		Accuracy				Precision				Recall				F-Score			
O	Domain	MU	KL	DP	RG	MU	KL	DP	RG	MU	KL	DP	RG	MU	KL	DP	RG
0.5	Blocks	**0.95**	0.62	0.93	0.84	**0.95**	0.33	0.77	0.56	0.90	0.50	**1.00**	**1.00**	**0.90**	0.40	0.87	0.71
	Hanoi	**0.97**	0.90	0.93	0.68	**0.91**	0.80	0.77	0.38	**1.00**	0.80	**1.00**	**1.00**	**0.95**	0.80	0.87	0.56
	SkGrid	0.75	0.75	0.57	**0.88**	0.50	0.50	0.35	**0.64**	0.50	0.50	0.80	**0.90**	0.50	0.50	0.48	**0.75**
1.0	Blocks	**1.00**	**1.00**	0.95	0.96	**1.00**	**1.00**	0.83	0.83	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	0.91	0.91
	Hanoi	**1.00**	**0.95**	0.90	0.78	**1.00**	0.90	0.71	0.48	**1.00**	0.90	**1.00**	**1.00**	**1.00**	0.90	0.83	0.65
	SkGrid	0.85	**0.95**	0.65	0.90	0.70	**0.90**	0.40	0.69	0.70	**0.90**	0.80	**0.90**	0.70	**0.90**	0.53	0.78
Avg	Blocks	**0.97**	0.81	0.94	0.90	**0.97**	0.60	0.80	0.70	0.95	0.75	**1.00**	**1.00**	**0.95**	0.67	0.89	0.81
	Hanoi	**0.99**	0.93	0.91	0.73	**0.95**	0.85	0.74	0.43	**1.00**	0.85	**1.00**	**1.00**	**0.98**	0.85	0.85	0.61
	SkGrid	0.80	0.85	0.61	**0.89**	0.60	**0.70**	0.37	0.67	0.60	0.70	0.80	**0.90**	0.60	0.70	0.51	**0.77**

Table 2: Impact of noise: comparing MU, KL, DP, RG with varying observability and with 4 noisy observations in $\boldsymbol{O}$.

Related Work

A large body of work involves learning for *planning* domains (**??**). While some approaches learn action models from data, they do not link these action models to policies for reaching specific goals (**????**).

For example, Zeng et al. (**?**) use inverse reinforcement learning (IRL) to learn the rewards of the actor and then use an MDP-based GR. However, for GR, the motivation (rewards) that lead the actor to choose one action over another is redundant. By directly using RL, we skip this stage and learn utility functions or policies based on past actor experiences towards achieving specific goals. Amado et al. (**?**) learn domain theories for GR using autoencoders. However, they require observation of all possible transitions of a domain in order to infer its encoding, whereas we need only a small sample of transitions to learn a utility function informative enough to carry out GR effectively.

Unlike approaches that learn models for planning, we do not reason about the plan of the acting agent, but rather about the plan of another agent. In this case, we cannot control the actor's choices, and we might not know or care how the actor represents the environment and the task. Nevertheless, we need to be able to find a good-enough explanation for its actions to be able to assist it (as in the kitchen example from Figure 1). This setup is not the one used in existing work on learning other agent's behavior, e.g., the LOPE system (**?**), (**?**), and IRALe (**?**), as these systems choose the execution se it learns from. We can, however, use observed actions of other agents to improve our learning process. Gil (**?**) does so by investigating cases where executing new experiments can refine operators.

Other metric-based GR use distance metrics between an optimal plan and the observation se which can somewhat alleviate the need in online planner executions (**??**). This work differs from this problem statement, as it relies on the distance between a Q-function and an observation sequence rather than an optimal plan and an observation sequence.

Discussion and Conclusion

Recognition (GR) as model-free reinforcement learning, which obviates the need for an explicit model of the environment and candidate goals in the GR process. Our framework uses learned Q-values implicitly representing the agents under observation in lieu of explicit goals from traditional GR. This approach allows us to solve GR problems by minimizing the distance between an observation sequence and Q-values representing goal hypotheses or policies extracted from them. The GRAQL instantiation includes several possible distance measures we can derive from the Q-tables, based on KL-divergence, MaxUtil, and Convergence point. Our distance measures are competitive with the reference approach from the literature (**?**) in all experimental environments, and some distance measures outperform the reference approach in most domains, especially when the observation sequence is noisy or partial.

Besides recognition performance, GR needs to be computationally efficient so that an observer can quickly make decisions in response to the recognized goal in real-time. In this respect, our approaches differ substantially from recent planning-based GR, as it shifts almost all computation load to a pre-processing stage, instead of costly online planner runs. While computing the policies for each candidate goal is even more costly than running a planner for each goal, this computation can be done once prior to the recognition stage, and then the computation of processing observations is trivial and proportional to the number of observations. Finally, learning the policies saves the time of a domain expert who needs to carefully design the planning dynamics—a cost which is even harder to quantify.

In closing, our work paves the way for a new class of GR approaches based on model-free reinforcement learning. Future work will focus on new, more robust distance measures and mechanisms to handle noise explicitly, as well as experimenting with models learned using function approximation (e.g., neural networks). While our work is theoretically compatible with non-tabular representations of the value functions, we chose to focus our experiments on domains that are translatable to PDDL so our approach can be compared to planning-based GR. GRAQL does not depend on PDDL, as we can apply the learning stage on any domain where we can compute a policy, and then infer the correct goal using the set of observations and learned policies.

Ea deserunt perferendis modi, mollitia numquam deserunt incidunt ut doloremque voluptates ab amet aliquam atque natus, magni eum expedita eius cupiditate eaque?Hic nostrum illum quidem laboriosam dolores voluptatum odit molestiae ullam, inventore labore ducimus odio, officiis eligendi omnis odio, reiciendis cumque eaque atque itaque reprehenderit cupiditate autem, similique nesciunt suscipit vitae?Quisquam nobis tempora hic in quam nemo error, quibusdam nemo consequuntur placeat, quam ullam officiis

obcaecati sit hic fuga magni quo sunt, nihil facere nemo perferendis inventore odio neque eum vitae adipisci incidunt.