60.8	60.2	50.1
ACT (full)	82.2	62.8	61.9	57.2
***ACT* with unaligned foundation models**				
Gemma 2B SFT	57.7	38.0	40.5	17.0
Gemma 2B ACT	**62.7**	**42.6**	**44.0**	**24.8**
Mistral 7B SFT	57.7	53.8	51.4	27.7
Mistral 7B ACT	**75.7**	**58.1**	**57.6**	**31.9**

All 3 components — action contrast, trajectory simulation, and on−policy rollouts — are essential for ACT's effectiveness.

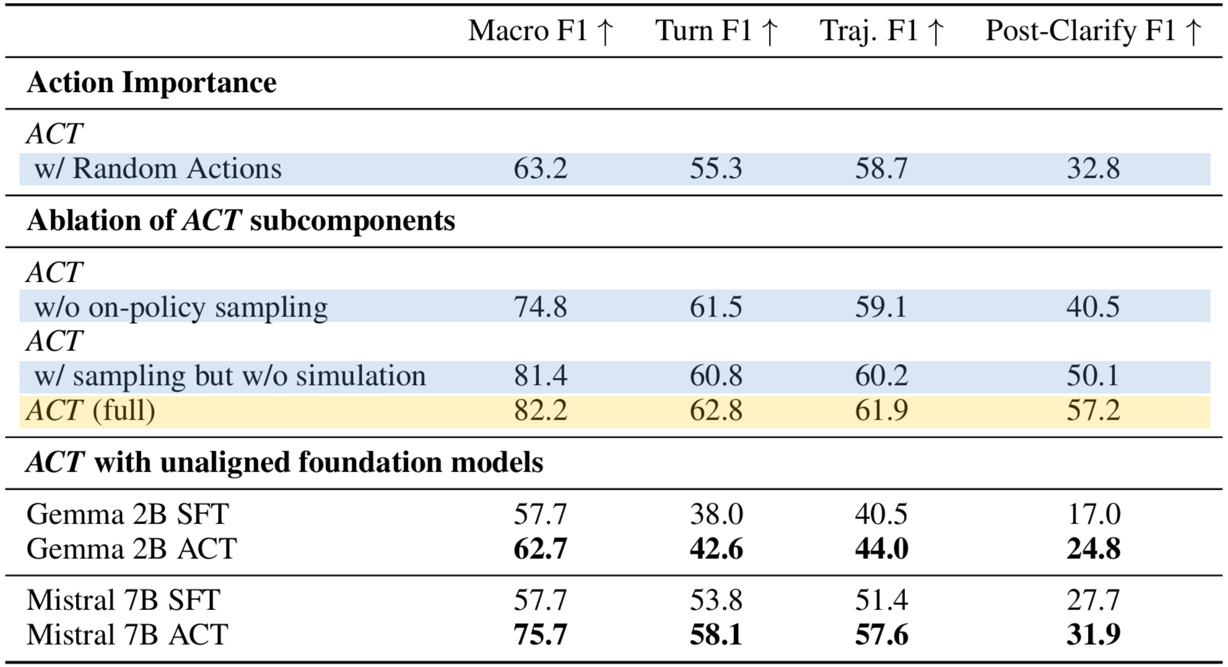
Ablation Study

- **Ablations**

  - **Without action contrast**:
    performance drops significantly
    → contrasting Clarify vs Answer is essential.

  - **Without on-policy rollouts**:
    improvement is restricted
    → quasi-online updates are crucial.

  - **Without trajectory simulation**:
    performance is limited
    when only single responses are considered
    → multi-turn outcomes matter.



	Macro F1 ↑	Turn F1 ↑	Traj. F1 ↑	Post-Clarify F1 ↑
Action Importance				
ACT w/ Random Actions	63.2	55.3	58.7	32.8
Ablation of *ACT* subcomponents				
ACT w/o on-policy sampling	74.8	61.5	59.1	40.5
ACT w/ sampling but w/o simulation	81.4	60.8	60.2	50.1
ACT (full)	82.2	62.8	61.9	57.2
***ACT* with unaligned foundation models**				
Gemma 2B SFT	57.7	38.0	40.5	17.0
Gemma 2B ACT	**62.7**	**42.6**	**44.0**	**24.8**
Mistral 7B SFT	57.7	53.8	51.4	27.7
Mistral 7B ACT	**75.7**	**58.1**	**57.6**	**31.9**

All 3 components — action contrast, trajectory simulation, and on−policy rollouts — are essential for ACT's effectiveness.

Conclusion

Conclusion

- **Conclusion**
  - ACT extends alignment from *response-level* → ***action-level*** → ***trajectory-level***.
  - Consistently improves performance on ambiguous QA, MRC, and Text-to-SQL.
  - Demonstrates data efficiency and works even without gold action labels (pseudo-label ACT).
- **Discussion**
  - ***Clarifying*** questions are a key behavior for mixed-initiative agents.
  - ACT shows that preference learning can be generalized beyond final answers to action choices.
- **Limitations**
  - Simplified action space: only {CLARIFY, ANSWER}; real-world systems need richer actions.
  - ***Simulator reliance***: user simulator may not reflect real human responses → limits ecological validity.
  - Dataset coverage: not all types of real dialogue ambiguity.

Aligning LLMs at the action and trajectory level is both feasible and beneficial,
but richer action spaces and real user feedback are key for future progress.

Thank You

Yejin Yoon

HYU NLP Lab.
Hanyang University, South Korea

stillwithyou@hanyang.ac.kr