We thank all the reviewers for their insightful comments, suggestions, and positive feedback. We will update the paper based on the constructive comments provided. We start by addressing a common point regarding the use of OpenIE methods for text-based games, and then address questions from individual reviewers.

Comparison with OpenIE-based agents. One common point raised in the reviews is the relationship between our approach and approaches that use OpenIE algorithms for graph construction, such as KG-DQN [4] and KG-A2C [3]. These are indeed important prior works, and the key distinction between GATA (our work) and [4] and [3] is that they rely on handcrafted filters combined with OpenIE algorithms, whereas GATA is entirely data-driven. We actually put significant effort into including [4] as a baseline in the months prior to our submission. However, unfortunately, we were unable to faithfully reproduce their model or achieve non-trivial performance on our evaluation games. These problems persisted even after contacting the original authors for assistance, and we believe two key issues were (a) the hand-crafted filters in [4] not generalizing across games and (b) substantial scalability issues. More generally, off-the-shelf OpenIE tools are suboptimal for maintaining dynamic knowledge graphs in text-based games; in particular:

- **Low recall**: Tools like OpenIE can hardly cover certain node types appearing in text-based games. Some nodes that describe states of objects are absent from $O_t$. For instance, the resulting observation of making a meal will be “you made a meal”, but the “consumed” states of the ingredients are absent from $O_t$ thus require reasoning. Typically, such types of nodes are not entities (nor even noun phrases), making it difficult for tools like OpenIE to detect them.

- **Low precision**: OpenIE generates a large amount of unrelated tuples (e.g., from a sentence “what is that?” it extracts nodes “what” and “that” connected by edge “is”). Empirically, a set of filtering heuristics are needed as used in [3, 4].

- **Dynamicity**: Since we are dealing with a dynamic environment, some handcrafted heuristics for maintaining/updating the knowledge graph (tuples extracted by OpenIE) are needed. For instance, after moving to a new room, the content of the KG shouldn’t be discarded but some relations need to be changed (e.g., location of the player).

Reviewer #1:

- **Analysing generated graphs**: R1 raises an important point on analysing the graphs generated. For the analyses of information encoded by the graphs, we refer them to Appendix D.5. Here, apart from the high-res visualization of the graphs, we conduct probing tasks to study the information encoded by the generated belief graphs. Results suggest the graphs can encode important information (Table 7). We will further discuss these results in the revision.

- **Other games**: An alternative testbed would be to use the curated list of text-based games supported by Jericho [14]. However, the list is small with only 50 highly diverse games. Attempting to solve the very difficult problem of out-of-distribution generalization (OODG) on diverse Jericho games was out-of-scope for this current submission, which focuses on achieving state-of-the-art results on in-domain generalization. However, improving on GATA to tackle Jericho and other more challenging generalization tasks is an objective of future work.

Reviewer #2:

- **Motivation for COC**: Both the graph updater pre-training approaches (OG and COC) aim to extract useful information that is sufficient to reconstruct the text observation. Contrastive learning has been shown to be an effective unsupervised training regime to learn useful information in several prior works [7, 43, 22]. The motivation behind contrastive unsupervised training is that one does not require to train complex decoders. For example, compared to OG, the COC’s objective relaxes the need for learning syntactical or grammatical features and allows GATA to focus on learning the semantics of the $O_t$. This is an important point, and we will clarify this motivation in the revised main text.

- **Graph as memory**: We agree with the reviewer that a graph-based representation also serves as a structured memory for the agent. In particular, we believe such structured representations (KG) generalize well for text-based games owing to two reasons: a) they serve as a good inductive bias as the states can be factorized in terms of nodes and relations; and b) they act as a memory which aids the agent to act effectively in a partially observed scenario.

Reviewer #3:

- **Scalability**: You raise some important clarification points regarding scalability. One important clarification is that at every step, the belief graph $G \in [-1, 1]^{N \times N \times N}$ is decoded using $f_d$ from a single vector $h_t$ (as illustrated in Fig 2), and the size of this vector $h_t$ is significantly smaller than the full tensor (i.e., dimension 64, as described in appendices). We use this low-dimensional vector as the model’s recurrent state, which alleviates scalability issues during learning. More generally, regarding scalability and relationships to traditional knowledge base construction (KBC), it is important to note that our approach is more related to recent work on neural relational inference (NRI, Kipf et al., arXiv:1802.04687) than traditional KBC; in particular, we seek to generate task-specific graphs, which tend to be dynamic, contextual and relatively small, whereas traditional KBC focus on generating large, static graphs. As a result, we believe discussing extrapolations of GATA’s graph extraction to IE domains is out of scope of this paper.

Reviewer #4:

Table 2: We will make the presentation of our experiments more clear in the revised version.