The Shoutcasters, the Game Enthusiasts, and the AI: Foraging for Explanations of Real-Time Strategy Players

SEAN PENNEY, JONATHAN DODGE, ANDREW ANDERSON, CLAUDIA HILDERBRAND, LOGAN SIMPSON, and MARGARET BURNETT, Oregon State University, USA

Assessing and understanding intelligent agents is a difficult task for users who lack an AI background. "Explainable AI" (XAI) aims to address this problem, but what should be in an explanation? One route toward answering this question is to turn to theories of how humans try to obtain information they seek. Information Foraging Theory (IFT) is one such theory. In this paper, we present a series of studies[1] using IFT: the first investigates how expert explainers *supply* explanations in the RTS domain, the second investigates what explanations domain experts *demand* from agents in the RTS domain, and the last focuses on how both populations try to explain a state-of-the-art AI. Our results show that RTS environments like StarCraft offer so many options that change so rapidly, foraging tends to be very costly. Ways foragers attempted to manage such costs included "satisficing" approaches to reduce their cognitive load, such as focusing more on What information than on Why information, strategic use of language to communicate a lot of nuanced information in a few words, and optimizing their environment when possible to make their most valuable information patches readily available. Further, when a real AI entered the picture, even very experienced domain experts had difficulty understanding and judging some of the AI's unconventional behaviors. Finally, our results reveal ways Information Foraging Theory can inform future XAI interactive explanation environments, and also how XAI can inform IFT.

CCS Concepts: • **Human-centered computing** → **User studies**; • **Computing methodologies** → *Intelligent agents.*

Additional Key Words and Phrases: Explainable AI, StarCraft, Information Foraging, Empirical Studies with Humans, Qualitative Analysis, Humans evaluating AI

ACM Reference Format:
Sean Penney, Jonathan Dodge, Andrew Anderson, Claudia Hilderbrand, Logan Simpson, and Margaret Burnett. 2019. The Shoutcasters, the Game Enthusiasts, and the AI: Foraging for Explanations of Real-Time Strategy Players. *ACM Trans. Interact. Intell. Syst.* 1, 1, Article 1 (January 2019), 43 pages. https://doi.org/FIXMEandOtherMetadata

1 INTRODUCTION

Suppose a domain expert with no AI background must assess an AI agent in their domain. Although a user can already observe the system without explanations, machine learning systems typically appear as "black boxes" to end users, as users are not shown why the system behaves the way

[1] This paper contains revised versions of [14, 51], comparison of their results, and new study of humans assessing a real AI.

Authors' address: Sean Penney, penneys@eecs.oregonstate.edu; Jonathan Dodge, dodgej@eecs.oregonstate.edu; Andrew Anderson, anderan2@eecs.oregonstate.edu; Claudia Hilderbrand, minic@eecs.oregonstate.edu; Logan Simpson, logansimpson1@outlook.com; Margaret Burnett, burnett@eecs.oregonstate.edu, Oregon State University, Corvallis, OR, USA.

2160-6455/2019/1-ART1 $15.00
https://doi.org/FIXMEandOtherMetadata

it does [33]. Ideally, the system could provide explanations to the human assessor in a user-friendly way to improve their mental model. If a domain expert making such assessments is not an expert in the complex AI models, there may be a gap between the knowledge they need to make such assessments vs. the knowledge they have in the domain. To close this gap, a growing area known as "explainable AI (XAI)" aims to enable domain experts to understand complex AI systems by requesting explanations. Prior work has shown that such explanations can improve mental models [33, 35], user satisfaction [2, 27], and users' ability to effectively control the system [5, 7, 34].

To create an explanation system, we set out to draw from the expertise of expert "explanation suppliers" to understand how they structure, produce, and phrase explanations in their domain. We also drew from "explanation consumers" to understand how a system that provides explanations can support users' information-seeking in the environment. The domain we used for our investigation was real-time strategy (RTS) games.

1.1 Why RTS?

RTS games are a popular test bed for artificial intelligence (AI) research, and platforms supporting such research continue to improve [72]. Part of the interest in the RTS domain stems from how challenging it is for AI, due to real-time adversarial planning requirements within sequential, dynamic, and partially observable environments [50]. In fact, among the categories for AI domains proposed by Russell and Norvig [63], RTS consistently falls among the hardest categories. As a result, the planning problem in RTS is quite complex, and planning problems are important for a wide variety of intelligent agents (e.g. AI controlled groups of robots [69]). Ideally, an interactive explanation system could help users of such systems assess when the agent is making decisions "for the right reasons," to ward off "lucky guesses" and legal/ethical concerns (see [32]).

The essence of RTS games is that players compete for control of territory by fighting for it. Each player raises an army to fight their opponents, which takes resources and leads players to *Build* new bases to gain more resources. Players also can consume resources to create *Scouting*[2] units, which lets them learn about their enemies' movements to enable *Fighting* in a strategic way. (For a more in-depth explanation of the domain, refer to Online Appendix A or Ontañón, et al. [50].)

This paper considers how an automated RTS explanation generator *should* behave, via three formative studies. We chose StarCraft II as our specific RTS because of the following attributes:

- Shoutcasters: StarCraft has risen to the point of e-sport, leading to professional play and commentary. In StarCraft e-sports, two players compete while shoutcasters provide real-time commentary, usually as a pair. Shoutcasters (**Shoutcasters Study**) can help provide insights for XAI tools. We can learn where and how shoutcasters forage in the domain to find information needed to generate their commentary and explanations—which are valued by their consumers (e-sport audiences), as evidenced by the fact that people pay shoutcasters to produce this content. We can also use the types of explanations that shoutcasters generate as a basis of comparison for what participants in our Game Enthusiasts Study actually seek.
- Information Foraging Tools: StarCraft is mature software, so we can run our **Game Enthusiasts Study** with participants investigating replays using built in tools. Further, these tools are very similar to those used by shoutcasters.
- AI: DeepMind has released libraries for AI development in StarCraft [72] and subsequently created a superhuman AI [73]. However, it does not explain itself, so we investigated how shoutcasters and a professional player went about understanding it in our **AlphaStar Study**

[2] An important element of StarCraft is the "fog-of-war" mechanic, where the environment is only partially observable (shown later in Figure 5). Thus, players must actively discover the actions their opponents are taking.

using multiple kinds of analyses, with some using the tools from the Shoutcaster Study and others using the tools from the Game Enthusiasts Study. Figure 2 clarifies this relationship.

1.2 Research Goals

To consider how people might develop an understanding of an AI agent's behavior, we drew from multiple perspectives. The Shoutcasters Study analyzed the "supply" of expert explanations in the RTS domain. The Game Enthusiasts Study investigated the "demand," or how users would request and consume information while performing agent assessment. Finally, to complement the fact that, in both the Shoutcasters and Game Enthusiasts Studies the player was a human, in the AlphaStar Study the player was an AI agent. We use these different perspectives to triangulate, compare, and contrast the results.

The overall goals of this work were to investigate the following:

(1) *What questions can commentary by shoutcasters answer and how are they composed?*
(2) *What foraging challenges do domain experts face when they assess an agent performance?*
(3) *How do the utterances and foraging behaviors of shoutcasters compare to domain experts?*
(4) *How do the behaviors of assessors differ when they evaluate a real AI, as opposed to when they evaluate humans?*

This paper also contributes the first study to investigate how expert suppliers of explanations meet the demands of explanation consumers who are assessing an RTS intelligent, from the perspective of Information Foraging Theory (IFT). IFT is a theoretical framework that can enable generalization beyond the features of a specific interface, connecting our results with other work on humans engaged in information-seeking settings. Our aim is to contribute new insights for both future XAI systems and to reveal open problems in IFT.

2 BACKGROUND AND RELATED WORK

In XAI, what are explanations actually for? One characterization is that their purpose is to improve the mental models of the AI systems' users. Mental models, defined as "internal representations that people build based on their experiences in the real world," enable users to predict system behavior [49]. In essence, the goal of XAI explanations is to improve users' mental models of a system enough to succeed at tasks requiring a rudimentary understanding, such as assessing whether an AI agent is "good enough" for their purposes.

Through their influences on users' mental models, explanations can be powerful in shaping the attitudes and skills of users. Tullio et al. [70] examined mental models for a system that predicted the interruptibility of their managers. They found that the overall structure of their participants' mental models was largely unchanged over the 6 week study, although they did discount some initial misconceptions. Bostandjiev et al. studied a music recommendation system and found that explanations led to a remarkable increase in user-satisfaction [7]. To improve mental models by increasing transparency of a machine learning system, Kulesza et al. identified principles for explaining (in a "white box" fashion) how machine learning based systems make its predictions more transparent to the user [33]. In their study, participants used a prototype based on these principles and showed up to 52% improvement in their mental model quality. Anderson et al [3] explore different explanation strategies, and the effect of these strategies on the mental models of non-AI experts about an AI agent. Among the conclusions, they found that a combination of explanation strategies can be more effective at improving mental models than using one or another.

Several studies have also found that explanations have been able to improve users' ability to control the system. Stumpf et al. investigated how users responded to explanations of machine learning predictions, finding that participants were willing to provide a wide range of feedback to improve the system [67]. Kulesza et al. found that the participants who were best able to customize recommendations were the ones who had adjusted their mental models the most in response to explanations about the recommender system [34]. Further, those same participants found debugging more worthwhile and engaging. Similarly, the study by Roy et al [62] investigates the relationship between automation accuracy and controllability of an automated task. They observed that high controllability and self-reported satisfaction remained constant even with very low accuracy.

In the domain of this paper—intelligent agents in RTS games—there is research into AI approaches [50] but little research investigating what humans need or want explained. One exception is the work of Cheung et al., who studied how people watch the RTS genre, and created personas for various types of viewers [10]. One of these personas was a shoutcaster persona, and Cheung et al. discussed how shoutcasters affect the spectator experience [10]. Another contingent of researchers is working toward applying machine learning to automatically summarize different aspects of sports, potentially assisting sportscasters or shoutcasters in their work. Examples include automatically generating e-sports statistics [68], automatically extracting football play diagrams from raw video [66], and dialog evaluation systems (e.g., [43]), which could be used by future attempts to generate automated commentary from replays to assess the quality of the generated shoutcasters. More directly relevant to our work, Metoyer et al. studied how experienced players provided explanations in the RTS domain to novice users while demonstrating how to play the game [46]. Our work draws upon the qualitative coding schemes they developed from the content and structure of the expert players' explanations.

The work most similar to one of the studies we present in this paper is Kim et al.'s study of agent assessment in StarCraft [29]. Their study invited experienced players to assess skill levels and overall performance of AI bots by playing against them. They observed differences between humans' ranking and an empirical ranking based on the bots' win rate at AI competitions. Building on that work, Kim et al. applied a similar methodology where participants compared multiple AIs (selected longitudinally from 5 years of AI competitions) by playing against each [30]. They observed that StarCraft AI were still quite weak compared to humans[3] (humans won 88% of the matches). Further, they coded reasons their participants provided for their performance scores and presented some recommendations for AI developers to improve their AIs. Our work differs from theirs in that our participants did not play (first person perspective), but instead strove to understand and explain by interacting with a game replay (third party observer perspective).

In everyday conversation, people obtain explanations by asking questions. Drawing on this point, Lim et al. categorized questions people ask about AI systems in terms of "intelligibility types" [41]. Their work investigated participants' information demands about context-aware intelligent systems powered by decision trees, determining which explanation types provided the most benefit to users. They found the most often demanded questions were why and why not (why did or didn't the system do X?). We provide more details of that work and build on it in later sections.

In recognition of the particular importance of these two intelligibility types, researchers have been working on Why and Why Not explanations in domains such as database queries [6, 24], robotics [22, 44, 61], email classification [36], and pervasive computing [71]. Many of these techniques are also applicable to models treated as a "black box" [21, 31, 60].

Other research has built upon Lim et al.'s intelligibility types by demonstrating that the intelligibility type(s) a system supports impact which aspects of users' attitudes are affected. For example, Cotter et al. found that justifying why an algorithm works the way it does (but not how it works)

[3] Very recently, AI made significant progress in this aspect when DeepMind's AlphaStar beat MaNa, one of the world's strongest professional StarCraft players, following a successful benchmark match against his teammate TLO [73]. AlphaStar played the full game of StarCraft II, using a deep neural network that is trained directly from raw game data by supervised learning and reinforcement learning.

increased users' confidence (blind faith) in the system—but did not improve their trust (beliefs which inform a full cost-benefit analysis) in the system [13]. Further, it seems that the relative importance of the intelligibility types may vary from one domain to another. For example, Castelli et al. found that in the smart homes domain, users showed a strong interest in What questions, but few other intelligibility types [9]. Finally, some of these intelligibility types have attracted attention from social scientists who seek to help ground AI researchers' efforts in cognitive theories [25, 47].

Inspired by works such as Hoffman et al. and Miller [25, 47], our paper also employs a cognitive theory lens—information foraging theory (IFT)—because explaining AI to people involves those people seeking ("foraging for") information. We chose IFT because of its prior success in both explaining people's information seeking behaviors and in informing designs to support those behaviors. IFT has a long history of revealing useful and usable information functionalities in other information-rich domains, especially web environments [57] and software development environments [16, 54, 58]. Further, IFT has been shown to be particularly pertinent to domain experts like end-user programmers attempting to understand and assess system behaviors [4, 20, 37, 38, 65]. Although IFT constructs have been used to understand human information-seeking behavior in these situations and others such as web navigation [11, 17], debugging [16, 38, 54], and other software development tasks [48, 52, 55, 58], IFT has not been used in RTS environments like StarCraft II. This is one of the gaps our work aims to help fill.

3 METHODOLOGY

To inform the design of explanation systems, we collected explanations from experts in the RTS domain (shoutcasters) and comments from game enthusiasts trying to understand an RTS game (played by human players), to investigate how their explanation-oriented behaviors aligned. We also collected data from shoutcasters and players trying to understand an RTS game played by an AI agent. Our overall research questions were:

RQ1 What information do explanation suppliers (professional shoutcasters) seek about an RTS game being played, where do they look, how do they seek it to supply explanations, and what do their explanations contain?

RQ2 What information do explanation seekers (our participants) seek about an RTS game being played, and how do they seek it?

RQ3 How do the above explanation supply from RTS experts (shoutcasters) and demand from RTS enthusiasts (our participants) align?

RQ4 How do the answers to these questions change when a real AI plays RTS games?

3.1 Shoutcasters Study: Expert suppliers of explanations

The purpose of the shoutcasters study was to understand how expert explainers in the RTS domain construct explanations. In StarCraft, as in other settings (e.g. [28]), *"commentators are valued for their ability to expose the depth of the game. ...as* ***information gatekeepers****, the commentator (along with the observer-cameraman) is the person who has the most influence on the audience."* [10]. To ensure we studied high quality explanations and gameplay, we considered only games from professional tournaments denoted as "Premier", or among the top 30, by TeamLiquid[4]. For empirical viability, we considered only matches that had games that are not hours long. Using these criteria, we selected 10 matches available with video-on-demand from professional StarCraft II tournaments between 2016 and 2017 (Online Appendix Table B.13). Because professional tournaments have differing amount of games per match, we decided to randomly select one game from each match we chose

[4] TeamLiquid is a multi-regional eSports organization that takes a keen interest in professional StarCraft II. http://wiki.teamliquid.net/starcraft2/Premier_Tournaments

for analysis. By selecting one game per match, we were able to obtain a diverse pool of shoutcasters and professional players to analyze. In our shoutcaster study, sixteen distinct individuals appeared across the 10 videos, with two shoutcasters commentating each time.

3.1.1 Analysis Methods. Our first step was to filter shoutcasters' utterances. Because shoutcasters have an audience to both inform and entertain, they tend to fill dead air time with jokes. To filter these out, two researchers independently coded 32% of statements in the shoutcaster study corpus as relevant or irrelevant to explaining the game. We achieved a 95% inter-rater reliability (IRR), as measured by the Jaccard index. (The Jaccard index is the size of the intersection of the codes applied by the researchers divided by the size of the union [26].) After achieving agreement, the researchers split up and coded the remainder of the corpus. As a result of applying this filter, each statement had at least *some* explanatory power. We did not quantify relevancy further, though it could range from events plainly visible onscreen to deep dives on players' strategic choices.

Implicit questions answered: We then coded the resulting set of shoutcasters' utterances in terms of what (implicit) questions the shoutcasters were answering with their explanations, using the Lim & Dey [40] question types. We added a Judgment code to capture shoutcaster evaluation on the quality of actions. Using this code set (detailed later in Section 4), two researchers independently coded 34% of the 1024 explanations in the shoutcaster study corpus, with 80% inter-rater reliability (Jaccard). After achieving IRR, the researchers split up the remainder of the coding (Section 4's Table 2 and Figure 3).

Explanation content: We also coded the explanation content, drawing Metoyer's analysis of explaining Wargus games [46] as a code set, and adding a few more codes to handle differences in the games played and the study structure. Two researchers independently coded the shoutcaster study corpus, one category at a time (e.g., objects, actions, etc.), achieving an average of 78% IRR on more than 20% of the data in each category. One researcher then finished coding the remainder of the corpus (summarized in Section 4's Table 4, and detailed in Online Appendix Table B.15).

Where and how they found information: To analyze where and how shoutcasters foraged for the information from which they constructed their information, we coded using IFT constructs. The "where's" came from patches, so we simply counted the casters' navigations among patches. Changes in the display screen identified most of these[5] for us automatically. For the "how's" of their foraging, we coded the 110 instances of caster navigation by the context where it took place, based on player actions—Building, Fighting, Moving, Scouting—or simply caster navigation. Two researchers independently coded 21% of the data in this manner, with IRR of 80%. After achieving IRR, one researcher coded the remainder of the data; the complete code set is given in Online Appendix Table B.16.

3.2 Game Enthusiasts Study: Seekers of explanations

For game enthusiasts' perspectives, we conducted a pair think-aloud study, where participants worked to understand and explain the behavior of an intelligent agent playing StarCraft II, a popular RTS game [50] that has been used for AI research [72]. We brought participants in as pairs because of the social norm of communicating continuously when working with another person, a norm we leveraged to gather as much verbal data as possible.

3.2.1 Study Setup. Our setting for the study was StarCraft II replay files. A StarCraft II replay file contains an action history of a game, but no information about the players (i.e., no pictures of players and no voice audio). The StarCraft II replay tool also provides functionalities allowing

[5] But if a caster points at something to bring the other's attention to it—but does not have the mouse—the viewer cannot see it. If the caster uses a keyboard shortcut to look at something, it could be another thing the viewer cannot see or notice.

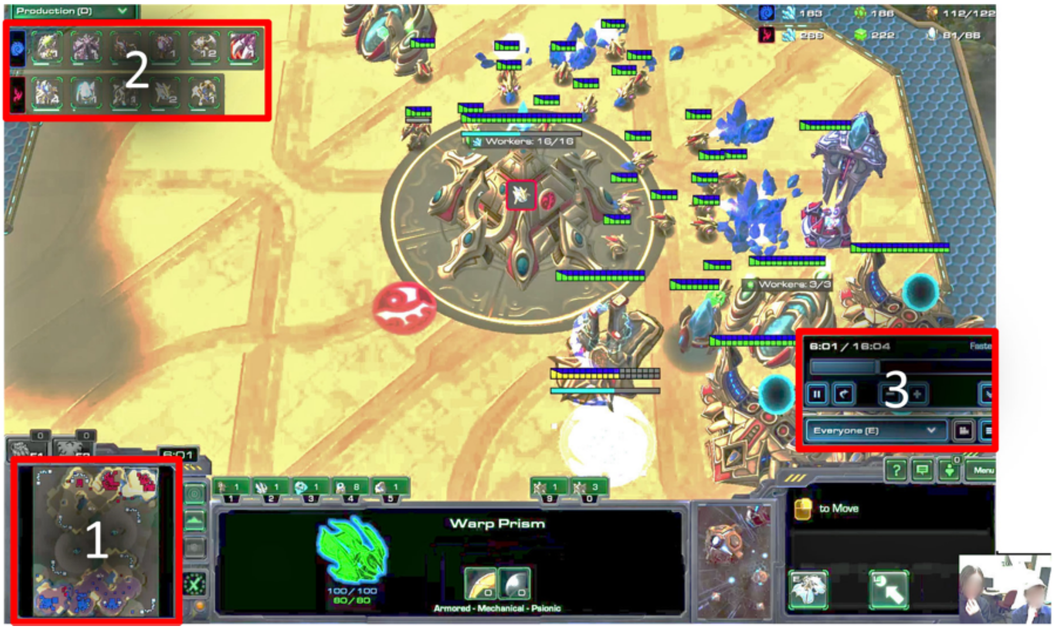


Fig. 1. A screenshot from our study, with participants anonymized (bottom right corner). Superimposed red boxes point out: (1, bottom left) the *Minimap*, a birds-eye view enabling participants navigate around the game map; (2: top left) a drop-down menu to display the *Production tab* for a summary of build actions in progress; (3, middle right) *Time Controls* to rewind/forward or change speed. Our data collection device also captured the participants' faces during the study (bottom right corner), but we have removed portions from this capture that might compromise participants' anonymity.

players to gather additional information about the replay on demand, such as by navigating around the game map, drilling down into production information, pausing, rewinding, fast-forwarding, and more, as shown in Figure 1.

For this study, we used game 3 of a match between professional players ByuL and Stats during the IEM Season XI - Gyeonggi tournament[6]. The replay we chose to analyze was a representative sample in terms of game flow, e.g., initially building up economy, some scouting, then transitioning to increasing combat [50]. Also, this match had been analyzed by shoutcasters, allowing us to to compare analysis strategies between shoutcasters and our participants, as discussed in Section 5.2.3. Further, this choice allowed us increase the size and variety—in terms of shoutcaster and player representation—of our shoutcaster study data corpus.

3.2.2 Deception: A fake AI. Our study included an element of (IRB-approved) deception: we told the participants that one of the players they were observing was an AI, even though that was untrue. To do so, we hid both (human) players' names, displaying them as Human and CPU1, and told participants that CPU1 was under AI control. To encourage them to aim for a real understanding of an agent and its weaknesses, we also told them the AI was not fully developed and had some flaws.

An alternative to deception might have been to use a replay of a game with an intelligent agent playing, but we were unable to do so because of two constraints. (1) We needed replay files that had high-quality gameplay. (2) We preferred replay files that had been analyzed by shoutcasters.

[6] The IEM tournament series is denoted as a "Premier Tournament," just as our video files from the shoutcaster study were. The replay file is public at: http://lotv.spawningtool.com/23979/.

However, at the time of our study, we were unable to locate an analyzed intelligent agent in the RTS domain with high enough quality for our investigation[7], i.e., without limitations like exploiting "a strategy only successful against AI bots but not humans" [29].

Participants believed our deception and were convinced that the player in the replay, CPU1, was an AI. For example, Pair5-P10 speculated about the implementation, "*he must have been programmed to spam.*" Some participants speculated on what advantages the AI should have:

Pair10-P19: "*Well, ideally, the AI can perfectly multitask.*"

Pair5-P10: "*Zerg player seems to be very focused on spreading creep [corruption of the terrain, granting Zerg units faster movement], and because he is an AI he does it well.*"

Participants did notice, however, the AI at times behaved like a human:

Pair10-P20: "*I've not thought of that angle for some reason: The AI trying to act like a human.*"

3.2.3 Participants. We wanted participants familiar with the StarCraft user interface and basic game elements, but without knowledge of machine learning or AI concepts, so we recruited StarCraft players at Oregon State University with at least 10 hours of prior experience—but excluding computer science students. Also, to avoid language difficulties interfering with the think-aloud data, we accepted only participants with English as their primary language. With these criteria, 20 undergraduate students participated (3 females and 17 males), with ages ranging from 19–41, whom we paired based on availability. Participants had an average of 93 hours of casual StarCraft experience and 47 hours of competitive StarCraft experience (Online Appendix Table B.14).

3.2.4 Main Task Procedures. For the main task, each pair of participants interacted with a 16-minute StarCraft II replay while we video-recorded them. Before they started, we gave them a short tutorial of the interactive replay instrumentation's capabilities (Figure 1), which allowed participants to actively forage for information within the replay. Examples of foraging options in this environment were to move around the game map, move forward or backward in time, find out how many units each player had, and drill down into specific buildings or units. Note that these features are the same as those available to shoutcasters, except time controls since shoutcasters must commentate in real-time[8].

Participants watched and foraged together as a pair to try to make sense of the agent's decisions. One participant controlled the keyboard and mouse for the first half of the replay, and then the second player took over for the second half. To help them focus on the decisions, we asked them to write down key decision points, which we defined for them as, "an event which is critically important to the outcome of the game." Whenever they encountered what they thought was a key decision point, they were instructed to fill out a form with its time stamp, a note about it, and which player(s) the decision point was about. In this study, all participants were successful in determining what a decision point should be based on our informal explanation.

3.2.5 Retrospective Interview. After the main task, we conducted a multi-stage interview based on the actions the participants took during the main task, to add context to what participants wrote down during the main task. To do so, we played parts of our recording of their main task session, pausing along the way to ask why they chose the decision points they did. The wording we used was: *"In what way(s)is this an important decision point in the game?"*

We went through the main task recording again, pausing at navigations to ask questions drawn from prior research [55]. See Table 1 for a full listing of the interview questions and their triggers.

[7] After we completed this study, a system satisfying these constraints became available, which we employed in our AlphaStar study (Section 3.3).

[8] Table 8 shows task times for game enthusiasts, alongside a comparison with shoutcaster time (equal to game length).

When a participant paused the replay...	
	- What about that point in time made you stop there?
	- Did you consider stopping on that object at any other point in time?
When a participant navigated to a patch...	
	- What about that part of the game interface/map made you click there?
	- Did you consider clicking anywhere else on the game interface/map?
When a participant navigated away from a patch (or unpaused)...	
	- Did you find what you expected to find?
	- What did you learn from that click/pause?
	- Did you have a different goal for what to learn next?

Table 1. Interview questions (drawn from prior research [55]), and the triggers that caused us to ask them.

Since there can be so many navigations in one game, we sampled pre-determined time intervals to enable covering several instances of each type of navigation for all participant pairs.

3.2.6 Analysis Methods. To investigate what information participants sought, we qualitatively coded instances in the main task where participants asked a question out loud, using the codeset outlined later in Table 2. We have already described this codeset (Section 4), because we used it on the shoutcaster data, where we achieved inter-rater reliability (IRR)—80% agreement on 34% of that corpus. Given that, the same researchers also split up the coding of the participant data (Table 5).

Questions asked: To investigate the types of information participants were seeking when asking what questions, a group of three researchers affinity diagrammed the participant data, generating six categories: Where, Information Availability, Identification, Quantification, When, and Resolving Confusion. They then used these categories as a codeset. After an IRR of 83% was achieved on more than 20% of the game enthusiasts study corpus, one researcher coded the remainder of the corpus. The code set will be detailed further Section 4.3 (Table 6).

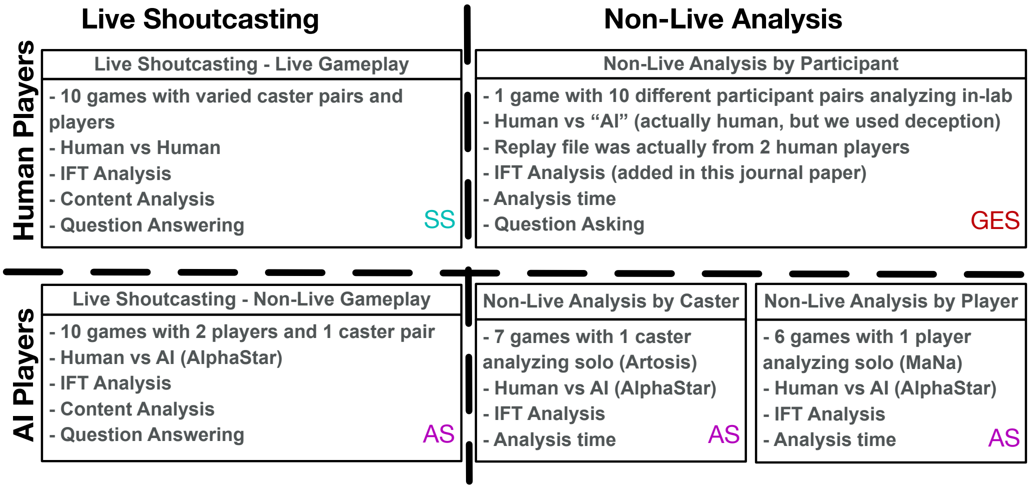
Where/how they found information: To compare participant foraging behaviors with shoutcasters, we collected the same navigations within the information interface as in Section 3.1.1 (Table 9).

Reasons to forage: To investigate which foraging paths participants took, we qualitatively analyzed their responses to the interview questions. To develop a code set to answer the question "*Why was the participant seeking information?*", a group of four researchers started with affinity diagramming to categorize participants' answers, which produced following codes: Monitoring State, Updating Game State, Obsolete Domain, and New Event. Two researchers then individually qualitatively coded the participants' responses using this code set on 20% of the data, achieving 80% agreement. One researcher then completed coding the remainder alone (Table 7).

What decisions were "key": Finally, to investigate the foraging cues that participants tied with decisions, we qualitatively coded the decision point forms that the participants used during the main task. Affinity diagramming produced four codes: building/producing, scouting, moving, and fighting. We coded 24% of the 228 identified decision points according to this code set and reached IRR of 80%, at which point one researcher coded the remaining data (Table 10, Figures 9 and 10).

3.3 AlphaStar Study

After we completed the first two studies, DeepMind released AlphaStar, an AI capable of defeating professional StarCraft players [73]. To demonstrate their achievement, they released several videos and replays from the matches. These files enabled us to obtain three new types of data, outlined in Figure 2: **1)** Live shoutcaster analysis of AlphaStar's play by two of the shoutcasters from our shoutcaster study (Artosis and Rotterdam). **2)** Non-live analysis from a player's perspective (player MaNa). **3)** Non-live analysis from a shoutcaster's perspective (shoutcaster Artosis).



Live Shoutcasting

Non-Live Analysis

Human Players

Live Shoutcasting - Live Gameplay
- 10 games with varied caster pairs and players
- Human vs Human
- IFT Analysis
- Content Analysis
- Question Answering

SS

Non-Live Analysis by Participant
- 1 game with 10 different participant pairs analyzing in-lab
- Human vs “AI” (actually human, but we used deception)
- Replay file was actually from 2 human players
- IFT Analysis (added in this journal paper)
- Analysis time
- Question Asking

GES

AI Players

Live Shoutcasting - Non-Live Gameplay
- 10 games with 2 players and 1 caster pair
- Human vs AI (AlphaStar)
- IFT Analysis
- Content Analysis
- Question Answering

AS

Non-Live Analysis by Caster
- 7 games with 1 caster analyzing solo (Artosis)
- Human vs AI (AlphaStar)
- IFT Analysis
- Analysis time

AS

Non-Live Analysis by Player
- 6 games with 1 player analyzing solo (MaNa)
- Human vs AI (AlphaStar)
- IFT Analysis
- Analysis time

AS

Fig. 2. A depiction of the data sources used in this paper, and how they relate to each other. (AlphaStar Study’s Non-Live IFT analysis is in the Supplemental Documents only.) SS = Shoutcaster Study, GES = Game Enthusiast Study, AS = AlphaStar Study. StarCraft II’s Live vs. Non-Live (replay) tools are very similar; the biggest difference is that with the Non-Live tool, an analyst can pause and move backward/forward in time.

What shoutcasters and players observed: For the non-live analysis (i.e. no time pressure, so the game can be freely rewound/paused/etc), we were not able to conduct interviews, so the approach outlined in Section 3.2.6 was not available. However, it was clear from each video on AlphaStar that the AI took actions that the players and shoutcasters thought were odd. Thus, we focused on how they approached discovering, understanding, and judging these types of behaviors.

Where and how they found information: To do this, we performed an IFT analysis as outlined in Section 3.1.1[9]. Next, we analyzed the text document shoutcaster Artosis generates during his non-live session with the replays. Note that Artosis does the live cast, and *then* does the non-live analysis, so this artifact contains his initial questions, a summary of each analyzed game, and an in depth analysis of a single AI behavior (we expand on this in Section 6). Last, we collected data on the length of each game, as well as the time to analyze that performance, to compare with our participants’ in-lab task times from Study 2. See Figure 2 for a description of the data sources used in our studies and how they relate to each other.

4 RESULTS: THE SHOUTCASTERS

Shoutcasters are both expert explainers and expert information foragers in this domain. To consider both of these roles, we expand **RQ1** *(Information shoutcasters seek to provide explanations)* into 4 parts: Implicit questions, Explanation components, Shoutcasters’ patches, and Foraging patterns.

4.1 RQ1.1 Implicit questions shoutcasters’ explanations answer

Shoutcasters crafted explanations to answer implicit questions (i.e., questions their audience “should be” wondering) about player actions. Thus, drawing from prior work about the nature of questions people ask about AI, we coded the 1024 shoutcasters’ explanations using the Lim & Dey intelligibility types [41] as per Section 3.

The shoutcasters were remarkably consistent (Figure 3) in the types of implicit questions they answered. As Table 2 sums up, the shoutcasters overwhelmingly chose to answer What, with What

[9] See our supplemental materials for these data.

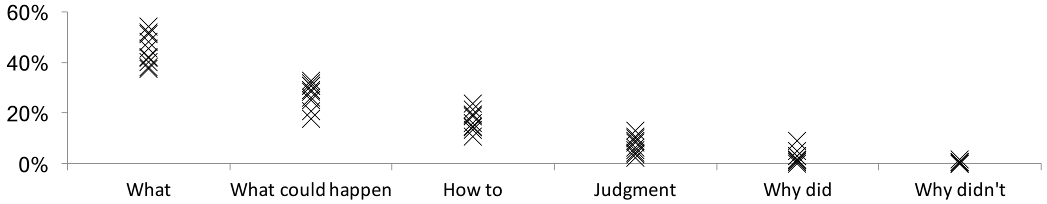


Fig. 3. Lim & Dey question frequency. Questions answered by shoutcasters, with one glyph per team. Y-axis represents percentages of the utterances which answered that category of question (X-Axis). Note how casters structured answers consistently, with both a similar frequency ranking and similar usage frequencies.

Code	Freq	Description	Example
`What`	595	What the player did or anything about game state	"The liberators are moving forward as well"
`What Could Happen`	376	What the player could have done or what will happen	"Going to be chasing those medivacs away"
`How To`	233	Explaining rules, audience tips, directives, high level strategies	"He should definitely try for the counter attack right away"
`*Judgment`	112	Evaluation of player actions	"Very good snipe there for Neeb"
`Why Did`	27	Why the player performed an action	"...that allowed Dark to hold onto that 4th base, it allowed him to get those ultralisks out"
`Why Didn't`	6	Why the player did not perform an action	"The probe already left a while ago, so we knew it wasn't going to be a pylon rush"

Table 2. Utterance types. This code set is based on the schema proposed by Lim & Dey. The asterisk denotes the code that we added, `Judgment`, because the casters judged actions based on their quality. (Note the total is greater than 1024 because explanations answered multiple questions and/or fit into multiple categories.)

Could Happen and How To also high on their list. While Lim & Dey [40] found that Why was the most demanded explanation type from participants, the shoutcasters rarely provided Why answers. More specifically, in the Lim & Dey study, approximately 48 of 250 participants, (19%) demanded a Why explanation. In contrast, in our study only 27 of the shoutcasters' 1024 utterances (approximately 3%) were Why answers, versus 595 What answers.

Although definitions on what the "What" explanation type actually is can vary (Table 3), the value of "What" to XAI is not controversial. Lim & Dey showed that AI users inquire about a system's present, past, and/or future state and behavior in a variety of ways. For example, consider an alert system to remind a user about an upcoming anniversary. Users might ask the system what values led to this decision, such as the date on the calendar and their availability that day (Table 3, Line 1). They might also ask about what the system will do about the situation, such as to flash a light to remind them, or to show a picture of their spouse (Line 2). They might also ask what else the system is doing, such as booking a table at a restaurant or ordering flowers for the significant other, which it might not display to the user (Line 3). Our definition (Line 4) aims to unify these concepts, since all of these definitions seemed to point towards the idea of *state*.

4.1.1 Discussion and Implications for a Future Interactive Explanation System.

Why so few Whys? Should an automated explainer, like shoutcasters, eschew Why explanations, in favor of What?

Author	**Description**	**Source**
B. Lim	*What is the value in the context?*	[42]
B. Lim & A. Dey	*What is the system doing?*	[41]
B. Lim & A. Dey	*What (else) is the system doing?*	[40]
J. Dodge et al. (this paper)	*Anything about game state.*	

Table 3. The different definitions of "What" from Lim's work in context-aware applications and this paper.

Shoutcasters explained in real time as the players performed their actions, and it takes time to understand the present, predict the future, and link the two. Spending time in these ways reduces the time available for explaining interesting activities happening in present. The corpus showed shoutcasters interrupting themselves and each other as new events transpired, as they tried to keep up with the time constraints. This also has implications to the audience's workflow because the audience must mentally process shoutcasters' departures from present action, even when interesting events unfold continuously.

Even more critical to an automated explanation system, Why questions also tend to require extra effort (cognitive or computing resources), because they require connecting two time slices. For example, to construct the following Why explanation, shoutcasters had to connect past information (scouting the phoenix) with a prediction of the future (investing in spore crawlers):

> Shoutcaster Team 10: *"After seeing the first phoenix [flying unit] and, of course, the second one confirmed, Snute is going to invest in a couple spore crawlers [air defense structure]."*

Answering Why Didn't questions was even rarer than answering Why questions (Table 2). Like Why questions, Why Didn't questions required shoutcasters to make a connection between previous game state and a potential current or future game state. For example,

> Shoutcaster Team 2: *"The probe [worker unit] already left a while ago, so we knew it wasn't going to be a pylon [support structure] rush."*

Why Didn't answers' rarity is consistent with the finding that understanding a Why Didn't explanation requires even more mental effort than a Why explanation [41].

As for an interactive explanation system, supporting Why questions requires solving both a *temporal credit assignment problem* (determining the effect of an action taken at a particular time on the outcome) and a *structural* one (determining the effect of a particular system element on the outcome). See [1] for an accessible explanation of these problems.

Can we approximate Why? The shoutcasters found a potentially "satisficing" approximation of Why, a combination of What and What Could Happen, the two most frequent explanation types. Their What answers explained what the player did, what happened in the game, and description of the game state. These were all things happening in the present, and did not require the additional cognitive steps required to answer Why or Why Didn't, which may have contributed to its high frequency. For example:

> Shoutcaster Team 4: *"And now [marines are going] on to [kill] the SCVs [workers], we need to see some units here for Xy."*

This example illustrates What (marines killing Player Xy's workers) combined with What Could Happen (a possibility that in the future we might "see some units"; i.e., some reinforcements arriving). However, the statement makes no explicit causal linkage between the two events and is thusly not a Why, although a knowledgeable listener might *infer* that the reinforcements are intended to protect the workers. Combinations like this enabled the shoutcasters to predict the future from the *present*, without also requiring them to connect current or future states to information they would have had to recall or re-find from *past* states.

The other two frequent intelligibility types in shoutcasters' commentary, Judgment and How To, also sometimes approximated "Why" information. For Judgment, shoutcasters evaluated an action:

> Shoutcaster Team 1: *"Nice maneuver from [player] Jjakji, he knows he can't fight [player] Neeb front on right now, he needs to go around the edges"*

Casters gave the audience tips and explained high level strategies for How To. For example:

> Shoutcaster Team 10: *"Roach [[ranged unit] and] ravager [long range unit] in general is really good.."*

The next rule-like How To example is an even closer approximation to "Why" information.

> Shoutcaster Team 8: *"Obviously when there are 4 protoss units on the other side of the map, you need to produce more zerglings [inexpensive unit], which means even fewer drones [worker unit] for [player] Iasonu"*

The shoutcasters were giving a rule in this case: given a general game state (protoss units on their side of the map) the player should perform an action (produce zerglings). However, the example does more; it also implies a Why answer to the question "Why isn't Iasonu making more drones?" Since this implied answer simply relates the present to a rule or best practice, it was produced at much lower expense than a true Why answer that required tying past events to the present.

Mechanisms shoutcasters used to circumvent the need for resource-intensive Why explanations, such as using How To, may also be ways to alleviate the same problems in explanation systems. Using satisficing in this way may also benefit explanation consumers, if the *type* of such explanations (e.g., What) aligns with the types the consumer is hoping for. We return to this question in Section 5, where we investigate the types of explanations the game enthusiasts themselves were seeking.

4.2 RQ1.2 Relationships and objects in shoutcasters' explanations

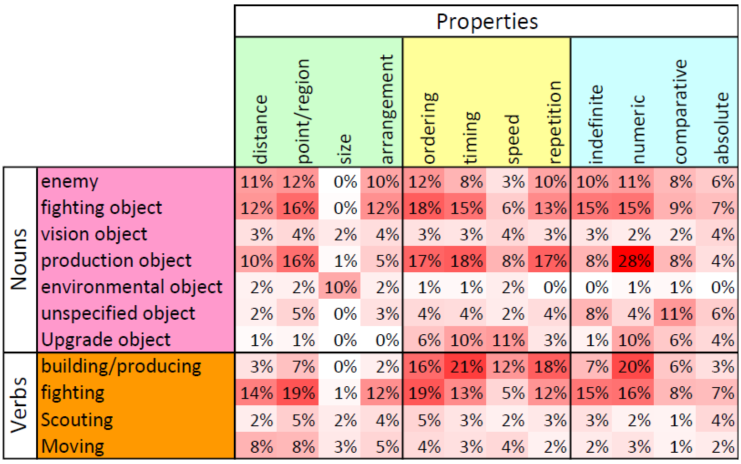
In explaining RTS (real time *strategy*) games, the shoutcasters were themselves strategic in how they put their explanations together. To understand what these experts' construction of explanations might suggest for informing future explanation systems' content, we drew upon a code set from prior work [46], as mentioned in Section 3. We used these codes to investigate the patterns of nouns, verbs, and adjectives/adverbs in these professionally crafted explanations. Online Appendix Table B.15 details the concepts the shoutcasters emphasized the most, and Table 4 shows how they combined these concepts. The shoutcasters then leveraged their noun and verb choices by strategically attaching properties to them (Table 4).

To illustrate, consider a very simple strawman explanation for an action like "Because I have a lot of marines in the enemy base." Implicit in this example are a number of design decisions, such as the granularity of describing the type, number, and location of the objects involved. The analysis presented in this section is intended to inform these kinds of design decisions.

4.2.1 Spatial Properties: "This part of the map is mine!" Shoutcasters often used spatial properties in their explanations to impart strategy. One such property was *distance*. The degree to which an RTS player's strategy is aggressive or passive is often evident in their choice of what distance to keep from their opponent, and the shoutcasters often took this into account in their explanations. One example of this was evaluation of potential new base locations.

> Shoutcaster Team 5: *"If he takes the one [base] that's closer that's near his natural [base], then it's close to Innovation so he can harass."*

Here, the shoutcasters communicated the control of parts of the map by describing each base as a *region*, and then relating two regions with a *distance*. The magnitude of that distance then informed the player's ease in attacking. Of the shoutcasters' utterances that described *distance* along with *production object*, 27 out of 44 showed this pattern.



		Properties											
		distance	point/region	size	arrangement	ordering	timing	speed	repetition	indefinite	numeric	comparative	absolute
Nouns	enemy	11%	12%	0%	10%	12%	8%	3%	10%	10%	11%	8%	6%
	fighting object	12%	16%	0%	12%	18%	15%	6%	13%	15%	15%	9%	7%
	vision object	3%	4%	2%	4%	3%	3%	4%	3%	3%	2%	2%	4%
	production object	10%	16%	1%	5%	17%	18%	8%	17%	8%	28%	8%	4%
	environmental object	2%	2%	10%	2%	1%	1%	2%	0%	0%	1%	1%	0%
	unspecified object	2%	5%	0%	3%	4%	4%	2%	4%	8%	4%	11%	6%
	Upgrade object	1%	1%	0%	0%	6%	10%	11%	3%	1%	10%	6%	4%
Verbs	building/producing	3%	7%	0%	2%	16%	21%	12%	18%	7%	20%	6%	3%
	fighting	14%	19%	1%	12%	19%	13%	5%	12%	15%	16%	8%	7%
	Scouting	2%	5%	2%	4%	5%	3%	2%	3%	3%	2%	1%	4%
	Moving	8%	8%	3%	5%	4%	3%	4%	2%	2%	3%	1%	2%

Table 4. Co-Occurrence Matrix. Across rows: *Nouns* (pink, top rows) and *Verbs* (orange, bottom rows) codes. Across columns: *Spatial properties* (green, left), *Temporal properties* (yellow, center), and *Quantitative properties* (blue, right). Co-occurrence rates were calculated by dividing the intersection of the subcodes by the union.

4.2.2 Temporal Properties: "When should I..." Casters' explanations often reflected players' strategies for allocating limited resources using *speed* properties:

> Shoutcaster Team 4: *"We see a really quick third [resource base] here from XY, like five minutes third [(emphasizing that building a third base only 5 minutes in is very quick)]"*

Since extra bases provide additional resource gathering capacity, the audience could infer that the player intended to follow an "economic" strategy, as those resources could have otherwise been spent on military units or upgrades. This contrasts with the following example,

> Shoutcaster Team 8: *"He's going for very fast lurker den.. [unit production building]"*

The second example indicated the player's intent to follow a different strategy: unlocking stronger units (lurkers).

4.2.3 Quantitative Properties: "Do I care how many?" Shoutcasters very often described quantities without numbers or even units, instead focusing on *comparative* properties (Table 4). For example,

> Shoutcaster Team 1: *"There is too much supply for him to handle. Neeb finalizes the score here after a fantastic game"*

Here, "supply[10]" is so generic, we are not even told the kinds of things Neeb had—only that he had "too much" of it.

In contrast, when the shoutcasters discussed cheap military units, like "marines" and "zerglings," they tended to provide *type* information, but about half of their mentions still included no precise numbers. Perhaps it was a matter of the high cost to get that information: cheap units are often built in large quantities, so deriving a precise quantity is often tedious. Further, adding one weak unit has little impact on army strength, so getting a precise number may not have been worthwhile. Consider the following example, which quantified the army size of both players vaguely, using *indefinite quantity* properties:

> Shoutcaster Team 6: *"That's a lot of marines and marauders [heavy infantry unit] and not enough stalkers [mobile unit]"*

[10] "Supply" is used in an overloaded fashion here, while looking at *military units*, as opposed to the traditional definition—maximum army size the player can have at one time.

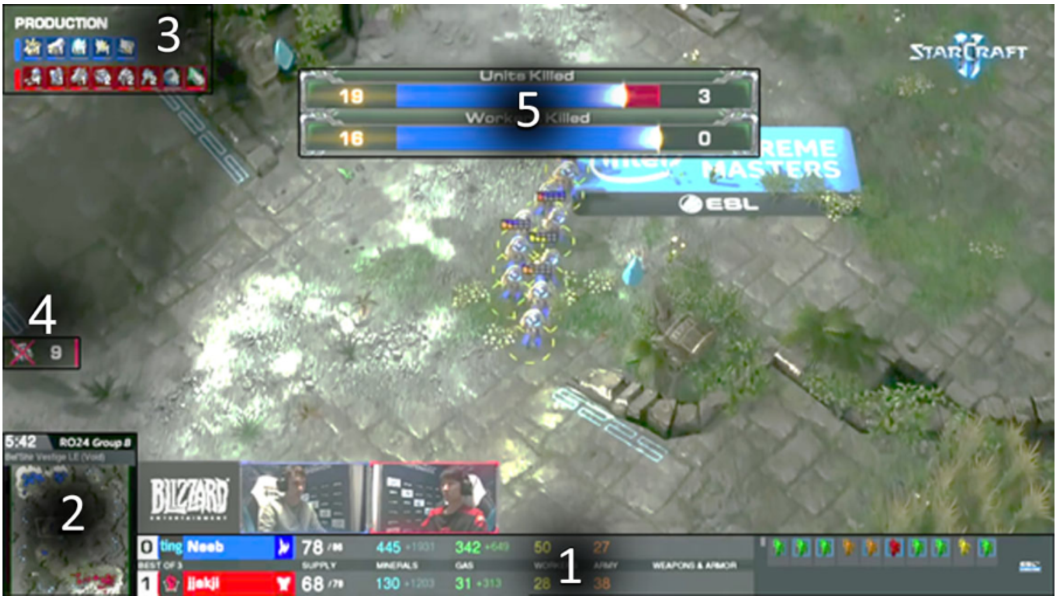


Fig. 4. A screenshot from an analyzed game, modified to highlight the patch types available to casters: *HUD* [1, bottom] (Information about current game state, e.g., resources held, income rate, supply, and upgrade status); *Minimap* [2, lower left] (Zoomed out version of the main window); *"Tab"* [3, top left] (Provides details on demand, currently set on "Production"); *Workers killed* [4, center left] (Shows that 9 Red workers have died recently); *Popup* [5, center] (visualizations that compare player performance, usually shown briefly). Regions 3 and 5 will be detailed in Figures 6 and 7.

Workers are a very important unit in the RTS domain. Consistent with this importance, workers are the only unit where the shoutcasters were automatically alerted to their death (Figure 4, region 4), and are also available at a glance on the HUD (Figure 4, region 1). Correspondingly, the shoutcasters often gave precise quantities of workers (a *production object*). Workers (workers, drones, scvs, and probes) had 46 co-occurrences with *numeric quantities*, but only 12 with *indefinite quantities* (e.g., lot, some, few). To illustrate:

> Shoutcaster Team 2: *"...it really feels like Harstem is doing everything right, and [yet] somehow ended up losing 5 workers"*

4.2.4 Implications for a Future Interactive Explanation System. These results suggest that an effective way to communicate strategy and tactics involves its critical nouns (objects) and actions (verbs) with particular properties, as per the shoutcasters' explanation constructions shown in Table 4. Doing so affords a succinct way to communicate about strategies/tactics, with a potentially lighter load for the audience than with an exhaustive explanation of strategy.

Specifically, spatial properties abstract above the objects on the map to strategies, like when shoutcasters used distance among the objects to point out plans to attack or defend. Temporal properties can be used to explain available strategies for resource allocation. Finally, quantitative properties help ensure alignment in the level of abstraction used by the human and the system. For example, a player can abstract a quantity of units into a single group or think of them as individual units. Knowing the level of abstraction that human players use in different situations can help an interactive explanation system choose the level of abstraction that will meet human expectations.

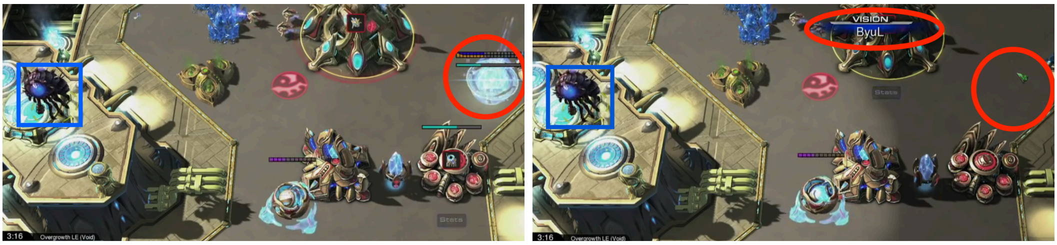


Fig. 5. Shoutcasters have “godlike” powers of *full* observation, shown in the figure on the left. Meanwhile, agents only have *partial* observation shown in the figure on the right. Both figures are from essentially the same time, and the right hand figure shows that the Blue player does not know about the building the Red player has created (highlighted by the red circle). The reason for this is that fog-of-war covers the map for agents — unless the agent controls an object nearby. In this case, the blue player’s unit is highlighted by a blue square in both images, but it is not close enough to see the region highlighted by the red circle.

4.3 RQ1.3 Shoutcasters’ Patches

Where did shoutcasters get the information they used in their explanations? We turn to two frameworks to investigate their information seeking behaviors.

First, Information Foraging Theory (IFT), which has been used in work such as [56], groups patches (areas containing information) into patch types. In the domain studied in [56], patch types were defined as views (sub-windows) in the program used for software development tasks and these patch types consisted of one or many patches. The notion of patches and patch types provides a vocabulary for talking about places in any domain. Originally based on predator-prey models in the wild, basic IFT constructs are the *predator* (information seekers) seeking *prey* (information goals). In the context of XAI, such prey are evidence of the agents’ decision process, which are then consumed to create explanations for agents’ actions. In IFT terms, when deciding where to forage for information, predators (e.g., AI users or assessors) make cost/benefit estimates, weighing the information value per time cost of staying in the current patch (e.g., location on an RTS game map or supplemental information in tab) versus navigating to another patch [57]. Predators, however, are not omniscient: they decide based on their perceptions of the cost and value of the available options. Predators form these perceptions using their prior experience with similar patches [54] and the cues (signposts in their information environment like links and indicators) that point toward various patches. Of course, predators’ perceived values and costs are often inaccurate [55].

Our second framework is the Performance, Environment, Actuators, Sensors (PEAS) model [63]’s common framework for conceptualizing intelligent agents. By considering both IFT and PEAS, we can group the patch types in StarCraft II into the four PEAS categories. By grouping patch types in this way, the PEAS model enables us to consider the kinds of information within these patches at a level of abstraction above that of StarCraft characters and constructs.

We begin with the PEAS patch type most frequently accessed by the shoutcasters. **Actuators** allow agents to interact with their environment, such as a robotic arm moving objects. One way the shoutcasters could observe Actuator patches was via the Production tab (Figure 4, region 3. Expanded in Figure 7), which shows the objects each player is building, namely that Player Blue was building[11] 5 types of objects, whereas Red was building 8. These patches were so important to the shoutcasters that they kept visualizations of actions in progress “always-on.” In fact, prior to the game in our corpus, Team 3 discussed the Production tab’s importance to doing their jobs:

[11] RTS actions have a duration, meaning that when a player took an action, time passed before observed consequences.

Shoutcaster Team 3a: *"What if we took someone who knows literally nothing about StarCraft, teach them a few phrases and* **what everything is on the production tab***?"*
Shoutcaster Team 3b:*"Oh, I would be out of a job."*

Second most sought by the shoutcasters were **Environment** patch types. The environment is where the agent is situated, such as Figure 4's main screen. It also shows the game state, such as the map, structures, and units. Environment patch types (listed in Online Appendix Table B.16) share the trait that they are complete in their information portrayal. For example, the Units tab does not just show *visible* units, but rather *all* units on the map.

From an agent perspective, **Sensors** are how the environment is perceived. In StarCraft II, individual units allow observation of the local area around them (Figure 5). As these units traverse the map, the Minimap patch type (Figure 4, region 2) reveals the regions that have been observed by these sensors. Because shoutcasters have "superpowers" of *full* observation of the environment (whereas players could only *partially* observe it), one might think shoutcasters would have no need for sensors. However, to understand the effectiveness of a player's scouting endeavor, shoutcasters often needed to perform a Vision Toggle (Online Appendix Table B.16) to see only what each player could see on the Minimap and on the main window, as shown in Figure 5.

What is interesting about **Performance** patch types—which measure an agent's assets, resources, successes, and failures—is how *un*interested shoutcasters were in them. An example is in Figure 4, in which shoutcasters can access information in regions 4 & 5 to quickly compare the players' losses. However, the shoutcasters rarely accessed performance patch types—only about once per game (detailed in Online Appendix Table B.16).

4.3.1 Implications for a Future Interactive Explanation System. Abstracting beyond the StarCraft components to the PEAS model revealed a pattern of the shoutcasters' behaviors, which we characterize as: "keep your Sensors close, but your Actuators closer." This aligns with other research, showing that real-time visualization of *agent actions* can improve system transparency [76]. However, these results contrast with the explanation systems that tend to prioritize Performance measures. Our results instead suggest that an explanation system should prioritize useful, readily accessible information about agents' actions and observations (Actuator and Sensor patch types).

4.4 RQ1.4 How shoutcasters forage between patches

To navigate among these patches to get to the right information at the right time, the shoutcasters' foraging seemed to follow a common "loop" (illustrated in Online Appendix Figure B.14). The shoutcasters tended to start (and end) at the "always-on" Actuator-related patch types of current state's actions in-progress; when something triggered a change in their focus, such as impending combat, they checked the Environment for current game state. Once the object of their excitement resolved, the shoutcasters either returned to the Actuators or transitioned to Performance patch types for evaluative purposes. If they needed more information about what a player saw that the Actuator patch types did not provide, they shifted their focus to the Sensors to see through a player's eyes.

Information Foraging Theory (IFT) explains why information predators leave one patch to move to another, such as when the shoutcasters left Actuator patch types. According to IFT, predators choose navigations as cost/benefit decisions, based on the value of information in the patch a predator is already in versus the value per cost of going to another patch [57]. Staying in the same patch is generally the least expensive, but when there is less value in the current patch than the *expected value* the predator perceives in another, they move to that other patch. They form the perception of this value from both their prior experience with patches [54] and from the cues (signposts in their information environment) that point toward content available in other patches.

IFT's scent construct is the predator's attempt to maximize the expected value per expected cost of making a navigation, by analyzing these cues. We saw shoutcasters "sniff out" cues in the following example where shoutcasters discussed unit production, then one of the shoutcasters explained why the next navigation was made:

> Shoutcaster Team 1: *"But that robo [unit production structure] after this warp prism [transport unit] really does need to get back into pumping out these powerful colossi [massive siege unit] or disruptors [long range unit]. I see a couple units on the bottom left hand side as well, probably some adepts [mobile unit]."*

Seeing those units prompted a navigation to the minimap patch type in order to look at information there. The shoutcaster had some expectation of value, which was learning about the units, and an expectation of the cost, which was the click on the minimap.

The predator's perception of value is relative to the predator's information goals, and works in reactive IFT have shown that goals often change in reaction to changing environments and changing contexts [12, 39, 53], often in response to events [8, 18]. Thus, in reactive foraging, signs of events are themselves scent-bearing cues: they "provide users with concise information about content that is not immediately available" [57]. In our study, predators following such event-based cues was particularly apparent with signs of impending combat.

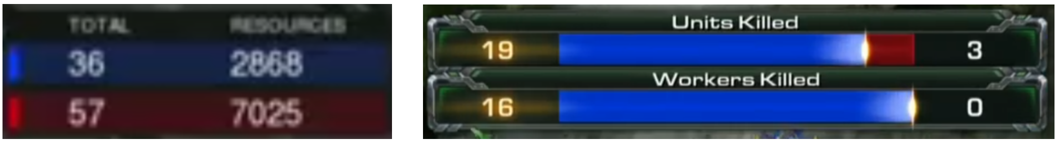
Impending combat, indicated by the co-location of opposing units (e.g., in Online Appendix Figure B.14), was the most common event-based cue triggering a move from the Actuators type (Production tab) to the Environment type (Units tab). This combat cue led to the shoutcaster navigating to a new patch to inform themselves of the environment. In fact, combat cues triggered navigations to the Units tab most frequently, accounting for 59% of the navigations (51 navigations[12]).

Combat ending was another event-based cue that triggered a navigation to a Performance patch type. 10 of the 13 navigations to a past-facing Performance patch type occurred shortly after combat ended to evaluate the action. However, shoutcasters infrequently visited other Performance patch types, such as Income, Units Lost, and Army tabs, in an attempt to reason why a player had accrued an in-game lead or how big that lead was (7 navigations).

When shoutcasters detoured from this A-E-P loop to a Sensor patch type, it was usually to enrich the information environment via the Vision Toggle (36 navigations). The cue(s) that led to this navigation were not obvious in the data, but assessing a scouting operation was the common theme. Recall that shoutcasters had "godlike" power of map vision, but the Vision Toggle enrichment operation allowed shoutcasters to see the game through the eyes of either player (Figure 5). Toggling to this filtered view of the world allowed the shoutcasters to assess what information was gathered by each player via scouting actions (29 of the 36 total Vision Toggles). This provided shoutcasters with insight on actions which might be a surprise for the opposition.

Besides following cues, IFT has another foraging operation: *enriching* their information environment to make it more valuable or cost-efficient [57]. Filtering through the Vision Toggle was one example of this, and another was when shoutcasters added on information visualizations derived from the raw data, like Performance measure pop-ups or other basic visualizations. Figure 6 shows two examples of enrichments available to shoutcasters. These Performance measures gave them information about the ways one player was winning at a glance. The most commonly accessed Performance measures showed the number of units lost and their total value, in terms of resources spent (e.g., the Units Lost tab, shown in Figure 6). This measure achieves "at a glance" by aggregating *all* the data samples together by taking a *sum*; derived values like this allow the visualization to scale to large data sets [64]. However, the lower data aggregation patch types were more heavily accessed. The shoutcasters accessed the Production tab to see units grouped by type, as Figure 7

[12] The data underpinning the summary that follows is detailed in Online Appendix Table B.16.



TOTAL	RESOURCES
36	2868
57	7025

Fig. 6. The Units Lost tab (left image) shows the number of units lost and their total value, in terms of resources spent, for both players. In this example from Shoutcaster Team 2, Blue Player (top) has lost merely 2800 minerals worth of units so far in this game, while Red has lost more than 7000. The Units Killed popup (right image) allows shoutcasters to quickly compare player performance via a "tug-of-war" visualization. In this example from Shoutcaster Team 1, Blue Player (left) has killed 19 units, while Red has killed merely 3. The main difference between these two styles of visualization is that the tab offers more options and information depth to "drill down" into.

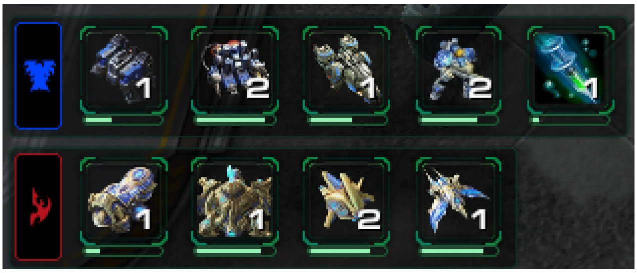


Fig. 7. The Production tab, showing the build actions currently in progress for each player. Each unit/structure type is represented by a glyph (which serves as a link to navigate to that object), provided a progress bar for duration, and given the number of objects of that type. Thus, we can see that Blue Player (top row) is building 5 different types of things, while Red (bottom row) is building 4 types of things. The Structures, Upgrades, and Units tab look fairly similar to the Production tab.

shows, so *type* information was maintained with only *positional* data lost. This contrasts with the Minimap (medium aggregation), in which type information is discarded but positional information maintained at a lower *granularity*. Shoutcasters sought Performance measure patch types primarily to understand present state data (HUD), but these patch types were also the only way to access *past* state information.

4.4.1 Implications for a Future Interactive Explanation System. These results have several implications for automated explanation systems in this domain. First, the A-E-P+S loop and how the shoutcasters traversed it reveals priority and timing implications for automated explanation systems. For example, the cues that led the shoutcasters to switch to different information patches could also be cues in an automated system about the need to avail different information at appropriate times. For example, shoutcasters showed a strong preference for actuator information as "steady state" visualization, but preferred performance information upon conclusion of a subtask.

Viewing the shoutcasters' behaviors through the dual lens of PEAS + IFT has implications for not only the kinds of patches that an explanation system would need to provide, but also the cost to users of not making these patches readily accessible. For example, PEAS + IFT revealed a costly foraging problem for shoutcasters due to inaccessibility of some Actuator patch types. Specifically, there was no accessible mechanism by which the shoutcasters could navigate to an Actuator patch with fighting or scouting actions in progress. Instead, the only way the shoutcasters could find these actions was via painstaking camera placement. The shoutcasters moved the camera countless times using the Minimap, traditional scrolling, or via tabs with links to units or buildings. Yet, despite all these navigation affordances, sometimes the shoutcasters were unable to place the camera on all

Intelligibility type	Pair 1	Pair 2	Pair 3	Pair 4	Pair 5	Pair 6	Pair 7	Pair 8	Pair 9	Pair 10	Questions	"Answers"
`What`	2	3	41	1	6	14	10	1	8	62	148 (73%)	595 (44%)
`What Could Happen`		1	1		3	1	1		2	7	16 (8%)	376 (28%)
`Why Did`			2			3	1			8	14 (7%)	27 (2%)
`How To`	1		3							5	9 (4%)	233 (17%)
`Judgment`			3		3					2	8 (4%)	112 (8%)
`Why Didn't`	1		1							5	7 (3%)	6 (<1%)
Total	4	4	51	1	12	18	12	1	10	89	202	1349

Table 5. Lim & Dey questions participants asked each other, by participant pair, using the same code set as Table 2. Question column gives count and frequency of total questions from the Game Enthusiast Study, whereas the Answer column considers all utterances from the Shoutcaster Study (total exceeds 1024 because of multiple coding). Note how often `What` questions were asked, both by the population of participants as a whole, and by a few individual pairs, where it was particularly prevalent.

the actions they needed to see. For example, while Shoutcaster Team 4 had the camera on a fight at Xy's base, a second fight broke out at DeMuslim's base, which they completely missed:

Shoutcaster Team 4a: *"Xy actually killed the 3rd base of DeMuslim."*
<surprised and noticing something amiss, the team tries to figure what happened>...
Team 4b: *"Oh my god, you're right."*
Team 4a: *"Yeah, it was killed during all that action."*

But was overcoming these foraging difficulties important? The data here relate to difficulties expert explainers faced while attempting to *supply* explanations they thought their audience would value. We will return to this issue from an explanation *demand* perspective in Section 5.2.

5 RESULTS: THE GAME ENTHUSIASTS

Our second study, in which we tasked experienced StarCraft players with discovering key decisions leading to a game's outcome, was designed to answer **RQ2**, which we expand into 3 parts: Questioning Behavior, Foraging among many paths, and Decisions & Distractions.

5.1 RQ2.1 Questions game enthusiasts ask when demanding information

To situate our investigation in the literature of humans trying to understand AI, we coded questions participants asked using the same Lim & Dey intelligibility types [40] that we used to code shoutcaster explanations. Table 5 presents the results, broken down by participant pairs.

We have already mentioned that prior research has reported Why questions to be much in demand [40, 41]. Yet, our participants, like the shoutcasters, rarely asked them—10% of participants' questions fell into the Why and Why didn't categories combined. Also similar to the shoutcasters, participants greatly preferred What questions, accounting for over 70% of their questions (Table 5).

In comparing the participants' questions with the shoutcasters' "answers" (Table 5), two points stand out: the popularity of What intelligibility type information by both the shoutcasters and the participants, and the relatively low interest in the Why and Why Didn't intelligibility types by both the shoutcasters and the participants.

One difference between the game enthusiast study participants and the shoutcasters is that the participants showed relatively little interest in any of the intelligibility types beyond What, but the shoutcasters imparted What Could Happen & How To information in relatively high frequencies. This difference could relate in part to both the size of the shoutcasters' vs. the participants' audiences and to the differences in goals in these two populations. Shoutcasters explain to a broad, unseen, and potentially well-informed audience. Their jobs depend upon meeting that audience's needs, so

Types of What questions	**Freq**
Identification of noun/verb: Question about what an object in the current game state game is, or what action is taking place, e.g.: -Pair9-P17: "*Is than an Overseer there?*"	59 (43%)
Quantification: Question about quantity of object in the current game state, e.g.: -Pair10-P19: "*Wait, second Gateway or first Gateway?*"	25 (18%)
Temporal: Question about prior game state, e.g.: -Pair6-P12: "*When did he get zerglings?*"	24 (17%)
Resolving Confusion: Question to clarify current game state, e.g.: -Pair3-P5: "*What's going on over here?*"	19 (14%)
Where: Question about location of object in current game state, e.g.: -Pair10-P20: "*Where did that probe go?*"	7 (5%)
Information Availability: Question about what game state information is available to player, e.g.: -Pair3-P5: "*Aren't they seeing each other?*"	4 (3%)

Table 6. Types of What questions participants asked during the main sessions.

we can assume that most of their information is reasonably appreciated. This suggests that their What Could Happen & How To information are of value to their audience—value that may not be easily obtained from someone with less expertise than these professional explainers. Because these types seem to add value that may not be easily obtainable without the professional explainers, these kinds of explanations may also build user confidence in the explainer, a possibility that may also be leveraged by future interactive explanation systems. The participants in our game enthusiasts study, on the other hand, had audiences of one person (their partner). It seems likely that they tailored their questions around this evaluation of the agent's play and to their expectations of questions their partner could understand and answer.

5.1.1 The Many Flavors of "What" Prey.

Why the differences from prior research results?

One hypothesis from the participants' interest in What explanations could be that, in this kind of situation, participants' prey was simply "play-by-play" information. However, this hypothesis is not well supported by the data. Although participants did seek *some* play-by-play information (Pair3-P5: "*...so he just killed a scout, right?*"), several prey patterns in their What questions went beyond play-by-play (Table 6). Three of these patterns accounted for about one-third of the What questions.

The "Drill-Down What" of Current State. One common question type participants asked when pursuing prey involved "drilling down" to find the desired information. These often came in the form of identifying an object/action in the current game state (43%) or quantifying an object (18%).

Navigating in pursuit of this kind of prey was often costly to the participants. The least expensive way was navigating via a drop-down menu (2 clicks) in region 2 of Figure 1, but participants instead often foraged in other ways. To find the answer to a question (like Pair3-P6's earlier) participants sometimes navigated to several unit producing structures on the map, into a structure, and then on to the next. For example, the following question required participants to drill down into several structures on the map to answer it: Pair3-P6: "*Is the human building any new stuff now?*" Pair 3 made seven navigations to answer their question about "building new stuff."

Shoutcasters' commentary tended to meet the participants' interest in these questions: 56% of shoutcasters' What comments answered questions about identifying an object/action and 28% covered the quantity of an object. As an example of the match to shoutcasters' comments, question about identifying units: Pair6-P12: "*I think, well, we have a varied composition, besides roaches and what are these?*" would be well-matched to shoutcaster explanations such as this one: Shoutcaster Team 3: "*We have the ravagers now coming up..*" Questions from participants about the quantity of an object, such as Pair7-P14 asking "*Does he have any zealots or stalkers?*" could be answered by shoutcaster explanation like the following:

Shoutcaster Team 2: *"Couple of stalkers, I'm not even sure if blink [an upgrade] is done yet."*

That the shoutcasters' "supply" of explanations met participants' "demand" for their two most popular types of What questions provides support for the possibility of using shoutcasters' explanations as a possible content model for future explanation systems.

The "Temporal What" of Past States. A second common prey pattern, exhibited by 4 of the 10 participant pairs, was asking What questions to fill in gaps regarding past states. "Temporal What" accounted for 15 instances (about 10%) of their Whats[13]. For example:

Pair3-P6: "*When did he start building [a] robotics facility [unit producing structure]]?*"

As to answering such questions, shoutcasters did sometimes answer temporal questions, but at a much lower rate (3% overall) than participants asked them. For example:

Shoutcaster Team 2: *"The probe already left a while ago..."*

This difference may have been affected by two factors. First, shoutcasters had to provide commentary in "real time," and could not easily afford to go back in time to check on details of past states. Participants, on the other hand, sometimes did choose to go back in time to fill in temporal knowledge gaps, especially . However, (the second factor), one reason participants did so was an artifact of the experiment set-up: some participants needed to remind themselves of a timestamp for a decision point they had identified, which we detail in Section 5.2.

The "Higher-Level What". The next most common prey pattern was at a higher level of abstraction than the specific units or events, aiming instead toward more general understanding of what was going on in the game. These What questions arose in 14% of the instances of What questions. For example, Pair10-P20 asked, "What's going on over there?" in which "there" referred to a location on the map with military units that could have been gearing up for combat. We did not count the number of shoutcaster comments that answered this question because we could not narrow them down in this way. That is, although many of their comments could be said to be applicable to this type of question, the same comments were also applicable to more specific questions. For example:

Shoutcaster Team 7: *"This is about to get crazy because [of] this drop coming into the main base [and] the banelings [suicide bomb unit] trying to get some connections in the middle."*

Shoutcaster Team 9: *"I like Elazer's position; he's bringing in other units in from the back as well."*

5.1.2 Questioning the Unexpected. Lim & Dey reported that when a system behaved in unexpected ways, participants' demand to know Why increased [40]. Consistent with this, when participants saw what they expected to see, they did not ask Why or Why didn't questions. In cases of the unexpected, however, a What prey pattern arose, in which participants questioned the phenomena before them. We counted nine What questions of this type:

Pair9-P17: "*...interesting that it's not even using those.*"
Pair10-P19: "*I don't get it, is he expanding?*"
Pair10-P19: "*Wow, what is happening? This is a weird little dance we're doing.*"
Pair10-P20: "*<when tracking military units> What the hell was that?*"

The unexpected also produced Why questions. About half of the participants' Why and Why Didn't questions came from seeing something they had not expected or not seeing something they had expected. For example:

[13] By the time of his thesis [42], Lim had split a "When" intelligibility type off into its own type, but it was originally part of the "What" intelligibility type.

> Pair1-P1: "*<noticing a large group of units sitting in a corner> Why didn't they send the big army they had?*"
>
> Pair10-P19: "*Oh, look at all these Overlords [support unit]. Why do you need so many?*"

5.1.3 Implications for a Future Interactive Explanation System. The participants' types of questions, viewed together with the shoutcasters' commentary implied that future explanation systems might shoutcasters as possible "gold standards" to inform designs in this domain. For example, the high rate of What questions from participants matched reasonably well with a high rate of What answers from shoutcasters. Drawing explanation system design ideas from these expert explainers may help inform the needed triggers and content of the system's What explanations.

The dominance of What questions also points to participants' prioritizing of state information in this domain. The noun/verb Whats were about objects or actions in a state they did not know about, quantification Whats were about identifying quantities of game objects in order to gain details about the state, temporal Whats were about past states they either hadn't seen or had forgotten and higher-level Whats were about understanding the purpose of a current or emerging state. Further, shoutcasters exceeded the participants' rate of Whats in the two most popular categories with their explanations. This suggests that, in the RTS domain, an explanation system's most sought-after explanations may be its explanations relating to identification of nouns, verbs, or quantities. This claim is strengthened by the similar distribution of intelligibility in Haynes et al.'s simulated air combat domain [23], which has similarities to the RTS domain.

As noted in prior research, unexpected behaviors (or omissions of expected behaviors) led to increases in questions for both the What and the Why intelligibility types [40]. If an explanation system can recognize unexpected behavior, it could then better predict when users will want Why and What explanations to understand the differences in expected and actual behavior. One way to accomplish this would be to compare agent behavior against standard "build orders" that human players follow and look for deviations.

These results suggest that an interactive explanation should, like the shoutcasters, prioritize availing What, What Could Happen & How To information. Further, the lack of interest by both the shoutcasters and the participants in the other intelligibility types suggests that those types should not be the highest priority for creators of interactive explanation systems for the RTS domain.

5.2 RQ2.2 How game enthusiasts forage among many evolving paths

As an information environment, RTS games have foraging characteristics that set them apart from other information environments previously studied from an IFT perspective, such as web sites [57] and programming IDEs [55]. These previously studied domains are relatively static, with most changes occurring over longer periods. In contrast, an RTS information environment changes rapidly and continually, driven by actions that do not originate from the foragers themselves. This difference in the information environments produces new cognitive costs for foragers. One such difference is that our participants had to spend some of their cognitive effort simply monitoring the overall game state, looking for suitable cues to appear that they would need to investigate further.

Another cognitive difference arose in the structure of paths a forager might follow. In an RTS information environment, the number of paths increases as the the complexity of the game state increases, but the path lengths tend to be shorter than the lengths of foraging paths in other kinds of information environments. This difference is conceptualized in Figure 8. This means that most questions are answered within a few navigations. In foraging environments like IDEs, however, there might only be a few interesting links from any single information patch, but some can lead to lengthy sequences of navigations [55].

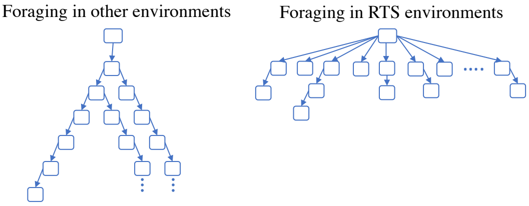


Fig. 8. Conceptual drawing of foraging in the RTS domain vs. previously studied foraging, intended to illustrate 2 different ways foraging can suffer from a "scaling-up problem." **Left:** In prior IFT literature about foraging in an IDE (integrated software development environments), predators have a few paths to consider at a time, but the paths are sometimes very deep, which can lead to the *depth* version of the "Endless paths" foraging problem [[55] Fig. 5]. **Right:** When foraging in an RTS environment like StarCraft, most navigation paths are shallow, but with numerous paths to choose from at the top level, leading to a *breadth* version we term the "Many paths" foraging problem.

Reasons for participants' path choices	**Freq**
Monitoring State: Continuous game state monitoring, such as watching a fight. -Pair4-P7: "*I wanted to see how the fight was going.*"	65 (46%)
New Event: Attending to a new event for which participant wished to satisfy curiosity about. -Pair2-P4: "*I saw there was a new building.*"	36 (25%)
Update Game State: Updating potentially stale game information that the participant explicitly stated prior knowledge about. -Pair1-P2: "*I was mainly looking at army composition, seeing how it had changed from the last fight.*"	29 (21%)
Obsolete Domain: Explicitly using domain info that may not be current, such as game rules (e.g., what buildings can produce). -Pair3-P6: "*I mainly clicked on the adept because I'm more familiar with [previous StarCraft version].*"	11 (8%)

Table 7. Why the participants sought the information they sought: Reasons, examples and frequencies.

5.2.1 Foraging in the RTS Domain. At first, there was hardly any difference between RTS foraging and other environments. During the early stages of a game, there are few units, buildings, or explored regions, so foraging is relatively straightforward. As one participant put it:

Pair7-P14: "*There is only so many places to click on at this point.*"

As long as this remained the case, each relevant path could potentially be carefully pursued, similarly to an IDE. Four participant pairs (Pair 2, Pair 4, Pair 7, Pair 9) paused the replay for an average of 90 seconds within the first 1:30. That early in the game, the environment was sparsely populated, but they closely studied individual objects and actions. In contrast, later in the game, when 50 of the same unit existed, they received much less attention than when there was just one.

Choosing among many available paths created cognitive challenges for participants. They needed to keep track of an increasing amount of information as the match progressed. Each time a player performed an action, it added information. If participants foraged for it, we coded their navigation as a New Event, which accounted for 26% of interviewed navigations (Table 7). For example:

> Pair10-P19: "*...noticed movement in the Mini-map and that the Zerg troops were mobilizing in some fashion. So I guess I just preemptively clicked...*"

The rate of path creation exacerbated the "many paths" problem, illustrated in Figure 8. Professional StarCraft players regularly exceed several hundred actions per minute (APM) [75]. This meant that players performed rapid actions that changed the game state. Each of these actions not only potentially created new paths; they potentially updated the existing ones. This caused the knowledge the participants had about paths not recently checked to become stale, which in turn led to a prevalence of re-foraging to see updates (21% in Table 7) or monitor state (46% in Table 7).

	Task Time	RTR	Rewind	Timestamp Rewind	Context Notes
Pair 1	20:48	1.3	3	1	Rewatched 1 fight
Pair 2	20:40	1.3			Extensive pause around 1:00 to evaluate game state
Pair 3	55:08	3.4	12	9	Rewatched fights and fight setup. Slowed down replay during 1 combat.
Pair 4	32:16	2.0	2		Rewatched opening build sequence and evaluated information available to the agent at a key moment. Many pauses to explain game state.
Pair 5	24:23	1.5	5		Rewatched unit positioning, AI reaction to events, and scouting effectiveness.
Pair 6	31:56	2.0	4	2	Rewatched 2 fights.
Pair 7	29:49	1.9			Made no use of time controls other than pausing to write down decision points.
Pair 8	21:27	1.3			Made no use of time controls other than pausing to write down decision points.
Pair 9	39:17	2.4	2	1	Rewatched 1 fight. Slowed down replay for whole task.
Pair 10	61:14	3.8	Lots	Some	Rewound extensively, in a nested fashion. Changed replay speed many times.

Table 8. Participant task time (33:42±14:18 minutes) and time control usage information. Note that the replay file was just over 16:04, so dividing each pair's time by 16 yields the third column, "*Real-Time Ratio* (RTR)" with Mean±SD=(2.1±0.89). Some of times participants rewound the replay were because we requested timestamps for events, shown in the fourth column, "*Timestamp Rewinds*." The last column provides any additional context in which replay and pause controls were used.

Since each event and its corresponding cues were visible for a limited time only, paths not chosen promptly by participants quickly disappeared. Further, paths are numerous and frequently updated. Thus, there is a large risk for paths of inquiry to be forgotten or unnoticed as the game proceeds, as in these examples:

Pair7-P14: "*Oh my gosh, I didn't even notice he was making an ultralisk den.*"

Pair3-P6: "*I didn't notice they canceled the assimilator.*"

5.2.2 Many Rapidly Updating Paths: Coping Mechanisms. Participants responded to this issue in several ways. First, some participants chose a path and stuck to it, ignoring the others. For example, Pairs 2, 7, and 8 (Table 8) analyzed the replay using not much more time than shoutcasters spend. Achieving this speed of analysis, however, required participants to ignore many game events.

For example, when asked about desire to click anywhere else, one participant volunteered:

Pair10-P19: "*Mmm, if I had multiple, like, different screens yeah. But no, that seemed to be where the action was gonna be.*"

In this fashion, participants chose to triage game events based on some priority order. In both of the following examples, the participants navigated away from the conclusion of a fight:

Pair6-P11: "*I wanted to check on his production that one time because he just lost most of his army and he still had some [enemies] to deal with.*"

Pair3-P5: "*I was trying to see what units they were building, after the fight, see if they were replenishing, or getting ready for another fight.*"

The second method participants used to manage the complexity of paths was to use the time controls to slow down, stop, or rewind the replay. For example, Pairs 3 and 10 rewound the most often (Table 8) and paid higher navigation costs to do so, but they viewed these navigations as worthwhile to providing necessary information:

		Shoutcasters		Participants	
	Patch Name	**Use Count**	**Frequency**	**Use Count**	**Frequency**
Performance	Units Lost/Killed	12	11%	4	6%
	No tab	N/A	N/A	4	6%
	Other performance patches	6	5%	8	13%
Environment	Units	51	14%	13	21%
	Upgrades	5	4%	4	6%
	Structures	2	2%	3	5%
Actuator	Production	Always on		11	18%
Sensor	Vision Toggle	36	32%	15	24%
	Minimap	Too many to count		Too many to count	
	Totals	112	100%	62	100%

Table 9. Patch type usage comparison summary. For the aggregation of performance patch types Shoutcasters: 2 patch types. Participants: 4 patch types (each < 5%). The most popular patch types for both groups are highlighted in grey.

> Pair6-P11: "*I looped back to the beginning of the final fight ... to see if there was anything significant that we had missed the first time around.*"

The cost of doing so was more than just time, however, because the more paths they monitored, the greater the cognitive load:

> Pair10-P19: "*There's just so much happening all at once; I can't keep track of all of it!*"

5.2.3 Patches used in Foraging: Does supply match demand? From an IFT perspective, the shoutcasters and participants in our game enthusiasts study foraged very similarly. The participants favored the same four patch types as the shoutcasters: the Production Tab, Minimap, Units (current), and the Vision Toggle (Table 9). These popular items tie to the same three PEAS types that shoutcasters had emphasized (Actuator, Environment, and Sensor), with only an occasional visit to the Performance Measures—around once per game by both shoutcasters and participants.

Shoutcasters and Participants cared about similar things: To directly compare shoutcasters and participants in *this* specific setting, we analyzed an additional video—the game from the game enthusiasts study[14]. We will refer to the shoutcasters in this game as Shoutcaster Team 11.

At times, Shoutcaster Team 11 and the participants made very similar observations, showing that they cared about the same things. For example, when participants noticed one player lagging behind on upgrades and unlocked units ("tech"):

> Pair5-P10: "*His [the "AI"] army is underteched and underupgraded.*"
>
> Pair9-P17: "*So again, I'm surprised they [the "AI"] haven't upgraded at all yet.*"

Shoutcaster Team 11 observed the same issue that participants did—that the "AI" had only lower quality units available and was behind on upgrades:

> Shoutcaster Team 11: *"He's [The "Human"] a tier 3 protoss, so to speak, with the templar archives [unit producing structure] right now, with charge [an upgrade] then into blink [an upgrade] and archons [strong units that require many "tech" preconditions], whereas zerg [The "AI"] is stuck on lair [tier 2] tech units."*
>
> Shoutcaster Team 11: *"And keep in mind, Byul [The "AI"] has no upgrades..."*

[14] As noted before, the replay file is available at http://lotv.spawningtool.com/23979/. There is a shoutcaster video of that same game available at: https://sc2casts.com/cast20687-Stats-vs-ByuL-Best-of-5-2016-IEM-Gyeonggi-Semi-Finals. It is game 3 in that link, cast by Rotterdam and ToD.

Shoutcasters vs. participants on where the agent was looking. Scouting activity reflects where the agent was looking, and the top of Figure 10 shows that the participants and Shoutcaster Team 11 both cared about scouting, although not always at the same times. In the first few minutes of the game, Shoutcaster Team 11 was more thorough, but many of the participants identified scouting activities at about the same time, and continued to show an interest until around 5:00, when fighting began. (Recall Figure 10.)

However, as discussed further in Section 5.3.2, participants' attention was drawn away from scouting when fighting began. In contrast, of Shoutcaster Team 11's 18 total utterances in this game pertaining to scouting, most occurred after 5:20 (*after* participants had generally stopped pointing out scouting actions). For example, at around 12:00, the shoutcasters emphasized the importance of scouting by pointing out how "scary" StarCraft can be in the absence of scouting:

> Shoutcaster Team 11: *"If you would take a look at the vision of the protoss now, you see nothing, Todd, you see absolutely nothing. ...it's nothing but darkness everywhere, and it's just scary to play StarCraft like that."*

Still, even though participants did not often mark late game scouting actions "key", sometimes they showed continuing interest through their conversation about what the agent could see:

> Pair3-P5: "*I mean they saw all the warp gates [unit producing structures] right?*"
> Pair9-P18: "*I'm gonna lock to the CPU's view as it gets that first scouting.*"

Shoutcasters and Participants used Actuator information quite differently. Shoutcasters treated the Production tab (shown in Figure 7) as an "always-on" visualization, and this usage was consistent across all observed shoutcasters. However, only 3 participant pairs (Pairs 1, 4, 7) followed this strategy, with the rest choosing primarily to leave whatever they were last looking at showing. Some chose to show a combination of Units and Production tabs, but Pair 10 made the unique choice of hiding the tab when they were not looking for something specific. The Production tab was the most frequently viewed, and some participants stated a strong preference for this info, e.g.:

> Pair6-P12: "*Is there a way to see what he's building behind the fight? Trying to produce units, doesn't seem like he is...*"
> Pair7-P14: "*I love the Production tab... makes it a lot easier to find build orders and stuff*"

5.2.4 Implications for a Future Interactive Explanation System. As the sections to this point have shown, the participants' costs to navigate to some of the prey became expensive. Inordinate expenses arose not only in the form of how many navigation actions the participants needed to take, but also in costs to human cognition.

The most important patch types were arguably Actuators and Sensors. As Section 3.1 showed, shoutcasters placed great value on these patch types. Further, shoutcasters have extensive experience foraging in the StarCraft domain, and this showed in their their foraging patterns. For example, the shoutcasters knew to make the Production tab an "always-on" visualization and did not need foraging cues to point their foraging in that direction. However, participants did not seem to have this knowledge. In fact, some participants' behavior hid this tab after completing a subtask, possibly to maximize screen real estate. The participants' failure to make good use of the Production tab suggests a need for better cues and foraging mechanisms in the environment to direct them there.

The participants also incurred high cognitive costs in the form of triage. Assessing an agent required participants to constantly choose among a great many possible paths—necessitating not just triage, but *rapid* triage. This kind of foraging is in stark contrast to recent IFT reports from the software engineering domain, e.g. "miles of methods" [55] (recall Figure 8). Foraging combined with triage has barely been considered in IFT literature (with the exception of [59]), and the added complication of paths soon "expiring" has not been considered at all in IFT research. How to address

Code	**Total**	Pair 1	Pair 2	Pair 3	Pair 4	Pair 5	Pair 6	Pair 7	Pair 8	Pair 9	Pair 10
Building-Expansion	52	7	7	8	8	-	6	3	-	7	6
Building - Rest	69	6	4	7	15	1	7	11	2	12	4
Building - All	114	13	11	15	21	1	12	12	2	18	9
Fighting - All	98	8	4	11	8	4	10	6	8	11	28
Moving - All	26	3	1	5	1	2	2	2	2	3	5
Scouting - All	23	1	2	1	5	1	1	3	-	3	6
Total	228	34	20	40	45	11	31	39	16	42	65

Table 10. Summary of decision points identified by participants. Sums may exceed totals, since each decision point could have multiple labels. Note how prevalent *Expansion* was within the *Building* category. (Note: Codes can co-occur, so the sum of *Expansion* and the rest of the *Building* codes exceed 114.)

this problem is thus an open question. However, one possibility for the RTS domain could be a recommender system that is lightweight enough and fast enough to keep abreast of the constantly changing availability of paths, to help people triage which path to follow next.

Even when a participant was following a "good" path, it was not infrequent for a more important path to suddenly appear, such as a critical battle, meaning that one of the paths had to be ignored to follow the other. When this happened, participants often forgot about or otherwise interrupted their paths of inquiry. In the slower-paced domain of spreadsheet debugging, participants faced with branching paths with multiple desirable directions became more effective when the environment supported a strategy they called "to-do listing" [19]. To-do listing was supported on its own or in composition with other problem-solving approaches, so it could also act as a strategy enhancer. Perhaps a similar strategy in the RTS domain could enable participants to carry on with their current path uninterrupted—but also keep track of the critical battle to attend to if/when the battle becomes more urgent than the current task.

5.3 RQ2.3 Critical decision points and how they're missed

Explaining an RTS game to a human may become impossible for a system to do if the human has missed the most critical points in the game's action. But keeping up on the critical points can be challenging for humans in the RTS domain, because players and intelligent agents make thousands of sequential decisions[15]. Indeed, the previous section has just illustrated one aspect this challenge in the costs our participants bore of trying to triage which information to pursue while at the same time actually pursuing other information.

Another aspect of this challenge could lie in participants' abilities to choose which cues will be the most important. To investigate this aspect, we asked participants to write down what they thought were the important game events, which we termed "key decision points". We defined the term key decision points to participants as "an event which is critically important to the outcome of the game," to give participants leeway to apply their own meaning. Since all participant pairs were examining the same replay file, we were then able to compare the decision points the different participants selected. That is, the cues in the information environment were the same for all the participants—whether they noticed them or not. Key decision points fell into four main categories: building/producing, fighting, moving, and scouting. Fighting and building comprised 85% of the 228 total decision points participants identified (Table 10).

[15] Although some literature describes AI approaches that aim to increase transparency (e.g. revealing instances of its decision-making neurons [77, 78]), there is a paucity of literature that investigates humans trying to understand AI decisions in such a setting. An exception is McGregor et al. [45].

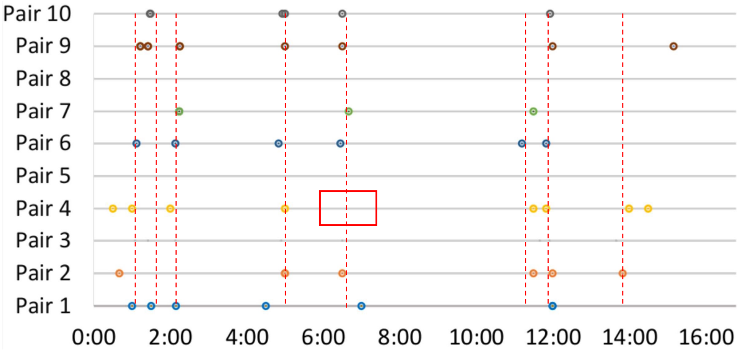


Fig. 9. Dots show the *Building-Expansion* decision points each participant pair (y-axis) identified over time (x-axis, minutes). Red lines show when *Expansion* events actually occurred. (Participants noticed most of them.) The red box shows where Pair 4 failed to notice an event they likely wanted to note, based on their previous and subsequent behavior.

5.3.1 Identifying Consistently Important Decisions. In fact, participants showed remarkable consistency about the importance of the expansion subcategory of *Building*. Eight of the ten participant pairs identified *Expansion* decision points, when a player chooses to build a new resource-producing base (Table 10). Extra resources from expanding allow a player to build more units, thereby gaining an economic advantage over their opponent.

Even so, they missed some of the cues pointing out expansion decisions. The event logs in the replay file reveal that new bases were constructed at roughly 1:00, 1:30, 2:00, 5:00, 6:30, 11:20, 12:00, and 13:45, each of which is marked with a red line on Figure 9. Only Pair 3 identified decision points for all 8 of these, and 7 pairs omitted at least one, with one example highlighted with a red box in Figure 9. Table 10 shows Pair 4 also finding eight expansion decision points, but one of those is about the commitment to expand, based on building other structures to protect the base, rather than the action of building the base itself.

5.3.2 Reasons for Distraction. Since expansion decisions were so important to participants, why did they miss some? "Distractor cues" in the information environment led participants *away* from desirable prey.

Recall that cues are the signposts in the environment that the predator observes, such as rabbit tracks. Scent, on the other hand, is what the predators make of cues in their heads, such as thinking that rabbit tracks will lead to rabbits. Participants were so distracted by cues that provided an alluring scent, albeit to low-value information, they did not notice the other cues pointing toward "Expansion" decisions. Distractor cues led participants astray from Expansion in nine cases and eight of them involved units in combat or potentially entering combat. (The ninth involved being distracted by a scouting unit.) For example, Pair 7 missed the expansion at the 13:45 minute mark, instead choosing to track various military units, which turned out to be unimportant to them:

Pair7-P14: "*These zerglings [inexpensive units] are still just chilling.*"

Interestingly, participants had trouble with distractor cues even when the number of entities competing for their attention was low. For example, even when the game state was simple—such as at 1:30 when the game had only 13 objects—participants *still* missed the Expansion events.

If some decision points went unnoticed even in simple game states, what *did* they notice, even in complex ones? *Fighting*. All participants agreed *Fighting* was key, identifying at least one decision point of that type (Table 9). The ubiquity of fighting codes is consistent with Kim et al. [29], who found that combat ratings were the most important to the participant's perception score.

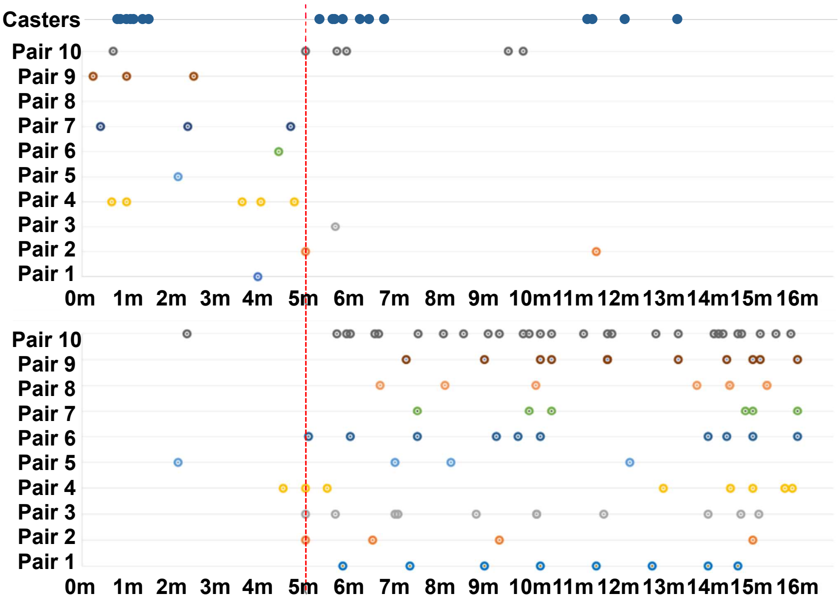


Fig. 10. **Top (Scouting):** The *Scouting* decision points identified by Game Enthusiast pairs (y-axis) over time (x-axis, minutes, bottom 10 rows), in comparison with Shoutcaster Team 11 utterances about scouting (top row). Notice that the enthusiasts (but not the shoutcasters) mostly ignore scouting after the red line depicting when *Fighting* events begin. **Bottom (Fighting):** The *Fighting* decision points Game Enthusiast pairs identified. As the two graphs together show, from the time *Fighting* events begin (red line), *Scouting* decision points were rarely noticed—despite important *Scouting* actions continuing to occur, as evidenced by shoutcasters' continued discussion of scouting (topmost row).

Fighting provided such a strong scent that it masked most other sources of scent, even those which participants prioritized very highly.

For example, as Figure 10 shows, the start of fighting decision points coincides with the time that scouting decision points vanish—even though scouting occurred throughout the game and that participants believed scouting information mattered:

> Pair4-P8: "*But it's important just to know what they're up to and good scouting is critical to know who you are going to fight.*"

5.3.3 Implications for a Future Interactive Explanation System. Participants had a tendency to follow cues that were interesting or eye-catching, at the expense of those that were important but more mundane. In this domain, the "eye-catching" cues were combat-oriented, whereas the "mundane" cues were scouting oriented. Other domains may have similar phenomena, wherein certain aspects of the agent's behaviors distract from other important cues due to triggering an emotional response in the viewer. As Chi explained, "*...A wealth of information creates a poverty of attention and a need to allocate that attention efficiently among the overabundance of information sources that might consume it*" [11]. Supporting users' attending to actions that are both mundane *and* important is a design challenge for future interactive explanation systems.

6 RESULTS: WHAT ABOUT WHEN THE PLAYER REALLY IS AN AI AGENT?

In Sections 4 and 5 we investigated how people analyze *human* play, to provide formative data toward explaining a real AI agent. At the time of those two studies, AI agents were not capable of

	Player: TLO					Player: MaNa					
Perspective	1	2	3	4	5	1	2	3	4	5	exhibition
live shoutcast	✓		✓			✓		✓	✓		✓
non-live analysis by *shoutcaster*					✓	✓	✓	✓	✓	✓	✓
non-live analysis by *player*						✓	✓	✓	✓	✓	✓

Table 11. Games found in Study 3's data corpus and the various perspectives from which they were analyzed. Please consult our supplemental materials for transcripts (just the live shoutcasts) and links to the videos.

playing an RTS game well enough to compete with humans—but then DeepMind demonstrated an agent capable of beating professional StarCraft players [73]. This development afforded us a chance to observe people trying to explain and understand a real AI agent.

The agent capable of defeating professional StarCraft players, AlphaStar, did so resoundingly[16]. However, during each type of match analysis (shown in Table 11: live shoutcasting, non-live analysis from the shoutcaster, and non-live analysis from the player), these professional players and shoutcasters suggested that AlphaStar exhibited strange behaviors.

Since AlphaStar kept winning, these players and shoutcasters could not simply dismiss the unorthodox choices. Now, we consider the extent to which these professionals were able to understand and explain[17] such behaviors when they initially responded to, investigated, and judged them.

6.1 Inexplicably non-"optimal" choices: AlphaStar rarely makes a "Wall."

Shoutcasters and player were both mystified as to why AlphaStar rarely arranged its base to form a wall[18]. That AlphaStar rarely exhibited this behavior, people found inexplicable.

The first time Shoutcaster Team 12 observed this, they expressed surprise:

Shoutcaster Team 12: " *This is a tactic that we use in Protoss vs Protoss to make kind of a wall, so that your opponent can't get into your base, but AlphaStar is skipping that altogether. There are* ***very few Protosses in Europe that don't wall off.***"

They went on to describe some dire consequences that can arise from the decision *not* to wall off:

Shoutcaster Team 12: "*But it's just I'm so interested right now because I see, for instance, [player] TLO is making two adepts [mobile unit], and this is the main reason why you'll make a wall at your ramp, so that those adepts can't come in and start to destroy your probes [workers].*"

In MaNa's game 3, after Shoutcaster Artosis had observed several more games, AlphaStar actually did choose to make a wall—garnering surprise that it had made an exception:

Shoutcaster Artosis: " *We see that this, this agent of AlphaStar has decided to wall that top of that ramp, so, you know, we've seen that off and on but mostly not happening.*"

From the player perspective, Player MaNa was also very surprised when AlphaStar finally used a wall, and bemoaned the lack of discernable patterns in the agent[19]

Player MaNa: "*I see the wall off, like, what, what is, what is going on? Like, the previous two games there was no wall-off, this time it's was a wall-off. So what is the approach here? What is he, what is AlphaStar going to do? I don't see any tendencies.*"

In the end, the shoutcasters made it quite clear they thought walling off is an "optimal" choice:

[16] The various AlphaStar agents beat both pro players 5-0, though MaNa was able to win the exhibition match after DeepMind forced AlphaStar to control the game in a more humanlike way [73].

[17] Artosis and MaNa were a little faster analyzing replays than the participants in our game enthusiasts study (Table 8), but the proportions in how those participants and the professionals here allocated their time were similar. The Real-Time Ratios for the non-live analyses reported in section ranged from 0.67 to 2.69 with Mean=1.25 ± .51 and Median=1.15.

[18] Making a wall refers to using buildings (mostly) to block off the entrance to one's base, making it harder to attack. An example of a wall is shown in Figure 11.

[19] One reason for the lack of discernable tendencies may lie in DeepMind's use of 5 different network configurations which were combined to form the AlphaStar "agent"—which really is more of a *team* of agents.

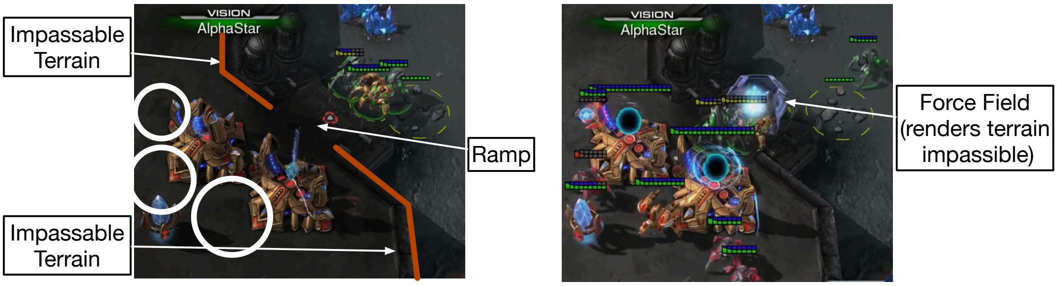


Fig. 11. Two frames from MaNa's analysis of game 5, separated by 2 seconds. The frames show a risky choice the AI made that most human players would avoid; and the AI then getting punished for it. **Left:** A ramp, with AlphaStar's army at the bottom. Around the ramp is terrain (highlighted by the superimposed red lines) that cannot be traversed by most ground units. At the top of the ramp, AlphaStar cannot see what enemy units await (note the superimposed white circles are empty). **Right:** Uh oh! Player MaNa did indeed have an army above the ramp (where white circles were), including a sentry. The sentry has cast a "Force Field"—which renders terrain impassable. The sentry used it to block the ramp, so AlphaStar's stalker in the top right is now prevented from joining the fight.

> Shoutcaster Team 12: *" Now, I think there are ways that you can get away with not walling off at the ramp. <later> Once again, we still have agreed, as a Starcraft 2 community, that* ***walling off is the optimal way*** *to play Starcraft 2—unless you are 100% convinced that you have the right answer for a potential first two adepts [mobile unit]."*

Thus, Shoutcaster Team 12 and Player MaNa clearly agreed that AlphaStar's rarity of walling off was a failing, and could not understand why AlphaStar rarely chose to engage in this presumably optimal behavior.

However, during the exhibition match, the longest-trained AlphaStar did choose to make a wall, and received the following response:

> Shoutcaster Team 12: *" We see that this agent of AlphaStar has decided to wall that top of that ramp. So, you know, we've seen that off and on, but mostly not happening. It is so interesting to think that it taught itself then though, because it has been, you know, playing against itself. The reinforcement learning that eventually, you know, one AlphaStar made two adepts [mobile unit] and killed a whole bunch of probes [workers] while the other AlphaStar is like 'Hey, I should actually start walling off!' Enough AlphaStars in a room for 200 years, and you get some very strong strategies. One of them was the one to open up with adepts..."*

This quote makes clear one of the challenges for AI learning that behaviors like walling are good—it is contingent on the opponent making specific choices.

6.2 Understanding "risky" choices: AlphaStar does not fear going up "Ramps."

For human StarCraft players, going up ramps is often considered to be very risky, because vision of what lies at the top of the ramp is limited, as shown in Figure 11. Yet AlphaStar considers going up ramps to be much more viable than humans do.

> Shoutcaster Team 12: *" Very risky by AlphaStar to go through a ramp like this is a very narrow ramp. <later> We are very worried in general to go up ramps, because once again, the force field is an incredibly powerful spell [also shown in Figure 11] in Protoss vs Protoss."*

AlphaStar's fearlessness paid great dividends during two games, resulting in wins and accolades:

> Shoutcaster Artosis: *"I feel like this is almost like a lesson, this game. Indecisiveness. It's like if you've chosen the strategy, you live by the sword, you die by the sword."*

In Game 1, MaNa also described how unexpected it was for his opponent to simply walk up the ramp—which led to AlphaStar winning because of a play error MaNa made:

Player MaNa: *"I forgot to turn this gateway [unit producing structure] into the warpgate [an upgraded unit producing structure]. That's a crucial mistake that should not have happened... ...so what everybody would do, a human player is like, look, it scouted my 4-gate [4 gateways] right? I'm* ***not going to be walking out that ramp****. There should be a sentry [support unit] ready, but, I made a mistake here."*

Later, in game 5 MaNa made a similar observation:

Player MaNa: *" But AlphaStar just goes in [up a ramp]. Doesn't care if I have a Sentry [support unit] or not... I do not."*

The Sentry unit is what can make going up a ramp dangerous[20], causing Artosis to observe:

Shoutcaster Artosis: *" It felt like sentries [support unit] were good against AlphaStar. It totally out micros in situation like stalker vs. stalker [mobile unit] but as soon as you have something like sentries it seems like it would fall down. Some of the TLO games were like that as well where it was just like shoving its way up here [ramps]."*

Toward the end of his non-live analysis, Shoutcaster Artosis offered the following summary of the weaknesses he had identified:

Shoutcaster Artosis: *"Over the course of these 11 games, the things that we found out is, AlphaStar had most problems with forcefields [spell] and ramps in general. It seems like the most mistakes were made in those types of situations. But it also got wins because of some of those moves too, right? Like, there's a lot of times where you give a ton of respect to a ramp as a human player because you have been hurt so many times trying to go up a ramp."*

This illustrates a challenge in understanding the strengths and weaknesses of AI systems, as the same risky behavior might be *both* a strength and a weakness simultaneously.

6.3 "Worthless" or "worthwhile" after all? AlphaStar employs tactics that are initially laughable.

Despite the overall strong performance from AlphaStar, there were a few moments where it made obviously bad decisions, garnering the amusement of the humans observing it. In the following example, MaNa managed to cause AlphaStar to move its entire army back and forth across the map several times, responding to harrassment from a single dropship.

Shoutcaster Artosis: *"Oh. Time to go home again. All the way from up there... Yeah. It's totally bugged out. It has no idea what to do here. It's just so funny. It keeps blinking [casting a spell that teleports stalkers a short distance]! I love it."*

Sometimes, however, when human observers' initial response was amusement, it then gave way to some respect. For example, in MaNa Game 5, AlphaStar made a series of *very* unorthodox choices (employing a shield battery rush, attempting a gas resource steal, and making a pylon in MaNa's base when he blocked the gas steal). MaNa's initial reaction to all of this was to laugh:

Player MaNa: *"So I don't let AlphaStar take my gas [resource node]. AlphaStar is like confused like, "Oh boop beep bawp beep beowp!" <****laughs****> Doesn't know what to do. So what AlphaStar does at this moment (and this confuses the crap out of me) he makes a freaking pylon [support structure] in my main! And at this point I'm losing my mind. I think I* ***laughed*** *at this moment. Yeah, I zoomed in on the pylon. Just realized that."*

Shoutcaster Artosis had a similar initial response:

[20] The Sentry is a unit capable of casting a spell called "Force Field" which makes a portion of the terrain impassable. This spell can be used to close off a ramp entirely, allowing a player to "split" the incoming army and fight only half at a time (as MaNa has demonstrated in the right side of Figure 11).

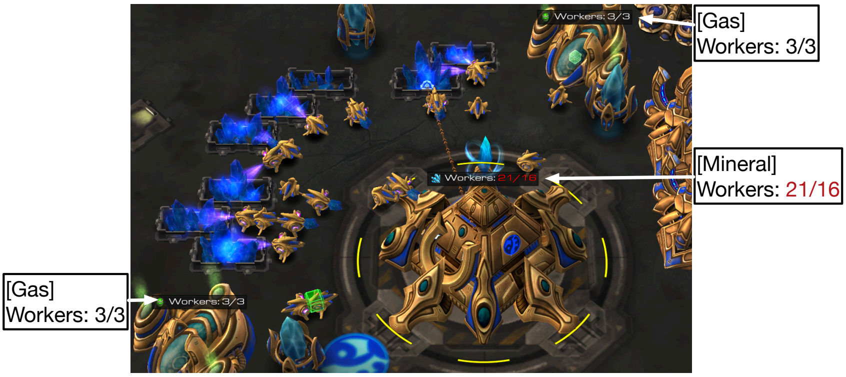


Fig. 12. Screenshot from replay file at 2:55 in AlphaStar vs MaNa game 2. We added callouts to highlight AlphaStar's worker allocation. Note that the interface suggests that 16 is the "correct" number of workers with which to harvest minerals, while AlphaStar has chosen to allocate 21. Note that this "Oversaturation," (as the shoutcasters call it) causes the text to turn red—compare to the glyph for workers harvesting gas.

Shoutcaster Artosis: *"Wow that's fascinating. Why did it make a Pylon? That seems to be about as* ***worthless as anything could ever be****."*

However, after about 8 minutes of observation and analysis, Artosis softened, stating:

Shoutcaster Artosis: *" Was that Pylon [support structure] something to buy time for the forward shield battery [a structure which heals units]? Because obviously if that Pylon is not there, the stalkers [mobile units] are down, looking around, right?...* ***The pylon helped it, no matter how you look at this, that Pylon was a worthwhile sacrifice****."*

Player MaNa, however, did not credit the pylon—instead crediting AlphaStar with a "master plan" and blaming himself:

Player MaNa: *" This game is... the most wrongly judged game that I have played in a while. Because I was losing 4-0, I thought that AlphaStar has a mastermind plan that it's not what it seems to be, you know? That this is something different. It's a* ***master plan****... <a few minutes later> ...And I am like "Oh my god. This is some bulls***." Because, what AlphaStar did, made no sense for me. I had no idea. I was so in the dark... <a few minutes later> This, this Immortal was super lucky, and this game I should have won, just like game 1, but because* ***I'm just a human player, I make mistakes****."*

The above example shows AlphaStar exhibiting a series of very strange behaviors in a single game. Both MaNa and Artosis found the behavior laughable at first, but Artosis then warmed to this tactical choice. His change in attitude toward this non-traditional choice AlphaStar made is a microcosm of our next example, which stretches across almost all the games AlphaStar played.

6.4 "A different kind of StarCraft": AlphaStar may have revolutionized resource gathering in StarCraft

In some ways, the AlphaStar agent breaks years of tradition. A prime example is that the conventional wisdom in StarCraft is to make 16 workers harvesting minerals in each base—a belief so strong that it has been baked into the glyphs of the game, as shown in Figure 12. However, AlphaStar consistently (but not always) produced 20+ workers harvesting minerals in its main base.

At first, in TLO game 3, the unorthodox play from AlphaStar was greeted with surprise:

Shoutcaster Team 12: *"Did you see the amount of probes that AlphaStar has in the main base? [AlphaStar has 27] I mean, it is technically mining a few more minerals, but in Starcraft 2,* ***we like to say that 16 probes on a mineral line is perfect****. Technically, you mine a little more* *with 18, a little more with 20,... but that difference is so small that you technically don't want to go there as a competitive Starcraft 2 player. We normally try to minimize the amount of extra workers"*

However, AlphaStar exhibited this behavior in most of the games. This, combined with how successful the AlphaStar agent was vs the pros, led the shoutcasters' attitudes about it to evolve, as shown by the following quotes, presented in chronological order:

Shoutcaster Team 12, TLO game 3: *" AlphaStar, who, once again, is* ***super saturating his nexus here [with 27 workers]****. It's definitely creating a pretty big advantage, since it has been building probes non stop as well. Our observer showing us the income right now on the side of AlphaStar quite a bit higher than the income of TLO [≈1800 to ≈1550]."*

Shoutcaster Team 12, MaNa game 1: *"Three or four probes already going down. Five. Well, one was a scout as well, so this is kind of a funny thing about AlphaStar though is that AlphaStar always* ***overmakes probes****, so the economy is not actually that bad."*

Shoutcaster Team 12, MaNa game 1: *"Each game so far, AlphaStar has been pretty slow on [making a second base], but continually making those probes.* ***21 in that main base*** *so far."*

Shoutcaster Team 12, MaNa game 4: *" We have seen TLO killing quite a few probes and seen MaNa do some damage as well to that economy. AlphaStar just kind of* ***makes a lot of probes to make up for that****... Most pros have come to the conclusion that losing workers this early into the game is not optimal. That's like a very long standing thing that is believed about Starcraft, you know. Starcraft's been out for over 20 years now. That is one of the basic tenets; don't lose your workers early on. They are very important, building up that economy."*

Shoutcaster Team 12, MaNa game 4: *" One big difference that AlphaStar is doing that almost no other professional StarCraft 2 player does is the* ***amount of workers that is being produced on one base****. You know, up to 24 like going over 24 is 100% pointless, but in general, going from, you know, above 16, there is just very slow progression, and we have decided, as a community, that we really don't want to go over 16 too much. Maybe 18 is acceptable, but normally, you don't see 24 workers on one base. That is one thing that truly stands out for me."*

From a game opponent's perspective, Player MaNa also seemed impressed, stating toward the end of the live shoutcast:

Player MaNa: *"After losing 5-0, I have learned a little bit. The way that AlphaStar played the matchup was not that something I had any kind of experience with. It was very hard to judge whatever AlphaStar was doing.* ***It was a different kind of StarCraft that I've ever played and it was a great experience for me to learn something new from an AI.****"*

Shortly after saying that, MaNa played the exhibition match. Indeed, he did learn something from the AI, as the shoutcasters observed during that match:

Shoutcaster Artosis: *" I'm a little bit surprised by Mana's approach here... He has too many workers, unless he's learning from AlphaStar already [MaNa has 21 workers on minerals]... He was saying that maybe he learned something from this. I mean,* ***if AlphaStar went 10-0 against pro gamers by overmaking probes, maybe all of us have been undermaking probes****."*

Note that these observations, made during the live analysis, caused Artosis to begin his non-live analysis with a desire to investigate this behavior (see Table 12, question 3). To do so, he consistently looked at the effect of the probe "oversaturation" by switching to the income tab at various points of time. At one such time, his response was:

Shoutcaster Artosis: *"Lets check that income tab again. Seventeen probes [for MaNa]. Then twenty-four probes [for AlphaStar]. And obviously that will fluctuate. But you can see that there is more income for the more probes."*

Toward the end of his analysis he summarized these checks as follows:

Beginning Questions + Ideas
1. The AlphaStar Agents that played TLO were not particularly good at Macro or Micro. Then how did they win? This might make them more interesting than the Agents who played Mana.
2. For Agents who were good at Micro or Macro, what did they do to allocate themselves to get into that position?
3. Higher Probe counts seemed to be consistent across all Agents. Why is this good or bad?
4. What did AlphaStar do well to win each game? Are there patterns here?
5. What are the interesting moves that AlphaStar did?

Table 12. Questions Artosis formed after his session shoutcasting games with AlphaStar, carried into his session doing non-live analysis. He consistently referred back to these questions between games.

> Shoutcaster Artosis: *" Throughout that I continued to look at... the oversaturation. Like every game, AlphaStar has a higher income and I mean this is not a big surpirse to anyone. We all know that sixteen out of sixteen is not completely perfect saturation but its very close. Many of the games it was sitting around twenty in the main base and that was giving a reasonable income boost over AlphaStar's opponents."*

Artosis ended his analysis session with a discussion[21] of the reasons why one might choose to oversaturate workers and the implications of that decision. One possibility is that extra workers provided resilience to the loss of workers, e.g.:

> Shoutcaster Artosis: *"It seemed like its Defense was based on... having enough probes, so when the oracle [flying unit, effective at killing workers] flew in and killed 5, it didn't really matter. "*

Another possibility Artosis proposed was that AlphaStar was producing the extra workers with the intent to send them to its next base as soon as it was ready for them:

> Shoutcaster Artosis: *"The overproduction of probes, if an oracle [flying unit] does not come in, the overproduction of probes is much, much better than that shield battery [structure which heals units] is it not? You just transfer them to your natural expansion. In StarCraft I, you are continually making workers with both Terran and Protoss for a very long time. Because transfers are such a big deal in StarCraft 1, you really want your base saturated as soon as... possible."*

Notably, Artosis's discussion of why one might choose to oversaturate workers was a detailed case analysis—and in some senses, a "rationale generation" exercise[22]. Note that explanation in the XAI sense has an element of introspection on the part of the actor (i.e. "Why did *you, specifically you,* do that?"), whereas rationale generation does not. Instead, it answers a slightly different question, namely, "Why would *anyone* do that?". However, despite rationale generation's capacity to help the human understand the domain—in this context, it may devolve into rationalization (meant here to carry the "connotation of making excuses" [15]). This points at a possible pitfall in human evaluation of AI, as Artosis noted:

> Shoutcaster Artosis: *" There's a lot of moves that AlphaStar does that... I can't figure out a good reason for them, right.* ***You don't want to just rationalize everything****, but I just wonder, right? The stalkers just move down and then as the probe came back they turn around and walk back a little bit and then they turn around again. And there's behaviors like this actually pretty often. And I can't come up with much of anything that would really explain it..."*

This remark brings up the danger of rationalization, namely that ascribing reasons for a behavior *post hoc* may not align with reasons *actually considered by the agent* when the behavior was chosen.

[21] See our supplemental materials for a transcript of the text document he created performing case analysis on oversaturation.

[22] We adopt Ehsan's usage of 'rationale': "*to refer to natural language-based post-hoc explanations that are meant to sound like what a human would say in the same situation. We opt for 'rationale generation' instead of 'rationalization' to signal that the agency lies with the receiver and interpreter (human being) instead of the producer (agent)*" [15].

6.4.1 Implications for a Future Interactive Explanation System. One unifying thread in this section is that often the “rationale” for the AI’s behavior was not known until much later. This will be a challenge for XAI systems, and may be related to the rarity of Why in this domain (e.g. an event happens, another event happens [possibly much later], and the semantic link between the two events). In these analyses, the assessors had all the pieces, but still took quite a while to assemble them—despite their significant domain expertise. An even greater challenge for XAI systems might be cases of Why where there is an event, a semantic link, and the *avoidance* of a future event, e.g. one gets a vaccine so they *do not* get a disease.

7 DISCUSSION

7.1 Foraging Costs in XAI

Throughout this paper, we have used Information Foraging Theory (IFT) to gain new insights into what and how to explain in XAI systems. IFT enabled us to abstract beyond game-specific objects to constructs grounded in an established theory for humans’ information seeking behaviors—allowing us to “connect the dots” between our work and other research about people seeking information.

Taken together, the IFT implications highlight the costs our participants bore in the difficult foraging problems they faced—some of which are new to IFT research. As one example, when the participants did not follow the “right” paths, they paid a high *information costs* in the form of lost information. However, it was not easy to find a “right” path quickly in the ever-changing game environment, and attempting to do so exacted a high *cognitive cost*. Even when participants relaxed the real-time pressure by pausing the replay and rewinding to review a sequence again, rewinding then incurred a high *navigation cost* for the rewind-positioning and pausing. On top of this, rewinding also incurred an additional *cognitive cost* of keeping track of both prior context and current context to keep them together.

In fact, in XAI, “lightweight foraging” may be all that participants can afford to do, due to the costs discussed above. Ragavan’s work with IFT coined this term in the context of examining foraging behaviors in Version Control Systems [59]. In software development projects with Version Control Systems, changes from multiple developers can occur more rapidly than any one developer has time to scrutinize, especially in large teams with many developers pushing commits. Ragavan’s participants cared primarily about changes that affected their current task (more than 70%), and ignored other information. This style of lightweight foraging was similar to foraging behaviors our study’s participants also showed. However, Ragavan’s participants seemed more focused on relevance than on cost, whereas our study’s participants faced so many costs, they had little opportunity to think beyond cost. Such high costs point to the challenges creators of future explanation systems will need to overcome to create interactive explanation systems in which domain experts can find what they need to know, when they need to know it.

7.2 How XAI Informs IFT

We now consider the opposite direction from most of this paper, and consider what XAI can contribute to IFT.

The RTS domain presents a complex and rapidly changing environment, more so than other IFT environments in the literature, e.g. Integrated Development Environments (IDEs) and web sites [16, 17, 55, 58]. In the RTS domain, hundreds of actions happen *each minute*. Further, the environment is continually affected by actions that do not originate with the forager.

The fact that participants faced many paths and had to rapidly triage which paths to follow (Section 5.2) presents an interesting IFT challenge. Previous research [55] identified a “scaling up problem” in IFT—a difficulty estimating value/cost of distant prey without paying those costs first in order to make the estimate. The XAI version of a scaling up foraging problem is different (recall Figure 8). Here, participants' foraging paths were short, but the many paths simultaneously available then required participants to compare the multiple paths' value/cost estimates to decide among them. This phenomenon can be regarded as a "breadth version" of the IFT scaling problem.

Previous work [55] has shown that "prey in pieces" creates foraging challenges, because finding and assembling all the bits can be tedious and error-prone. In the XAI setting, participants' prey is an agent's decision process, which is likewise "in pieces"—meaning that bits of it are scattered over many patches. The "prey in pieces" problem becomes especially pronounced in XAI because it may not be clear to a domain expert which bit of evidence is the "last" bit of the prey they need, whether there is more they need to know to understand the agent's decision—or even whether the same collection of evidence they have gathered would result in the agent behaving the same way in the future. One thing XAI reveals to IFT from this is an understanding of a new variant of the prey-in-pieces problem.

7.3 Threats to Validity

Every study has threats to validity [74]. For example, in qualitative analysis, segmenting spoken language into sentences can pose problems. In our study, much of our data contained events happening live and multiple speakers talking over one another, which introduced challenges in deciding how to segment. It is possible that the segmentation affected our coding, (e.g., perhaps some Why's got cut half during segmentation).

Aspects of our game enthusiasts study may have influenced participants to ask less questions, such as when they did not have a reasonable expectation of an answer from their partner or the interface. Some potential questions were also phrased as statements, such as:

> Pair9-P18: "*It's interesting that it [the "AI"] still hasn't taken its Vespene Geysers [resource nodes] at some of its bases.*"

Note how this statement begs the question of "*Why didn't* the agent take its resource nodes?". Also, participants took different amounts of time to do the task, ranging from 20 minutes to an hour—so certain participant pairs talked more than others. These effects create a form of sampling bias.

Similarly, the researchers assigned participants into their pairs during the game enthusiast study. This could have led to participants who were not well matched personally or stylistically, which could affect both their utterances and their foraging strategies. Alternatively, it also could be that utterances/strategies shifted during the study as the participants got to know each other.

Some aspects of the game enthusiasts study design may also have increased focus on certain temporal aspects of the game. This focus adjustment, in addition to the lab setting, probably made the tasks less representative of how experienced domain experts would forage for information to assess an intelligent agent.

We recruited domain experts (people with at least some gaming experience) to be our participants. However, domain experts are a simpler case for XAI than novices, because domain experts already have a foundation of concepts and vocabulary upon which an explanation can build. How our results generalize to populations without such domain expertise is an open question. Another possible threat is that our participants in the game enthusiasts study were primarily university students. An advantage is that university students are a popular demographic for RTS gaming, and this helped assure that participants would have the motivations and background we sought for our investigation. Still, students may be different from other game-playing populations such as in their social behaviors and life experiences, which may affect the generality of our results.

Last, we selected a small set of game replays from a very large pool, using criteria that focused on high-level competitions. It could be that our sample does not represent the population well.

Additionally, in the game enthusiast study, it could be that our task was not well matched to future human assessment of AI systems, since the pro players likely exhibited few discernable "bugs."

Threats like these can be addressed only by additional empirical studies across a spectrum of study designs, types of intelligent interfaces, and intelligent agents—and, most importantly, with an explanation interface.

8 CONCLUSION

In this paper, we have considered explainable AI in the RTS domain using information foraging theory perspectives from both expert explainers (shoutcasters) and domain experts. We also investigated how these expert explainers put together explanations in this domain, with an eye toward informing the content in future explanation systems for the RTS domain. Our third study then considered new challenges arising when experts try to understand and explain a real AI's actions in this domain. Among the results were:

- *Satisficing:* As model explainers, the shoutcasters revealed strategies for "satisficing" with explanations that may not have precisely answered all the questions the audience had in mind, but were feasible given the time and resource constraints in effect when comprehending, assessing, and explaining—all in real time as play progressed. These strategies may be likewise applicable to interactive explanation systems.
- *Strategic use of language:* The detailed contents of the shoutcasters' explanations revealed patterns of how they paired properties with different objects and actions. Interactive explanation systems may be able to leverage these patterns to communicate succinctly about an agent's tactics and strategies.
- *"What", not "Why":* The shoutcasters and the game enthusiasts alike favored What information over the abundance of Whys in others' previous research. Their Whats were nuanced, complex and, especially for the game enthusiasts, sometimes expensive.
- *Where the most information lay:* The most commonly used information patch types by both the shoutcasters and the game enthusiasts were the Actuators ("A" in the PEAS model). This suggests that explanation systems should consider maximizing availability of information from the Actuators, with strong support also for Environment and Sensor information.
- *Costs, costs, and more costs:* The dynamically changing RTS environment and the breadth-heavy information structure brought to both the shoutcasters and the game enthusiasts unique foraging problems with high navigation, information, and cognitive costs.
- *The unconventional: strength or weakness?* When evaluating the AlphaStar RTS agent, shoutcaster and game enthusiast alike ran into cases where it was *really* hard to tell a strength from a weakness. In lieu of explanations by the system itself, the humans attempted to rationalize the agent's behaviors in ways that seemed logical to them. It is possible that rationalizations like these may do more harm than good to people's mental models when they do not match the *actual* reasons the AI behaved the way it did.

More generally, the Information Foraging Theory perspective on these humans' attempts to explain and understand an AI provided a unifying view of the challenges ahead in explaining AI to people who are not AI experts. The use of this theory revealed opportunities for future explainable AI systems, to enable domain experts to find the information they need to understand, assess, and ultimately decide when and how much to trust their intelligent agents.

9 ACKNOWLEDGMENTS

This work was supported by DARPA #N66001-17-2-4030 and NSF #1314384. Any opinions, findings and conclusions or recommendations expressed are those of the authors and do not necessarily reflect the views of NSF, DARPA, the Army Research Office, or the US government.

REFERENCES

[1] Adrian K Agogino and Kagan Tumer. 2004. Unifying temporal and structural credit assignment problems. In *Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems-Volume 2*. IEEE Computer Society, 980–987.

[2] S. Amershi, M. Cakmak, W. Knox, and T. Kulesza. 2014. Power to the people: The role of humans in interactive machine learning. *AI Magazine* 35, 4 (2014), 105–120.

[3] Andrew Anderson, Jonathan Dodge, Amrita Sadarangani, Zoe Juozapaitis, Evan Newman, Jed Irvine, Souti Chattopadhyay, Alan Fern, and Margaret Burnett. 2019. Explaining Reinforcement Learning to Mere Mortals: An Empirical Study. In *IJCAI*.

[4] Balaji Athreya and Chris Scaffidi. 2014. Towards aiding within-patch information foraging by end-user programmers. In *IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*. IEEE, 13–20.

[5] Juan Felipe Beltran, Ziqi Huang, Azza Abouzied, and Arnab Nandi. 2017. Don't just swipe left, tell me why: Enhancing gesture-based feedback with reason bins. In *International Conference on Intelligent User Interfaces*. ACM, 469–480.

[6] Sourav S Bhowmick, Aixin Sun, and Ba Quan Truong. 2013. Why Not, WINE?: Towards answering why-not questions in social image search. In *ACM International Conference on Multimedia*. ACM, 917–926.

[7] Svetlin Bostandjiev, John O'Donovan, and Tobias Höllerer. 2012. TasteWeights: a visual interactive hybrid recommender system. In *ACM Conference on Recommender Systems*. ACM, 35–42.

[8] Barrett S Caldwell, Sandra K Garrett, and Karim C Boustany. 2010. Healthcare team performance in time critical environments: coordinating events, foraging, and system processes. *Journal of Healthcare Engineering* 1, 2 (2010), 255–276.

[9] Nico Castelli, Corinna Ogonowski, Timo Jakobi, Martin Stein, Gunnar Stevens, and Volker Wulf. 2017. What Happened in my home? An end-user development approach for smart home data visualization. In *ACM Conference on Human Factors in Computing Systems*. ACM, 853–866.

[10] Gifford Cheung and Jeff Huang. 2011. Starcraft from the Stands: Understanding the Game Spectator. In *ACM Conference on Human Factors in Computing Systems (CHI '11)*. ACM, New York, NY, USA, 763–772. https://doi.org/10.1145/1978942.1979053

[11] Ed H Chi, Peter Pirolli, Kim Chen, and James Pitkow. 2001. Using information scent to model user information needs and actions and the web. In *ACM Conference on Human Factors in Computing Systems*. ACM, 490–497.

[12] Robert Collins and David Jefferson. 1991. Representations for artificial organisms. In From Animals to Animats. In *Proceedings of the First International Conference on Simulation of Adaptive Behavior*. MIT Press.

[13] Kelley Cotter, Janghee Cho, and Emilee Rader. 2017. Explaining the News Feed Algorithm: An Analysis of the "News Feed FYI" Blog. In *ACM CHI Conference Extended Abstracts on Human Factors in Computing Systems*. ACM, 1553–1560.

[14] Jonathan Dodge, Sean Penney, Claudia Hilderbrand, Andrew Anderson, and Margaret Burnett. 2018. How the Experts Do It: Assessing and Explaining Agent Behaviors in Real-Time Strategy Games. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems (CHI '18)*. ACM, New York, NY, USA, Article 562, 12 pages. https://doi.org/10.1145/3173574.3174136

[15] Upol Ehsan, Pradyumna Tambwekar, Larry Chan, Brent Harrison, and Mark O. Riedl. 2019. Automated Rationale Generation: A Technique for Explainable AI and Its Effects on Human Perceptions. In *Proceedings of the 24th International Conference on Intelligent User Interfaces (IUI '19)*. ACM, New York, NY, USA, 263–274. https://doi.org/10.1145/3301275.3302316

[16] S. Fleming, C. Scaffidi, D. Piorkowski, M. Burnett, R. Bellamy, J. Lawrance, and I. Kwan. 2013. An information foraging theory perspective on tools for debugging, refactoring, and reuse tasks. *ACM Transactions on Software Engineering and Methodology (TOSEM)* 22, 2 (2013), 14.

[17] W. Fu and P. Pirolli. 2007. SNIF-ACT: A cognitive model of user navigation on the world wide web. *Human-Computer Interaction* 22, 4 (2007), 355–412.

[18] Sandra K Garrett and Barrett S Caldwell. 2009. Human factors aspects of planning and response to pandemic events. In *IIE Annual Conference. Proceedings*. Institute of Industrial and Systems Engineers (IISE), 705.

[19] V. Grigoreanu, M. Burnett, and G. Robertson. 2010. A strategy-centric approach to the design of end-user debugging tools. In *ACM Conference on Human Factors in Computing Systems*. ACM, 713–722.

[20] Valentina Grigoreanu, Margaret Burnett, Susan Wiedenbeck, Jill Cao, Kyle Rector, and Irwin Kwan. 2012. End-user debugging strategies: A sensemaking perspective. *ACM Transactions on Computer-Human Interaction (TOCHI)* 19, 1

(2012), 1–28.

[21] Alex Groce, Todd Kulesza, Chaoqiang Zhang, Shalini Shamasunder, Margaret Burnett, Weng-Keen Wong, Simone Stumpf, Shubhomoy Das, Amber Shinsel, Forrest Bice, et al. 2014. You are the only possible oracle: Effective test selection for end users of interactive machine learning systems. *IEEE Transactions on Software Engineering* 40, 3 (2014), 307–323.

[22] Bradley Hayes and Julie A Shah. 2017. Improving robot controller transparency through autonomous policy explanation. In *ACM/IEEE International Conference on Human-Robot Interaction.* ACM, 303–312.

[23] Steven R Haynes, Mark A Cohen, and Frank E Ritter. 2009. Designs for explaining intelligent agents. *International Journal of Human-Computer Studies* 67, 1 (2009), 90–110.

[24] Zhian He and Eric Lo. 2014. Answering why-not questions on top-k queries. *IEEE Transactions on Knowledge and Data Engineering* 26, 6 (2014), 1300–1315.

[25] Robert R Hoffman and Gary Klein. 2017. Explaining Explanation, Part 1: Theoretical Foundations. *IEEE Intelligent Systems* 32, 3 (2017), 68–73.

[26] Paul Jaccard. [n.d.]. Nouvelles recherches sur la distribution florale. *Bull. Soc. Vaud. Sci. Nat.* 44 ([n. d.]), 223–270.

[27] Ashish Kapoor, Bongshin Lee, Desney Tan, and Eric Horvitz. 2010. Interactive optimization for steering machine classification. In *ACM Conference on Human Factors in Computing Systems.* ACM, 1343–1352.

[28] Lucas Kempe-Cook, Stephen Tsung-Han Sher, and Norman Makoto Su. 2019. Behind the Voices: The Practice and Challenges of Esports Casters. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems (CHI âĂŹ19).* Association for Computing Machinery, New York, NY, USA, Article Paper 565, 12 pages. https://doi.org/10.1145/3290605.3300795

[29] Man-Je Kim, Kyung-Joong Kim, SeungJun Kim, and Anind K Dey. 2016. Evaluation of StarCraft Artificial Intelligence Competition Bots by Experienced Human Players. In *ACM CHI Conference Extended Abstracts.* ACM, 1915–1921.

[30] M. J. Kim, K. J. Kim, S. Kim, and A. K. Dey. 2018. Performance Evaluation Gaps in a Real-Time Strategy Game Between Human and Artificial Intelligence Players. *IEEE Access* 6 (2018), 13575–13586. https://doi.org/10.1109/ACCESS.2018.2800016

[31] Josua Krause, Adam Perer, and Kenney Ng. 2016. Interacting with predictions: Visual inspection of black-box machine learning models. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems (CHI '16).* ACM, New York, NY, USA, 5686–5697. https://doi.org/10.1145/2858036.2858529

[32] Cliff Kuang. 2017. Can AI be taught to explain itself? New York Times. (2017). Retrieved December 26, 2017 from https://www.nytimes.com/2017/11/21/magazine/can-ai-be-taught-to-explain-itself.html.

[33] T. Kulesza, M. Burnett, W. Wong, and S. Stumpf. 2015. Principles of explanatory debugging to personalize interactive machine learning. In *ACM International Conference on Intelligent User Interfaces.* ACM, 126–137.

[34] Todd Kulesza, Simone Stumpf, Margaret Burnett, and Irwin Kwan. 2012. Tell me more? The effects of mental model soundness on personalizing an intelligent agent. In *ACM Conference on Human Factors in Computing Systems.* ACM, 1–10.

[35] T. Kulesza, S. Stumpf, M. Burnett, W. Wong, Y. Riche, T. Moore, I. Oberst, A. Shinsel, and K. McIntosh. 2010. Explanatory debugging: Supporting end-user debugging of machine-learned programs. In *Visual Languages and Human-Centric Computing (VL/HCC), 2010 IEEE Symposium on.* IEEE, 41–48.

[36] T. Kulesza, S. Stumpf, W. Wong, M. Burnett, S. Perona, A. Ko, and I. Oberst. 2011. Why-oriented end-user debugging of naive Bayes text classification. *ACM Transactions on Interactive Intelligent Systems (TiiS)* 1, 1 (2011), 2.

[37] Sandeep Kaur Kuttal, Anita Sarma, Margaret Burnett, Gregg Rothermel, Ian Koeppe, and Brooke Shepherd. 2019. How end-user programmers debug visual web-based programs: An information foraging theory perspective. *Journal of Computer Languages* 53 (2019), 22–37.

[38] Sandeep Kaur Kuttal, Anita Sarma, and Gregg Rothermel. 2013. Predator behavior in the wild web world of bugs: An information foraging theory perspective. In *Visual Languages and Human-Centric Computing (VL/HCC), 2013 IEEE Symposium on.* IEEE, 59–66.

[39] Joseph Lawrance, Margaret Burnett, Rachel Bellamy, Christopher Bogart, and Calvin Swart. 2010. Reactive Information Foraging for Evolving Goals. In *ACM Conference on Human Factors in Computing Systems (CHI âĂŹ10).* Association for Computing Machinery, New York, NY, USA, 25âĂŞ34. https://doi.org/10.1145/1753326.1753332

[40] B. Lim and A. Dey. 2009. Assessing demand for intelligibility in context-aware applications. In *ACM International Conference on Ubiquitous Computing.* ACM, 195–204.

[41] B. Lim, A. Dey, and D. Avrahami. 2009. Why and why not explanations improve the intelligibility of context-aware intelligent systems. In *ACM Conference on Human Factors in Computing Systems.* ACM, 2119–2128.

[42] Brian Y Lim. 2012. *Improving understanding and trust with intelligibility in context-aware applications.* Ph.D. Dissertation. Carnegie Mellon University.

[43] Diane Litman, Steve Young, M.J.F. Gales, Kate Knill, Karen Ottewell, Rogier van Dalen, and David Vandyke. 2016. Towards using conversations with spoken dialogue systems in the automated assessment of non-native speakers of

English. In *SIGDIAL Conference*. 270–275.

[44] M. Lomas, R. Chevalier, E. V. Cross, R. C. Garrett, J. Hoare, and M. Kopack. 2012. Explaining robot actions. In *ACM/IEEE International Conference on Human-Robot Interaction (HRI)*. 187–188. https://doi.org/10.1145/2157689.2157748

[45] S. McGregor, H. Buckingham, T. G. Dietterich, R. Houtman, C. Montgomery, and R. Metoyer. 2015. Facilitating testing and debugging of Markov Decision Processes with interactive visualization. In *2015 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*. 53–61. https://doi.org/10.1109/VLHCC.2015.7357198

[46] Ronald Metoyer, Simone Stumpf, Christoph Neumann, Jonathan Dodge, Jill Cao, and Aaron Schnabel. 2010. Explaining how to play real-time strategy games. *Knowledge-Based Systems* 23, 4 (2010), 295–301.

[47] Tim Miller. 2017. Explanation in Artificial Intelligence: Insights from the Social Sciences. *CoRR* abs/1706.07269 (2017). http://arxiv.org/abs/1706.07269

[48] Nan Niu, Anas Mahmoud, Zhangji Chen, and Gary Bradshaw. 2013. Departures from optimality: Understanding human analyst's information foraging in assisted requirements tracing. In *ACM/ICSE International Conference on Software Engineering*. IEEE Press, 572–581.

[49] Donald A Norman. 1983. Some observations on mental models. *Mental Models* 7, 112 (1983), 7–14.

[50] S. Ontañón, G. Synnaeve, A. Uriarte, F. Richoux, D. Churchill, and M. Preuss. 2013. A Survey of Real-Time Strategy Game AI Research and Competition in StarCraft. *IEEE Transactions on Computational Intelligence and AI in Games* 5, 4 (Dec 2013), 293–311. https://doi.org/10.1109/TCIAIG.2013.2286295

[51] Sean Penney, Jonathan Dodge, Claudia Hilderbrand, Andrew Anderson, Logan Simpson, and Margaret Burnett. 2018. Toward Foraging for Understanding of StarCraft Agents: An Empirical Study. In *23rd International Conference on Intelligent User Interfaces (IUI '18)*. ACM, New York, NY, USA, 225–237. https://doi.org/10.1145/3172944.3172946

[52] Alexandre Perez and Rui Abreu. 2014. A diagnosis-based approach to software comprehension. In *ACM International Conference on Program Comprehension*. ACM, 37–47.

[53] David Piorkowski, Scott Fleming, Christopher Scaffidi, Christopher Bogart, Margaret Burnett, Bonnie John, Rachel Bellamy, and Calvin Swart. 2012. Reactive Information Foraging: An Empirical Investigation of Theory-Based Recommender Systems for Programmers. In *ACM Conference on Human Factors in Computing Systems (CHI âĂŹ12)*. Association for Computing Machinery, New York, NY, USA, 1471âĂŞ1480. https://doi.org/10.1145/2207676.2208608

[54] David Piorkowski, Scott D. Fleming, Christopher Scaffidi, Margaret Burnett, Irwin Kwan, Austin Z Henley, Jamie Macbeth, Charles Hill, and Amber Horvath. 2015. To fix or to learn? How production bias affects developers' information foraging during debugging. In *Software Maintenance and Evolution (ICSME), 2015 IEEE International Conference on*. IEEE, 11–20.

[55] D. Piorkowski, A. Henley, T. Nabi, S. Fleming, C. Scaffidi, and M. Burnett. 2016. Foraging and navigations, fundamentally: developers' predictions of value and cost. In *2016 ACM International Symposium on Foundations of Software Engineering*. ACM, 97–108.

[56] David Piorkowski, Sean Penney, Austin Z Henley, Marco Pistoia, Margaret Burnett, Omer Tripp, and Pietro Ferrara. 2017. Foraging goes mobile: Foraging while debugging on mobile devices. In *2017 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*. IEEE, 9–17.

[57] P. Pirolli. 2007. *Information Foraging Theory: Adaptive Interaction with Information.* Oxford Univ. Press.

[58] S.S. Ragavan, S. Kuttal, C. Hill, A. Sarma, D. Piorkowski, and M. Burnett. 2016. Foraging among an overabundance of similar variants. In *ACM Conference on Human Factors in Computing Systems*. ACM, 3509–3521.

[59] Sruti Srinivasa Ragavan, Mihai Codoban, David Piorkowski, Danny Dig, and Burnett Margaret. 2019. Version Control Systems: An Information Foraging Perspective. *IEEE Transactions on Software Engineering* (2019). https://doi.org/10.1109/TSE.2019.2931296

[60] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. Why should I trust you? Explaining the predictions of any classifier. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 1135–1144.

[61] Stephanie Rosenthal, Sai P. Selvaraj, and Manuela Veloso. 2016. Verbalization: Narration of autonomous robot experience. In *International Joint Conference on Artificial Intelligence (IJCAI'16)*. AAAI Press, 862–868. http://dl.acm.org/citation.cfm?id=3060621.3060741

[62] Quentin Roy, Futian Zhang, and Daniel Vogel. 2019. Automation Accuracy Is Good, but High Controllability May Be Better. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems (CHI '19)*. ACM, New York, NY, USA, Article 520, 8 pages. https://doi.org/10.1145/3290605.3300750

[63] Stuart J. Russell and Peter Norvig. 2003. *Artificial Intelligence: A Modern Approach* (2 ed.). Pearson Education.

[64] Robert Spence. 2007. *Information Visualization: Design for Interaction (2nd Edition)*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA.

[65] Sruti Srinivasa Ragavan, Sandeep Kaur Kuttal, Charles Hill, Anita Sarma, David Piorkowski, and Margaret Burnett. 2016. Foraging among an overabundance of similar variants. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. 3509–3521.

[66] David J Stracuzzi, Alan Fern, Kamal Ali, Robin Hess, Jervis Pinto, Nan Li, Tolga Konik, and Daniel G Shapiro. 2011. An application of transfer to american football: From observation of raw video to control in a simulated environment. *AI Magazine* 32, 2 (2011), 107–125.

[67] S. Stumpf, E. Sullivan, E. Fitzhenry, I. Oberst, W. Wong, and M. Burnett. 2008. Integrating rich user feedback into intelligent user interfaces. In *ACM International Conference on Intelligent User Interfaces*. ACM, 50–59.

[68] Adam Summerville, Michael Cook, and Ben Steenhuisen. 2016. Draft-Analysis of the Ancients: Predicting Draft Picks in DotA 2 using Machine Learning. https://aaai.org/ocs/index.php/AIIDE/AIIDE16/paper/view/14075

[69] Katia Sycara, Christian Lebiere, Yulong Pei, Donald Morrison, and Michael Lewis. 2015. Abstraction of analytical models from cognitive models of human control of robotic swarms. In *International Conference on Cognitive Modeling*. University of Pittsburgh.

[70] J. Tullio, A. Dey, J. Chalecki, and J. Fogarty. 2007. How it works: A field study of non-technical users interacting with an intelligent system. In *ACM Conference on Human Factors in Computing Systems*. ACM, 31–40.

[71] J. Vermeulen, G. Vanderhulst, K. Luyten, and K. Coninx. 2010. PervasiveCrystal: Asking and answering why and why not questions about pervasive computing applications. In *Intelligent Environments (IE), 2010 IEEE International Conference on*. IEEE, 271–276.

[72] Oriol Vinyals. 2017. DeepMind and Blizzard open StarCraft II as an AI research environment. https://deepmind.com/blog/deepmind-and-blizzard-open-starcraft-ii-ai-research-environment/

[73] Oriol Vinyals, David Silver, et al. 2019. AlphaStar: Mastering the Real-Time Strategy Game StarCraft II. https://deepmind.com/blog/article/alphastar-mastering-real-time-strategy-game-starcraft-ii.

[74] Claes Wohlin, Per Runeson, Martin Höst, Magnus C. Ohlsson, Björn Regnell, and Anders Wesslén. 2000. *Experimentation in Software Engineering: An Introduction*. Kluwer Academic Publishers, Norwell, MA, USA.

[75] Kevin Wong. 2016. StarCraft 2 and the quest for the highest APM. https://www.engadget.com/2014/10/24/starcraft-2-and-the-quest-for-the-highest-apm/

[76] Robert H Wortham, Andreas Theodorou, and Joanna J Bryson. 2017. Improving robot transparency:real-time visualisation of robot AI substantially improves understanding in naive observers, In IEEE RO-MAN 2017. *IEEE RO-MAN 2017*. http://opus.bath.ac.uk/55793/

[77] Tom Zahavy, Nir Ben Zrihem, and Shie Mannor. 2016. Graying the black box: Understanding DQNs. In *International Conference on International Conference on Machine Learning - Volume 48 (ICML'16)*. JMLR.org, 1899–1908. http://dl.acm.org/citation.cfm?id=3045390.3045591

[78] Matthew D. Zeiler and Rob Fergus. 2014. *Visualizing and understanding convolutional networks*. Springer International Publishing, Cham, 818–833. https://doi.org/10.1007/978-3-319-10590-1_53