A DATASETS

WebShop. WebShop (Yao et al., 2022) is a goal-oriented web navigation environment where an agent interacts with an e-commerce platform to fulfill natural language shopping instructions. A final reward in $[0, 1]$ is assigned based on how well the purchased item matches the instruction in terms of attributes and price. For training, we use the expert and GPT-4 ReAct-style trajectories provided by Song et al. (2024), retaining only those with final rewards above 0.7 to ensure quality.

Example WebShop Trajectory

Instruction:
I need a long clip-in hair extension which is natural looking, and price lower than 40.00 dollars.

Conversation:

- **Human:** You are web shopping. I will give you instructions about what to do. Every round I will give you an observation and a list of available actions...
- **GPT:** OK
- **Human:** WebShop [SEP] Instruction: [SEP] i need a long clip-in hair extension ...
- **GPT:** *Thought: I need to find a long clip-in hair extension ...* **Action:** `search[long clip-in hair extension]`
- **Human:** Observation: ... (product list with IDs and prices)
- **GPT:** *Thought: The first option B09C337K8S fits the budget and is natural looking...* **Action:** `click[b09c337k8s]`
- **Human:** Observation: ... (product details shown)
- **GPT:** *Thought: The details confirm it is natural looking and affordable.* **Action:** `click[buy now]`

Reward: 1.0

ScienceWorld. ScienceWorld (Wang et al., 2022) is a text-based embodied environment for conducting elementary science experiments across ten domains, including thermodynamics and electricity. Each task consists of multiple optional subgoals, and the final reward in $[0, 1]$ reflects the proportion of subgoals completed. Following Song et al. (2024), we exclude the two longest task types and subsample test instances with more than ten variations. We evaluate generalization using the original test set of unseen variations (e.g., training on boiling water and testing on boiling lead) and use the development set for seen scenarios. Expert trajectories are drawn from the official dataset, with additional GPT-4–generated trajectories filtered by final rewards, as in Song et al. (2024).

Example ScienceWorld Trajectory

Instruction:
Your task is to determine if a metal pot is electrically conductive. The metal pot is located in the kitchen. If conductive, place it in the blue box; if nonconductive, place it in the orange box.

Conversation:

- **Human:** You are a helpful assistant to do some scientific experiment in an environment. (Lists available rooms and actions.)
- **GPT:** OK
- **Human:** Your task is to determine if metal pot is electrically conductive...

- **GPT:** *Thought: I should locate the pot, probably in the kitchen.* **Action:** `look around`
- **Human:** Observation: You are in the foundry...
- **GPT:** *Thought: Wrong room, I need to go to the kitchen.* **Action:** `teleport to kitchen`
- **Human:** Observation: You teleport to the kitchen.
- **GPT:** *Thought: I should pick up the metal pot.* **Action:** `pick up metal pot`
- **Human:** Observation: Pot moved to inventory.
- **GPT:** *Thought: Now focus on the pot to test it.* **Action:** `focus on metal pot`
- **Human:** Observation: You focus on the pot.
- **GPT:** *Thought: I need to set up a circuit in the workshop.* **Action:** `teleport to workshop`
- **Human:** Observation: You teleport to the workshop (battery, wires, bulbs available).
- **GPT:** *Thought: Place the pot and build the circuit.* **Action:** `drop metal pot`
- **Human:** Observation: Pot placed in workshop.
- **GPT:** *Thought: Connect wires, battery, and pot in circuit.* **Action:** `connect battery anode to orange wire ...` (multiple connections omitted)
- **Human:** Observation: Circuit components connected.
- **GPT:** *Thought: Wait and check if the bulb lights up.* **Action:** `wait1`
- **Human:** Observation: Blue light bulb is on.
- **GPT:** *Thought: The pot is conductive. Place it in the blue box.* **Action:** `move metal pot to blue box`

InterCodeSQL. InterCodeSQL (Yang et al., 2023) is an interactive SQL querying environment where an agent must translate natural language questions into executable SQL programs through multi-turn interaction. The environment provides a binary reward, which we normalize to $[0, 1]$, indicating whether the final query produces the correct result. We use the expert trajectories collected by Xiong et al. (2024) with GPT-4, retaining only those with reward 1.0. Since the original dataset does not contain intermediate reasoning, each trajectory is augmented with ReAct-style (Yao et al., 2023) thought annotations generated by GPT-4, as in Xiong et al. (2024).

Example InterCodeSQL Trajectory

Instruction:
What are the enrollment and primary conference for the university which was founded the earliest?

Conversation:

- **Human:** You are a helpful assistant assigned with the task of problem-solving. To achieve this, you will interact with a MySQL Database system using SQL queries to answer a question. (Gives detailed formatting rules.)
- **GPT:** OK
- **Human:** What are the enrollment and primary conference for the university which was founded the earliest?
- **GPT:** *Thought: I need to check what tables are available.* **Action:** `SHOW TABLES`
- **Human:** Observation: `[('basketball_match',), ('university',)]`
- **GPT:** *Thought: The `university` table should have the info. I should check its schema.* **Action:** `DESC university`

- **Human:** Observation: `[('School', ...), ('Founded', ...), ('Enrollment', ...), ('Primary_conference', ...)]`
- **GPT:** *Thought: The table has Founded, Enrollment, Primary_conference. I can query the earliest Founded.* **Action:** `SELECT Enrollment, Primary_conference FROM university ORDER BY Founded ASC LIMIT 1`
- **Human:** Observation: `[(19067.0, 'Colonial Athletic Association (D-I)')]`
- **GPT:** *Thought: The earliest university has enrollment 19067 and primary conference Colonial Athletic Association (D-I). This is the final answer.* **Action:** `submit`

B Implementation Details

We adopt Llama-2-7B-Chat (Touvron et al., 2023) as the primary base model for constructing LLM agents, following prior work (Song et al., 2024). All models are optimized with AdamW (Loshchilov & Hutter, 2017), and we use the same hyperparameter settings across datasets for fair comparison. During the supervised fine-tuning (SFT) stage, we train for 3 epochs with a batch size of 48 and a learning rate of 1e-5, using 3% linear warmup followed by a cosine decay schedule. The base agent then explores each training instance once to collect failure trajectories. In the subsequent optimization phase, we train with a batch size of 32 and a learning rate of 1e-6 to 5e-7, with the DPO scaling factor β set to 0.1-0.5. The number of optimization epochs is fixed to 3, and the number of co-evolution iterations is set to 3 for WebShop and ScienceWorld and 5 for InterCodeSQL. All experiments are conducted on 8 NVIDIA H100 GPUs with 80GB.

C Quality of the hard negatives

C.1 WebShop

HDMI Cables under $50

Instruction: I'm looking for ten high-speed, gold-plated HDMI cables, with price lower than $50.00.

ETO: The agent selects a single ProHT 6' HDMI cable priced at $100.00, ignoring both the budget and the required quantity of ten. It proceeds to purchase without checking alternatives or verifying high-speed and gold-plated specifications.
Reward: 0.50 *Steps:* 4 *Outcome:* Failure

Ours: The agent searches specifically for multi-pack high-speed, gold-plated HDMI cables under the budget. It inspects the QualGear 10 ft HDMI 2.0 cable, verifies length, certification, and price, and selects a variant satisfying all constraints except the exact pack quantity.
Reward: 0.75 *Steps:* 5 *Outcome:* Failure

Hard Negative Justification: The trajectory performs structured filtering over pack size, cable type, certification, and budget. It misses only the strict ten-cable requirement, forming a near-success failure ideal for hard-negative training.

Solid Wood Storage Bench in Grey

Instruction: I want a solid wood bench with storage space for my living room, grey in color, and under $210.00.

ETO: The agent selects a grey accent bench after minimal inspection, without verifying solid-wood construction or cross-checking storage features, and purchases it without considering additional candidates or validating the price constraints.

Reward: 0.50 *Steps:* 3 *Outcome:* Failure

Ours: The agent explores multiple pages, filters benches by wood construction, storage capacity, and color, and selects a rustic grey storage bench that aligns with the material and functional requirements, reasoning about a small price deviation.
Reward: 0.75 *Steps:* 5 *Outcome:* Failure

Hard Negative Justification: The trajectory validates material, storage design, and color through multi-step attribute checks. It forms a structurally correct solution that narrowly misses the budget criterion, yielding a high-quality hard negative.

Machine-Washable Curtains (52”×90”)

Instruction: I need a machine-washable curtain for the living room, sized 52” wide by 90” long, priced under $60.00.

ETO: The agent clicks an early search result, selects the 52”×90” option, and buys it without verifying washability, comparing alternatives, or checking that the final price meets the budget.
Reward: 0.50 *Steps:* 4 *Outcome:* Failure

Ours: The agent navigates through multiple product pages, filtering by washability, size, and price. It identifies a curtain with a 52”×90” option, verifies that it is machine-washable and within budget, and chooses the matching size variant before purchasing.
Reward: 0.75 *Steps:* 8 *Outcome:* Failure

Hard Negative Justification: The trajectory conducts systematic elimination of mismatching candidates, checks all constraints, and produces an almost correct selection. Its structured decision process provides a prototypical hard-negative example.

C.2 SCIENCEWORLD

Moving a Non-Living Object to the Green Box

Instruction: Find a non-living object, focus on it, and move it to the green box in the workshop.

ETO: The agent teleports to the workshop, selects the yellow wire as the non-living object, and moves it into the green box. However, it fails to perform the required focus step and drifts into repeated `wait1` and `look around` actions, stalling without further task-aligned behavior.
Reward: 0.25 *Steps:* 15 *Outcome:* Failure

Ours: The agent selects the same yellow wire, places it into the green box, and then issues explicit focus actions on both the box and the wire inside it. It continues checking the environment and navigating purposefully, maintaining a coherent interpretation of the task even though the environment does not register success.
Reward: 0.75 *Steps:* 15 *Outcome:* Failure

Hard Negative Justification: The trajectory follows the full instruction—object selection, movement, and focused inspection—and only misses the success flag due to environment-level evaluation. It represents a near-solution failure and serves as an ideal hard negative.

Turning On a Green Light Bulb with Renewable Power

Instruction: Your task is to turn on the green light bulb. First, focus on the green light bulb. Then, create an electrical circuit that powers it on. Prefer renewable power sources when possible.

ETO: The agent explores the environment and interacts with various components such as wires, switches, and power sources. It partially assembles a circuit but alternates between focusing on unrelated objects and performing ineffective actions, leaving the circuit incomplete and the green bulb off by the end of the episode.
Reward: 0.43 *Steps:* 30 *Outcome:* Failure

Ours: The agent identifies the green light bulb early and issues repeated focus actions on it and on nearby circuit elements. It constructs a more coherent circuit by systematically connecting wires between the bulb and a renewable power component, checks the bulb's state multiple times, and maintains task-aligned reasoning, but still fails to trigger the environment's success condition.
Reward: 0.58 *Steps:* 30 *Outcome:* Failure

Hard Negative Justification: The trajectory follows the full instruction, including focusing on the target bulb and assembling a near-correct renewable circuit, and fails only due to subtle environment-level completion criteria.

Growing a Banana from Seed to Fruit

Instruction: Your task is to grow a banana. This requires obtaining banana seeds, planting them in soil, providing water and light, and waiting until the banana grows.

ETO: The agent collects several relevant objects such as seeds and containers but struggles with interaction ordering and location choice. It issues redundant navigation and inspection commands and fails to complete a coherent cycle of planting, watering, and waiting in a suitable environment, leaving the plant underdeveloped.
Reward: 0.36 *Steps:* 55 *Outcome:* Failure

Ours: The agent explicitly gathers banana seeds, moves them to appropriate soil or planter objects, and performs a structured sequence of planting, watering, and exposing the plant to light. It repeatedly checks the growth state and adjusts its actions, closely following the intended multi-step procedure even though the environment does not register task completion.
Reward: 0.50 *Steps:* 60 *Outcome:* Failure

Hard Negative Justification: The trajectory executes all key sub-tasks of seed collection, planting, watering, and monitoring, making it a faithful but slightly incomplete realization of the target behavior.

C.3 INTERCODESQL

Films Not Presented in China

Instruction: List the titles and directors of films that were never presented in China.

ETO: The agent inspects several tables but repeatedly issues queries referencing nonexistent columns (e.g., `Market`, `country`), incorrect table names (e.g., `film_market_estimation`), and invalid join paths. It ultimately fails to form any executable SQL command.
Reward: 0.00 *Steps:* 6 *Outcome:* Failure

Ours: The agent checks table schemas, identifies usable fields, and iteratively searches for the appropriate join through `market` after rejecting invalid table/column combinations. It

eventually constructs a syntactically valid SQL query that returns a set of film titles and directors.
Reward: 0.77 *Steps:* 9 *Outcome:* Failure

Hard Negative Justification: The trajectory demonstrates structured schema inspection and multi-step join reasoning. It forms an executable SQL query aligned with the task, making it a near-solution hard negative.

Reviewers Who Rated Above 3 Stars

Instruction: Find the names of reviewers who previously rated a movie more than 3 stars.

ETO: The agent misinterprets table schemas, issuing invalid joins between `reviewer`, `rating`, and `movie`. It repeatedly rechecks the same tables and produces SQL queries that reference nonexistent columns such as `reviewerID` or `name`. No executable query is generated across multiple attempts.
Reward: 0.00 *Steps:* 10 *Outcome:* Failure

Ours: The agent verifies table structures, identifies that reviewer names exist in `reviewer` and the ratings in `rating`, and constructs a correct join via `rID`. It executes a clean and fully functional query that returns the precise list of reviewer names.
Reward: 0.75 *Steps:* 5 *Outcome:* Failure

Hard Negative Justification: This trajectory exhibits correct schema interpretation and valid join construction. It reaches the correct SQL answer despite being labeled as failure, capturing the ideal form of a hard negative.

Gymnasts Ordered by Ascending Height

Instruction: Return the names of gymnasts ordered by their height in ascending order.

ETO: The agent attempts to query `gymnast` directly, repeatedly referencing nonexistent columns such as `name` and `height`. Despite multiple table inspections, it does not recognize that height and names reside in the `people` table rather than `gymnast`. It ends without producing any usable SQL.
Reward: 0.00 *Steps:* 6 *Outcome:* Failure

Ours: The agent correctly identifies that the `people` table contains both `Name` and `Height`. It inspects both `gymnast` and `people` schemas, realizes only `people` contains height values, and issues a valid query ordering by height.
Reward: 0.70 *Steps:* 6 *Outcome:* Failure

Hard Negative Justification: The agent performs correct table discovery and forms a valid height-sorted query. Although labeled as failure, the trajectory is structurally aligned with the task, illustrating a precise hard-negative example.

D THE EFFECT OF HARD NEGATIVES ON CAPTURING TASK-RELEVANT SUB-SKILLS

Our qualitative analysis shows that hard negatives play a direct role in improving the DPO training process. Because these trajectories contain structured demonstrations of navigation, tool use, object manipulation, and environment preparation, the target agent receives richer gradient signals than from ETO failures alone.

In the ScienceWorld example below, the hard negative includes all intermediate actions required to grow a lemon, while the baseline failure does not progress beyond repetitive invalid actions. After

referencing these subskill-rich trajectories during DPO, the target agent begins to reproduce the same multi-step procedures and achieves the task successfully.

These findings illustrate that hard negatives function as constructive guidance within the DPO objective, enabling the agent to internalize essential subskills that are otherwise absent in standard failure trajectories.

Growing a Lemon with Cross-Pollination

Instruction: Your task is to grow a lemon. This will require growing several plants and having them cross-pollinated to produce fruit. Seeds can be found in the bedroom. To complete the task, focus on the grown lemon.

ETO - Prediction: The agent retrieves the seed jar from the bedroom, teleports to the greenhouse, and plants lemon seeds directly into the three water-filled flower pots. It then alternates between `wait` and `look around` for many steps, repeatedly issuing invalid actions such as `focus on lemon` and `pick lemon` even though no lemon ever appears in the observations. The agent never prepares soil, never manipulates the environment for pollination, and ends in a long, unproductive loop.
Reward: 0.25 *Steps:* 60 *Outcome:* Failure

ETO - Trained Failure: The agent again retrieves the seed jar and plants lemon seeds into the three pots containing only water, then repeatedly waits and checks the greenhouse. It issues multiple invalid focus actions on the lemon tree, but the environment state never progresses beyond "lemon seed in water," indicating that the preconditions for growth and cross-pollination are not satisfied. No soil preparation or environmental control is attempted, so the episode remains a shallow failure without key subskills.
Reward: 0.25 *Steps:* 49 *Outcome:* Failure

Ours - Prediction: The agent again retrieves the seed jar from the bedroom, collects soil outside using the shovel, and fills all three greenhouse pots with soil before planting the lemon seeds. It waits for the trees to reach the reproducing stage with flowers, then observes the appearance of lemons on one tree. To encourage stable pollination, it explicitly closes both the outside and hallway doors, creating a controlled greenhouse environment, and continues waiting until a lemon is present. Finally, it focuses on the grown lemon, satisfying the task's success condition.
Reward: 1.00 *Steps:* 46 *Outcome:* Success

Ours - Trained Failure: The agent retrieves the seed jar, then picks up a shovel in the greenhouse and repeatedly teleports outside to dig up soil. It transports soil back to the greenhouse and fills all three flower pots, explicitly constructing "soil + water" planting conditions before moving lemon seeds into each pot. After staged waiting, it observes that one lemon tree now bears a lemon, and repeatedly attempts to focus on or pick the lemon with over-specified object references. The growth and pollination pipeline is correct, but the episode fails due to action-format errors at the final "focus on lemon" step.
Reward: 0.50 *Steps:* 60 *Outcome:* Failure (hard negative)
