REPRESENTING EXPERTISE ACCELERATES LEARNING FROM PEDAGOGICAL INTERACTION DATA

Dhara Yu, Karthikeya Kaushik, Bill D. Thompson
UC Berkeley
{dharakyu, karthikeya.kaushik, wdt}@berkeley.edu

ABSTRACT

Work in cognitive science and artificial intelligence has suggested that exposing learning agents to traces of interaction between multiple individuals can improve performance in a variety of settings, yet it remains unknown which features of interactions contribute to this improvement. We examined the factors that support the effectiveness of interaction data, using a controlled paradigm that allowed us to precisely operationalize key distinctions between interaction and an expert acting alone. We generated synthetic datasets of simple interactions between an expert and a novice in a spatial navigation task, and then trained transformer models on those datasets, evaluating performance after exposure to different datasets. Our experiments showed that models trained on pedagogical interactions were more robust across a variety of scenarios compared to models trained only on expert demonstrations, and that having the ability to represent epistemically distinct agents led to expert-like behavior even when expert behavior was rarely observed.

1 INTRODUCTION

Much of what we know about the world comes from observing interactions between other people (Chuey & Gweon, 2025). Examples of this form of learning range from listening to a question-and-answer session between a speaker and an audience member, to reading a comment thread on a cooking blog. Even young children are capable of learning from third-party observations, showing the ability to infer complex causal structure from overheard conversations (Bonawitz et al., 2011). Observing interactions between others has been argued to be crucial to the acquisition of language itself, as these interactions constitute a more substantive portion of child language data than has traditionally been assumed (Foushee & Srinivasan, 2024).

There is a striking parallel between this setting in human learning and large language model pre-training on internet corpora. Analogous to a child overhearing a conversation, LLMs are exposed to traces of interaction generated by other agents (e.g. in online forums), where the learning agent itself is not a participant in the interaction. Exposing LLMs, both in-context and through fine-tuning, to *additional* traces of interaction can improve performance in a variety of tasks, such as formal reasoning (Du et al., 2024; Subramaniam et al., 2025), reading comprehension (Khan et al., 2024), and moral dilemmas (Liu et al., 2023).

These findings raise the question of what structural properties of interaction might contribute to improved performance in LLMs. Work in cognitive science has suggested that learning from interactions might be useful because interaction exposes uniquely informative content beyond what is surfaced by a single agent acting alone, such as corrective feedback and recovery from mistakes (Fox Tree, 1999). This problem has also been studied within the robotics community, motivating approaches such as DAgger, which captures the idea that novice exploration exposes more of the state space (Ross et al., 2011). In human learners, corrective events are most likely to happen in interactions with knowledge asymmetries, e.g. *pedagogical* interactions between novices and experts, in which experts intervene on suboptimal behavior from novices. Cognitive models also suggest that for learning agents to benefit from these rich forms of data, it helps to be endowed with social reasoning capabilities, such as an ability to represent data-generating agents' differing levels of expertise (Shafto et al., 2014; Landrum et al., 2015).

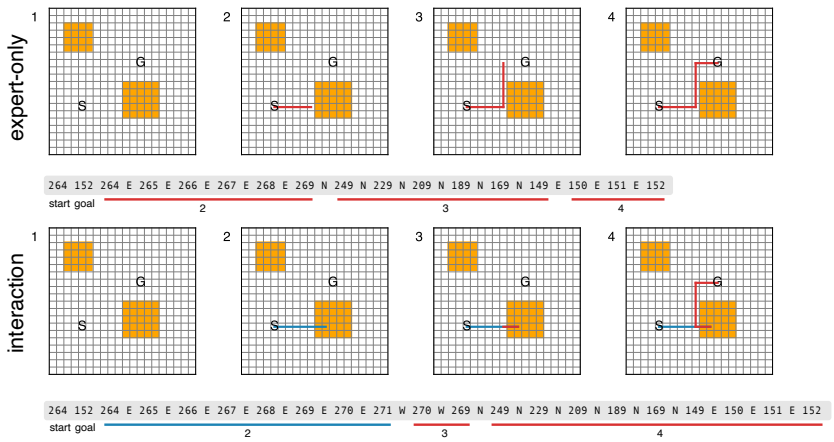

Figure 1: Example traces under expert-only and interaction policies. Grids show the agent's trajectory at 4 different time steps. Sequences are the representations of the full trajectory seen by the model, with the underlines indicating the part corresponding to the sub-trajectories at each timestep.

Here we evaluated if these factors also modulate the effectiveness of interaction data for language models, using a controlled paradigm inspired by past work characterizing representations in LLMs (Vafa et al., 2024). We generated synthetic datasets of agents interacting in a spatial planning task using symbolic planning algorithms, which captured the core sequential structure of natural language interaction. We manipulated the structure of the generated examples to capture the key distinctions between experts acting alone and multiple individuals interacting, focusing on a particular type of interaction: experts intervening on the behavior of a novice. We also manipulated the datasets to encourage distinct representation of agent types. Then, we trained transformer language models on these datasets and evaluated models' performance on the task in a variety of scenarios.

Our results revealed that models trained on traces of interaction were more robust in scenarios that an expert acting alone would be unlikely to encounter. Furthermore, we found that interaction data provided an even more expansive benefit when there were clear indicators of which agent (e.g., novice or expert) generated the data. This highlights potential mechanisms through which training on interaction data yields improved performance, and raises important questions for future work.

2 METHODS

2.1 SIMULATING INTERACTIONS

Our goal is to study when learning from interactions between novices and experts can lead to distinct benefits over learning from an expert acting alone. To do so, we needed a setting that allowed us to generate controlled datasets of *expert-only* and *interaction* behaviors, which could then be used as training data for a transformer. We generated this data in the context of a generic planning task. The goal of the task was to find the optimal path between a start point and a goal point, in a particular grid environment. Each grid contained certain high-cost cells; to maximize reward in the task, agents needed to take the most direct path to the goal that avoided costly cells. This task was suitable because interaction could be operationalized in a meaningful way, for example through correcting the trajectory of a novice. While simplified, this setting captures the core dynamics of pedagogical interactions unfolding over time in natural language datasets, e.g. Stack Overflow threads in which expert programmers help novices debug.

Here, we describe the procedure for generating datasets. We formulated the task as an MDP, where the set of states is defined by a 20×20 grid, with set of high-cost states H and a goal state g. The reward function $R_{H,g}$ assigns $+100$ for reaching g, -20 for states in H, and -1 otherwise (see Appendix A.2 for formal definitions). A trajectory τ is an alternating sequence of states and actions representing a path from the start state s_0 to the goal state $g = s_T$: $\tau = (s_0, a_0, s_1, a_1, \ldots, a_{T-1}, s_T)$.

Next, we describe the expert-only and interaction policies used to generate trajectories: The *expert-only* policy π^* represents the optimal policy under the true reward function $R_{H,g}$ (Figure 1, top). The

interaction policy π_{int} represents the behavior of a novice acting under an incorrect reward function, who is then corrected by an expert if behavior diverges from what would be optimal under the true reward function. Intuitively, interaction is operationalized as the expert modifying the policy of the novice in a way that makes their ensuing behavior expert-like. First, we define the initial novice policy π_N as the optimal policy under incorrect reward function R_{naive} which does faithfully capture the intended goal state g but ignores the set of high-cost states H. The interaction policy π_{int} switches from π_N to π^* at timestep t^*, the timestep at which the agent has been in a high-cost state for 2 consecutive actions (Figure 1, bottom). If this condition is not met (because the trajectory under the novice policy incidentally avoids high-cost states), then no intervention occurs.

To construct training datasets, we generated a set of traces under a specific MDP (there were 10), following policy π^* or policy π_{int}. To construct a single trace, we sampled a start location and a goal location, generated the trajectory under the designated policy, and prepended the start and goal locations to the trajectory: (s_0, g, τ) (Figure 1). We sampled start and goal locations so that neither the start nor the goal state could be a high-cost state for the given MDP. This was to capture the intuition that an expert agent would not be tasked with navigating from a high-cost state, because it would have full knowledge of the environment and consequently avoid high-cost states altogether.

2.2 Training and Evaluation

We trained autoregressive transformer models on the traces generated under the expert-only policy and the interaction policy (see A.4 for additional details). To elicit model predictions, we provided the start location, the goal location and the first location of the path (which was the start location).

Unlike the planning algorithms used to generate training data, the learning models were not explicitly provided with the structure of the environment. Performing well on a held-out test set therefore required learning both the structure of the environment and the expert. To evaluate performance on the task, we used **exact match**, which was true of a model output if the produced trajectory was identical to the one under the ground-truth expert policy. We also used a more relaxed **correct path** metric, which was true if the trajectory was valid (e.g. it respected the allowed transitions between cells), and the trajectory ended in the goal state. This metric was diagnostic of whether a model had learned the constraints of the environment, not necessarily whether it had learned the optimal policy.

3 Results

3.1 Study 1

To evaluate how training on different datasets affected a model's ability to produce an optimal trajectory, we constructed 3 different test sets consisting of (start, goal) pairs. In *safe* trials, the provided start and goal states were not from the set of high-cost states, and the expert policy and the interaction policy both prescribed the same trajectory, meaning that the associated novice trajectories incidentally avoided entering any high-cost state. In *hazardous* trials, the start and goal states were similarly not high-cost states, but following the novice policy would involve traversing through high-cost states. Finally, in *recovery* trials, the start states were high-cost states while the goal states were not, probing the ability of the model to produce valid trajectories from states that would not be traversed by an expert. Each test set contained 320 held-out examples.

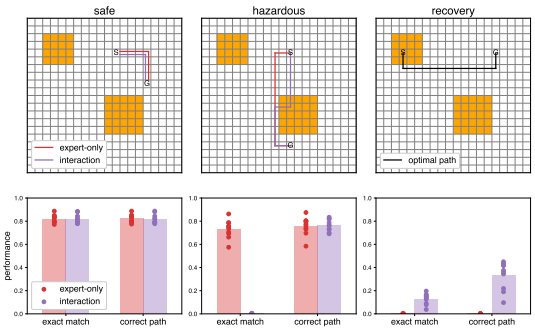

Figure 2: Study 1 results.

Figure 2 shows model performance across the 3 test sets. In the safe trials, expert-only trained models and interaction-trained models generally performed well, confirming that in favorable conditions, models could succeed in the task. In hazardous trials, expert-only models generally learned to avoid high-cost states, but interaction models did not, though they did produce valid trajectories: the raw input stream of actions and states generated under an interaction was insufficient to learn the generic behavior of avoiding high-cost states.

Recovery trials showcased the value of learning from exploration and recovery events in interaction data. In these trials, expert-only models failed to produce valid sequences. The intervention models, on the other hand, generated optimal trajectories for some (start, goal) pairs, and for a larger percentage of coordinate pairs, produced valid trajectories. This occurred even though these models were not trained on direct examples of optimal paths between a high-cost state and a goal state. One interpretation of the performance gap across model types is that expert-only data was not sufficiently constraining to infer that the optimal behavior in a high-cost state is to exit the state.

Overall, this pattern of results illustrates a tradeoff that arises when learning from two structurally different forms of input. Learning solely from traces of experts enables more expert-like behavior for parts of the state space that are well-traversed by the expert during learning. In contrast, learning from interaction can in principle enable recovery from suboptimal areas of the state space that an expert would not enter. This capacity for recovery could be particularly advantageous in settings where one may need to intervene on *another* agent's incorrect model, for example in teaching contexts.

The poor performance of interaction-trained models in hazardous trials suggests that learning from action sequences in the context of an interaction may be insufficient to acquire expert-like performance in some settings. In the next section, we explore how this can be ameliorated by training on datasets that reveal information about the agent that generated a particular sequence of actions.

3.2 Study 2

Here our goal was to evaluate if information about the agent generating the data can improve performance in hazardous trials, through learning the generic policy of avoiding high-cost states. To do so, we manipulated the training data to include explicit indicators of the source of certain actions in the trajectory, i.e. whether actions were brought about as a result of a novice or an expert. Specifically, we designated source indicator tokens to prepend at the beginning of agent trajectories. An "expert" indicator was prepended to the beginning of expert-only sequences, while a "novice" indicator was prepended to the beginning of intervention sequences. Then, for the intervention traces in which the expert took corrective action (because the novice proceeded through high-cost states), the beginning of the corrected sequence was prepended with the "expert" indicator (Figure A1). The tokens "novice" and "expert" do not have inherent semantics analogous to the meaning of those words in English; they are just indicators of 2 distinct types. This approximates scenarios in which there are outwardly visible cues in the data that indicate expertise or lack thereof.

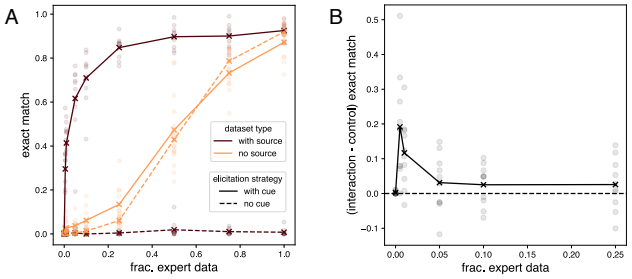

Figure 3: Study 2 results. A: performance on hazardous trials, with and without agent source indicators. B: difference in exact match accuracy (percentage points) between the models trained on interaction datasets and models trained on the matched control datasets, also on hazardous trials.

We hypothesized that revealing source information would be most helpful in settings where expert-only data is particularly scarce - a common situation for many specialized tasks. The intuition is that when there are only a few samples exhibiting optimal behavior, it helps to have an explicit signal indicating which traces illustrate that behavior. To test our hypothesis that separately representing agent types enables better generalization when exposed to limited expert-only data, we synthesized several datasets which varied in two respects (Figure A1). First, the data varied in whether the traces contained source indicators revealing the type of agent generating the trajectory (*with-source*) or not (*no-source*). Second, the data varied in the fraction of traces displaying expert-only or interaction behavior, ranging from containing only expert-only traces, to no traces generated under the expert policy (and being comprised solely of traces generated under the interaction policy).

For each unique composition, we constructed a test set of hazardous trials comprised of (start, goal) pairs that had not previously been seen together in training. We used 2 different strategies to elicit model predictions. Under the *no-cue* strategy, the model was provided with the start location and goal location, same as in Simulation 1. Under the *with-cue* strategy, the model was provided with

the start location, the goal location, the expert token, and then the start location again; this was done to explicitly condition the generation on an agent type (in this case, the expert type).

3.2.1 Study 2A: Learning from differentiated agents improves performance when expert-only behavior is rare

Models trained on with-source datasets outperformed models trained on no-source datasets, under the most favorable elicitation strategies for each dataset type (Figure 3A). This performance gap was most pronounced when the proportion of expert-only data was low; even when expert-only traces constituted just 0.5% of the training data (500 traces), models trained on with-source datasets produced optimal trajectories a mean of 30% of the time, whereas models trained on no-source datasets virtually never produced the correct trajectory. There was no difference in the no-source models' performances with or without the expert cue. In both cases, the percent of correct traces is linearly proportional to the fraction of the data that is expert-only. This highlights a fundamental limitation of learning from no-source data in this setting: the likelihood of producing the desired behavior is bounded by the frequency of that behavior in the training data.

Models trained on with-source datasets achieve better-than-linear performance (relative to the fraction of expert-only demonstration observed during training) when provided with an expert cue, but fail entirely without this cue. This suggests that models have learned to rely on the source indicator to predict the ensuing behavior and therefore in its absence cannot do so. To overcome this brittleness, we ran an additional experiment in which we varied the frequency with which the source indicators appeared in the data (see A.5.1 for details). We found that reducing the frequency of source indicators led to models reaching ceiling performance in the no-cue condition (e.g. performance was linearly proportional to the fraction of the data that was generated by an expert; Figure A2). This suggests that training on partial, rather than full, source information improves robustness.

3.2.2 Study 2B: Interactions are more useful than single-agent traces in expert-scarce settings

Study 1 showed that learning from pure interaction data is strictly better than learning from pure expert-only data when the task involves recovering from a state that would not have been encountered by an expert in the first place. This raises the question of whether the properties of interaction data make it useful for learning more generic behaviors, namely the task of avoiding high-cost states altogether. To study this, we constructed a set of with-source *control* datasets, which were matched to with-source interaction datasets (introduced in Study 2A) in the percent of the data that was generated under the expert policy. The key difference was that in the control datasets, the remaining data was generated under the novice policy, rather than the interaction policy (e.g., the control dataset with 5% expert-only data contained 5000 expert-only traces and 95,000 novice-only traces).

Using the with-cue method to elicit predictions, we found that training on interaction data yielded higher performance when the proportion of expert-only data was low (5% or less; Figure 3B). Beyond that threshold, training on the control data was just as informative. We confirmed that this difference was driven by differences in the content of interaction traces, rather than the amount of training data (see A.6). This highlights another, more indirect situation where learning from interaction is beneficial: when interaction surfaces a related, but not directly illustrative behavior (e.g. recovering from a costly state, vs. avoiding costly states altogether). This facilitates learning the target expert behavior (avoiding costly states) when demonstrations of the expert behavior are scarce.

4 Discussion

Here we established a set of key properties that support the effectiveness of interaction data: 1) information asymmetries that create the opportunity for recovery events, which in turn exposes more of the state space in a way that is useful for learning, and 2) a capacity on the part of the learner to represent the distinct agents who generated separate parts of the interaction data. An important direction for future work is to validate these findings in more naturalistic settings, e.g. by modifying natural language interaction datasets along the dimensions identified by this analysis as improving performance, and training models on those datasets. These results represent an initial step toward understanding the value of learning from observed interactions, for both humans and AI systems.

REFERENCES

Elizabeth Bonawitz, Patrick Shafto, Hyowon Gweon, Noah D Goodman, Elizabeth Spelke, and Laura Schulz. The double-edged sword of pedagogy: Instruction limits spontaneous exploration and discovery. *Cognition*, 120(3):322–330, 2011.

Aaron Chuey and Hyowon Gweon. Theory of minds: early understanding of interacting minds. *Developmental Psychology*, 7(1):91–115, 2025.

Yilun Du, Shuang Li, Antonio Torralba, Joshua B Tenenbaum, and Igor Mordatch. Improving factuality and reasoning in language models through multiagent debate. In *Forty-first International Conference on Machine Learning*, 2024.

Ruthe Foushee and Mahesh Srinivasan. Infants who are rarely spoken to nevertheless understand many words. *Proceedings of the National Academy of Sciences*, 121(23):e2311425121, 2024.

Jean E Fox Tree. Listening in on monologues and dialogues. *Discourse processes*, 27(1):35–53, 1999.

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. *arXiv preprint arXiv:2402.06782*, 2024.

Asheley R Landrum, Baxter S Eaves, and Patrick Shafto. Learning to trust and trusting to learn: A theoretical framework. *Trends in Cognitive Sciences*, 19(3):109–111, 2015.

Ruibo Liu, Ruixin Yang, Chenyan Jia, Ge Zhang, Denny Zhou, Andrew M Dai, Diyi Yang, and Soroush Vosoughi. Training socially aligned language models on simulated social interactions. *arXiv preprint arXiv:2305.16960*, 2023.

Stéphane Ross, Geoffrey Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pp. 627–635. JMLR Workshop and Conference Proceedings, 2011.

Patrick Shafto, Noah D Goodman, and Thomas L Griffiths. A rational account of pedagogical reasoning: Teaching by, and learning from, examples. *Cognitive psychology*, 71:55–89, 2014.

Vighnesh Subramaniam, Yilun Du, Joshua B Tenenbaum, Antonio Torralba, Shuang Li, and Igor Mordatch. Multiagent finetuning: Self improvement with diverse reasoning chains. *arXiv preprint arXiv:2501.05707*, 2025.

Keyon Vafa, Justin Y Chen, Ashesh Rambachan, Jon Kleinberg, and Sendhil Mullainathan. Evaluating the world model implicit in a generative model. *Advances in Neural Information Processing Systems*, 37:26941–26975, 2024.

A APPENDIX

A.1 RELATED WORK

A.2 FORMAL DEFINITION OF TASK

Formally, we define a set of Markov Decision Processes (MDPs), where each MDP is defined by the tuple $\langle \mathcal{S}, \mathcal{A}, T, R_{H,g} \rangle$. The state space $\mathcal{S}$ is the set of cells in a 20x20 grid. The action space $\mathcal{A}$ is the set of the cardinal directions $\{N, S, E, W\}$. The transition function $T(s'|s, a)$, represents the probability of entering state s' from state s after taking action a: here we assume that this function is deterministic, and agents always transition to the intended state. Finally, we define reward function $R_{H,g}(s)$, which is parameterized by the high cost cells $H \subset \mathcal{S}$ and goal state g:

$$R_{H,g}(s) = \begin{cases} +100 & \text{if } s = g \\ -20 & \text{if } s \in H \\ -1 & \text{otherwise} \end{cases}$$

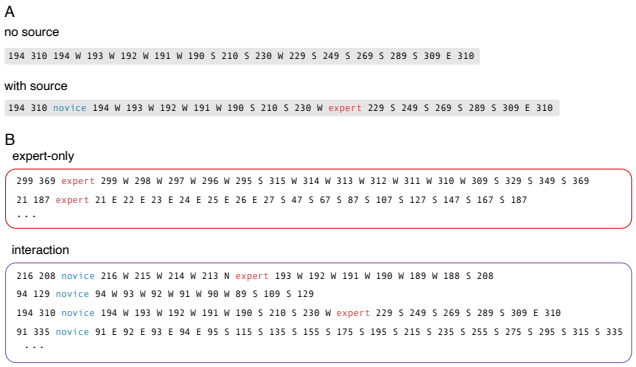

Figure A1: Overview of datasets with source indicators. A: example of a no-source trace and the corresponding with-source trace. B: depiction of an example dataset containing expert-only data and interaction data.

A.3 Grid generation procedure

We generated 10 unique grids, where the high-cost set $H \subset S$ is constructed by sampling $n \sim \text{Uniform}(1, 4)$ rectangular zones. The dimensions of each zone are sampled as $r, c \sim \text{Uniform}(3, 6)$, and their positions are sampled uniformly at random within the grid bounds.

A.4 Additional training and evaluation details

We used a smaller version of the LLaMa autoregressive transformer architecture, with hidden size layers of length 128, 4 hidden layers and 4 attention heads (a total of approximately 17.6 million parameters). We defined a vocabulary which contained a unique token for each of the 400 unique states as well as a token for each of the 4 possible actions corresponding to the cardinal directions. For all experiments, models were trained on 100,000 generated traces for 10 iterations (to convergence), with a batch size of 16. Separate models were trained for each of the 10 unique grids and 2 policies per grid, for a total of 20 trained models. Model predictions were generated using greedy decoding.

A.5 Additional results

A.5.1 Learning from differentiated agents improves performance under noisy differentiation

The higher performance ceiling of models trained on with-source data highlights the value of information about the type of agent generating the observed data. Yet learning agents may not always have access to unambiguous type information; for example, a true novice may not be able to distinguish between experts and less-skilled individuals. To capture these dynamics, we conducted an additional experiment in which we varied the frequency with which the tags appeared in the training data. Specifically, we reconstructed the training datasets so that for each occurrence of an agent tag, there was a $0 < k < 1$ probability that the tag was visible during training.

Under the with-cue condition to elicit trajectories, models trained on datasets with partial source information outperformed the linear trend even as the frequency of source indicators was reduced (Figure A2). This indicates that in this setting, veridical source information is useful at any volume, rather than being an all-or-nothing phenomenon.

We also investigated how training on datasets with partial rather than full source information affected performance, under the no-cue condition. When k, the likelihood of a source indicator token being visible in the data, was less than 1, models reached ceiling performance in the no-cue condition (e.g. performance was linearly proportional to the fraction of the data being generated by an expert). This highlights a key benefit of learning from data with partial, but not full source information: reducing the frequency of source indicators renders learning agents less dependent on these indicators for

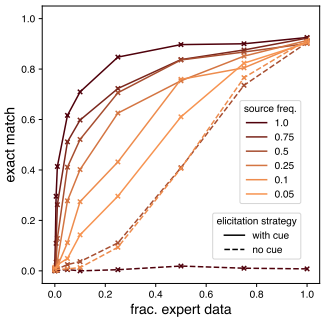

Figure A2: Performance on hazardous trials with varying frequencies of agent indicator tokens, under both prediction elicitation strategies.

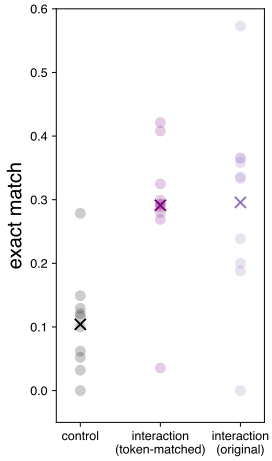

Figure A3: Performance on hazardous trials for models trained on 3 types of datasets: control datasets (500 expert-only traces, 99,500 novice-only traces), token-matched interaction datasets (500 expert-only traces, 93,500 interaction traces; total number of tokens is matched with control datasets), and the original interaction datasets (500 expert-only traces, 99,500 interaction traces).

producing correct trajectories, and therefore more robust in the no-cue condition. These results are consistent with the idea that learning from the partial datasets induces 2 modes: 1) relying on knowledge about the source to generate more accurate predictions when this information is available, in the with-cue condition, and 2) defaulting to the most common strategy seen in training in the absence of a cue.

A.6 Benefits of interaction data over single-agent data result from differences in content, not length

Interaction traces contain on average 6% more tokens than single-agent traces (mean length of a trace: 30.8 vs. 29.1). To confirm that the improvement in performance from training on interaction datasets over single-agent datasets was due to the structural properties of interaction, rather than higher volumes of data, we ran an additional follow-up experiment. We constructed an additional set of interaction datasets that were matched, in the number of expert traces, to the original interaction datasets consisting of 0.5% expert traces and the control datasets with 0.5% expert traces (a total of 500 expert traces). The key difference between these new interaction datasets is that the total number of tokens was matched to the total number of tokens in the 0.5% expert control datasets (approximately 29 million tokens). Models trained on the token-matched interaction datasets performed substantially better than models trained on the control datasets (Figure A.6), confirming that the value of interaction data came from its structural properties.