AFRL-RI-RS-TR-2021-189

PLATFORM FOR DISCOVERY OF OPTIMAL HUMAN-COMPUTER PROBLEM SOLVING ARCHITECTURE

UNIVERSITY OF WASHINGTON

NOVEMBER 2021

FINAL TECHNICAL REPORT

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

STINFO COPY

AIR FORCE RESEARCH LABORATORY
INFORMATION DIRECTORATE

■ **AIR FORCE MATERIEL COMMAND** ■ **UNITED STATES AIR FORCE** ■ **ROME, NY 13441**

NOTICE AND SIGNATURE PAGE

AFRL-RI-RS-TR-2021-189 HAS BEEN REVIEWED AND IS APPROVED FOR PUBLICATION IN ACCORDANCE WITH ASSIGNED DISTRIBUTION STATEMENT.

FOR THE CHIEF ENGINEER:

/ S /
STEVEN L. DRAGER
Work Unit Manager

/ S /
GREGORY J. HADYNSKI
Assistant Technical Advisor,
Computing & Communications Division
Information Directorate

REPORT DOCUMENTATION PAGE

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED START DATE	3. DATES COVERED END DATE
NOVEMBER 2021	FINAL TECHNICAL REPORT	NOVEMBER 2017	SEPTEMBER 2020

4. TITLE AND SUBTITLE
PLATFORM FOR DISCOVERY OF OPTIMAL HUMAN-COMPUTER PROBLEM SOLVING ARCHITECTURE

5a. CONTRACT NUMBER	5b. GRANT NUMBER	5c. PROGRAM ELEMENT NUMBER
FA8750-18-2-0271	N/A	61101E
5d. PROJECT NUMBER	**5e. TASK NUMBER**	**5f. WORK UNIT NUMBER**
TEAM	4U	WA

6. AUTHOR(S)
Popovic, Zoran; Gardner, Joshua; Mittal, Anant; Bauer, Aaron

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
University of Washington
4333 Brooklyn Ave NE Box 359472
Seattle WA 98195-9472

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Air Force Research Laboratory/RITA
525 Brooks Road
Rome NY 13441-4505

DARPA
675 North Randolph St
Arlington, VA 22203-2114

10. SPONSOR/MONITOR'S ACRONYM(S)
RI

11. SPONSOR/MONITOR'S REPORT NUMBER(S)
AFRL-RI-RS-TR-2021-189

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for Public Release; Distribution Unlimited. This report is the result of contracted fundamental research deemed exempt from public affairs security and policy review in accordance with SAF/AQR memorandum dated 10 Dec 08 and AFRL/CA policy clarification memorandum dated 16 Jan 09.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

This report describes several novel approaches broadly related to human-artificial intelligence (AI) collaborative teams in complex problem-solving environments. The research defines and implements techniques for understanding complex interactions between team members and computer agents, and distills from that analysis changes to both agents and team structures, all towards increasing the skill of individuals comprising the team and improving the overall outcomes and performance. This report describes each approach and its implementation and uses, show example results, and provide conclusions.

15. SUBJECT TERMS
Human-AI team collaboration; strategy detection; contrasting cases; imitation learning

16. SECURITY CLASSIFICATION OF: a. REPORT	b. ABSTRACT	C. THIS PAGE	17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES
U	U	U	SAR	36

19a. NAME OF RESPONSIBLE PERSON	19b. PHONE NUMBER *(Include area code)*
STEVEN L. DRAGER	**N/A**

PREVIOUS EDITION IS OBSOLETE.
STANDARD FORM 298 (REV. 5/2020)
Prescribed by ANSI Std. Z39.18

TABLE OF CONTENTS

TABLE OF FIGURES

1.0 SUMMARY

This report describes several novel approaches broadly related to human-artificial intelligence (AI) collaborative teams in complex problem-solving environments. The research defines and implements techniques for understanding complex interactions between team members and computer agents, and distills from that analysis changes to both agents and team structures, all towards increasing the skill of individuals comprising the team and improving the overall outcomes and performance. This report describes each approach and its implementation and uses, shows example results, and provides conclusions.

2.0 INTRODUCTION

This report evaluates human-AI team collaboration in complex problem-solving environments. By complex problems, this effort means a problem with no known best solution, and with no curriculum structure. In particular, a key testbed for development and evaluation are two scientific discovery environments: the Mozak environment (https://www.mozak.science/) designed towards advancing neuroscience discoveries, and Foldit, designed towards advancing biochemistry discoveries (proteomics more specifically). Both environments are composed of a mix of expert and non-expert users that produce either novel proteins or novel reconstructions of 3-Dimensional (3D) neuronal cells. As an example, in Mozak, teams annotate complex 3D volume images of neuronal cells in order to form "traces", or graphical structures representing the neuron as a set of nodes and edges in 3D space. An example of one annotation view in Mozak is shown in Figure 1.

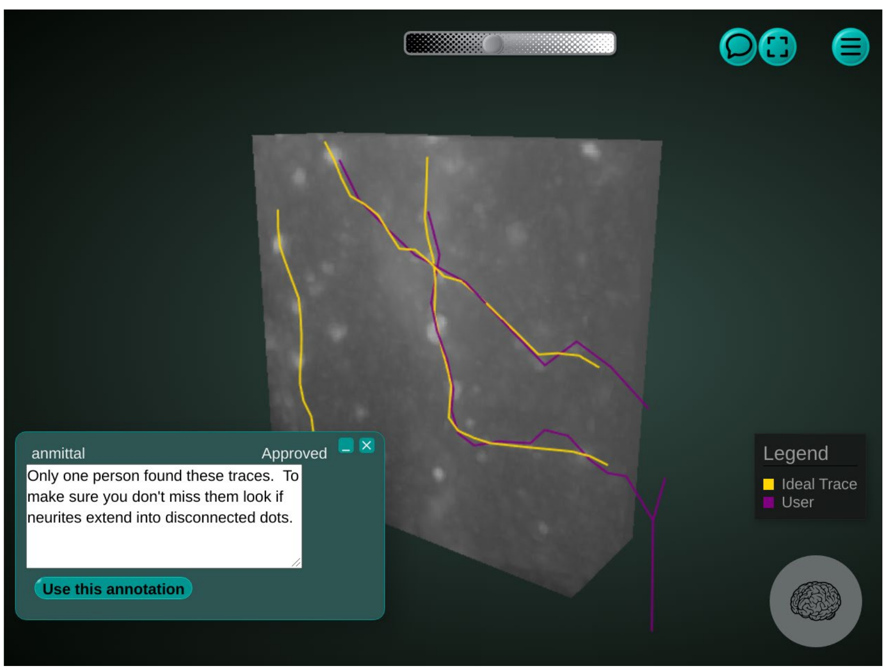

Figure 1. One View in the Mozak Environment

The testbeds present several key challenges of relevance to the goals of this project and can be roughly split into two components.

First, understanding the unknown emergent player individual and group strategies and how players respond to always changing problem states. This effort aims to automate this data-driven analysis so that structure and theory of behavior emerges from this analysis.

Second, devising intervention strategies that cause change in human behavior, and result in computer agent variations that together increase the skill level of humans and the overall quality of problem solutions. Due to the earlier termination date, the project did not fully develop automatic agent adaptation, although initial progress in that direction is described.

This research effort included analysis of game data from various Defense Advanced Research Projects Agency (DARPA) developed games, including field experiments for various test environments, but Foldit and Mozak will be the primary focus of this report and was the main environment in which this effort conducted research.

3.0 METHODS, ASSUMPTIONS, AND PROCEDURES

The research deployed several methods. Before describing them, the data available in the environments will be described. Mozak and Foldit are interactive game platforms, and all user actions are logged and available as timestamped clickstream entries with event metadata. Additionally, the raw image files containing the 3D neuronal imaging data and protein structures are available. Finally, there is also the data regarding the current state of the annotation of any particular neuron (the annotations are a set of nodes and edges, called a “trace”) and often a separate “gold-standard” version of this trace prepared by an expert annotator, which is used to evaluate the quality of traces.

The following sections will describe each of the methods used for each of the individual tasks.

3.1 Strategy Detection

The goal in strategy detection is, given a stream of player actions, to (a) provide an interpretable representation of this strategy, and (b) identify similar players. The approach in this research for strategy detection focused on adopting deep sequence models from the machine learning and reinforcement learning research communities (e.g. Sepp and Schmidhuber 2018), and applying them to sequences of player actions. Furthermore, recognizing that the problem state is also important in making sense of an action, this methodology was extended to evaluate sequences of (state, action) pairs. Since the development of a policy of action from a (state, action) pair is often called a *q-pair* in the reinforcement learning literature (Sutton and Barto 2018), this approach is referred to as “Quser Modeling”, see Figure 2.

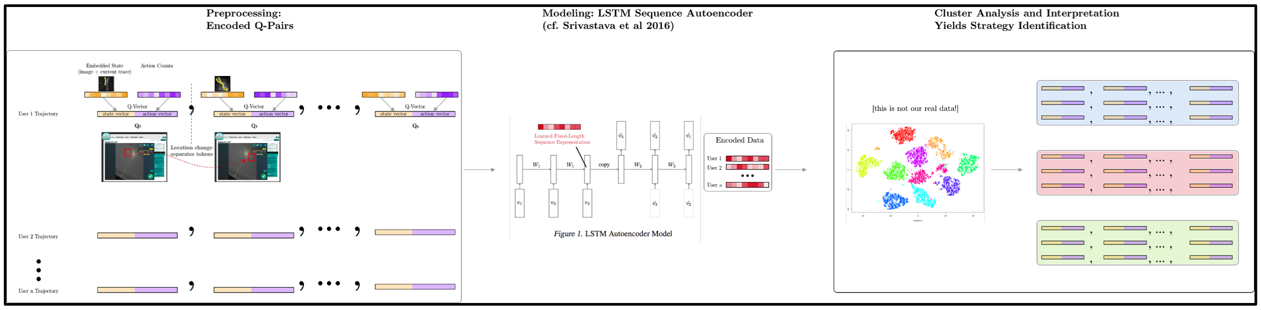

Figure 2. Quser Modeling Pipeline

The Quser model consists of two fixed encoders, which learn a "state vector" which is an encoded representation of the current problem state and represents its similarity to other problem states, and an "action vector" which represents the actions taken in response to that state, within a short fixed time window. The approach assembles a sequence of these (state, action) pairs, and applies a sequence autoencoder model which attempts to learn a single fixed-length vector representing the entire series of q-pairs for a given user. Finally, these encodings are used to group users together via clustering techniques. This allows identification of shared user strategies via inspection of the membership of each group. This also allows for the generation of recommended strategies, via the sequence model, and could support simple game-playing agents which represent a particular strategy (by drawing samples from the autoencoder model). This extends related work on modeling agents in games, e.g. (Tucker et al. 2018, Tastan and Sukthankar 2011).

3.2 Contrasting Cases for Player Improvement

The goal of this analysis is, given a game state, to identify a set of "contrasting cases" which can be used to teach the player optimal playing strategies. Contrasting cases are an idea adapted from educational science (Alfieri et al 2013), which represent two approaches to a problem -- a "positive case" which represents a correct solution to the problem (or to a similar problem), and a "negative case" which represents an incorrect solution to the problem (or a similar problem). Usually the negative case is selected such that it represents a common mistake or misconception. Contrasting cases can be generated on-the-fly from a trained model, which would allow for users to be educated in dynamic environments, and could serve as a recommendation engine of examples/non-examples of how to approach strategic situations in real-time. The use of deep learning-based models for the generation of contrasting cases is novel, to our knowledge. An example of the proposed contrasting cases pipeline is shown in Figure 3.

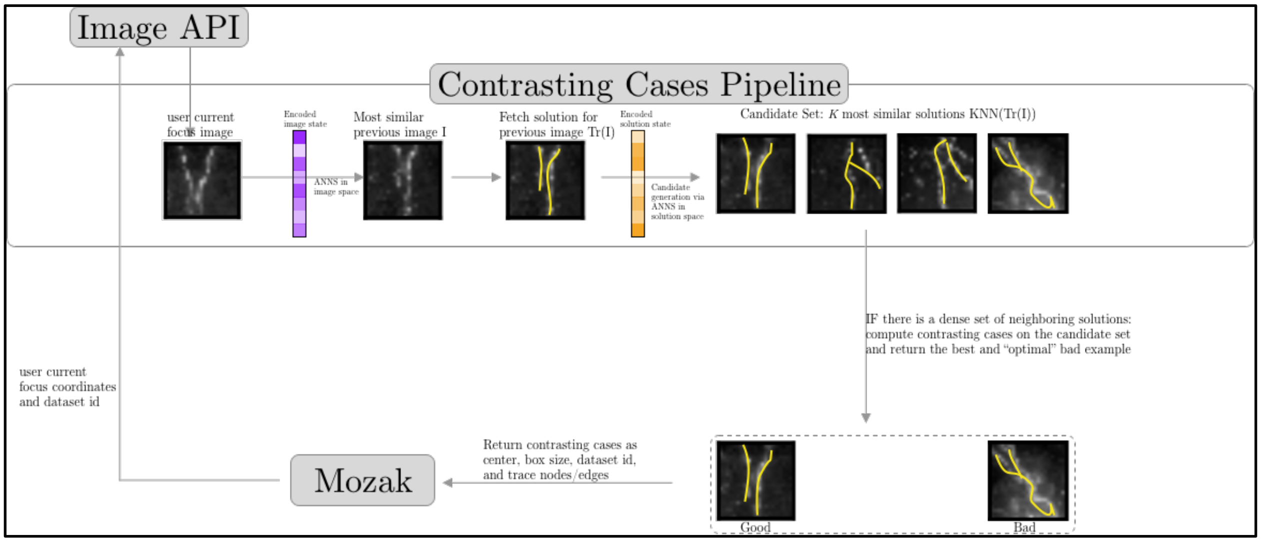

Figure 3. Contrasting Cases Pipeline

3.3 Imitation Learning from Expert Behavior

This strand is concerned with learning agents which mimic expert behavior. While prior work has often constructed models of player behavior via imitation learning or inverse reinforcement learning (Ng and Russell 200; Ziebart et al 2008), the proposed approach seeks to directly generate solutions to a given problem instance, mimicking the response an expert would provide without constructing an explicit model of the users' behavior. A learned model which effectively captures expert behavior is valuable for many reasons; one key reason it is valuable in big-data environments is that experts are scarce, and so expert behavior rarely scales. Furthermore, many tasks require the repetitive application of skills to similar scenarios, leading to duplicated efforts as similar problems are repeatedly solved; a learned model of expert behavior could allow an expert to avoid duplicated efforts and instead focus on solving the most difficult problems which an agent could not effectively solve, or on correcting the agent itself.

The proposed approach to learning models from expert behavior primarily focused on integrating approaches from deep learning and computer vision based on generative adversarial networks (GANs). In particular, the research explored two related architectures, one based on the pix2pix model (Isola et al. 2017) and another based on Flood-Filling Networks (FFN, Januszewski et al. 2016).

These models are trained on (image, annotation) pairs generated from a mix of expert and non-expert annotations in the Mozak platform.

3.4 Generating Insights from Players' Misconceptions

In Mozak, this is the "reconstruction" workflow for a neuron:

1. Publish a fresh volumetric image of a neuron (after receiving it from a neuroscience lab) on Mozak.
2. Players start tracing the neuron. Their trace data is stored as graphs with nodes and edges with their 3D coordinates in the pixel space.
3. A consensus graph is built continuously as players trace the neuron (consensus is generated if > 1 player trace nears each other in the 3D coordinate system).
4. The generated consensus graph is shared with the neuroscientists for editing and classification.
5. Final output of this process is called the gold standard which is nothing but a 3D graph with nodes and edges. This final gold standard is what the neuroscientists work with to further extend their research about how our brain works.

If one looks closely at the above workflow, there are multiple areas where there is scope for improvement. In particular, this research developed a curriculum generation end-to-end system which is intended to improve one, but most important, dimension of the “reconstruction” phase, i.e. improving player reconstruction behavior on Mozak.

With each neuron's structure being so different and varied it isn't clear what player behavior leads to less or high (more desirable) quality tracing. To tackle this problem, the research looked at it from the perspective of learning sciences - generating lessons from misconceptions. The research developed a data-driven approach to generate visual lessons or in other words insights out of (1) the players' low quality traces with respect to gold standard, and (2) neuron regions (in gold standard) where there was no consensus or almost no tracing.

These lessons/insights are annotated by experts and visually shown to the players before they begin working on a new neuron on Mozak. The players can rotate, zoom, pan, etc. the lesson to properly understand the misconception.

The entire pipeline of the lesson generation system is divided in 3 stages. In the 1st stage, the approach takes nodes from player traces and gold standard as raw data, to perform feature engineering, feature selection, normalization, and scaling. An important part of the 1st stage is to remove the nodes from player traces and gold standard which were high quality. High quality in this context means the regions where players were good at tracing. The assumption here is that low quality traces are probably due to players’ misconceptions. In the 2nd stage, the approach clusters these nodes using k-means++ (Arthur et al. 2007). The proposed approach uses k-means++ with the k-elbow method to find the optimum number of clusters. Since k-means is a centroid based clustering method, the approach then finds the 3 closest nodes (by calculating Euclidean distance) from each centroid and creating a bounding box of a variable threshold (64 was used as the threshold) for each of them. Using the (x, y, z) dimension of the bounding box, the approach plots the player traces, consensus, and the neurite in pixel space to visually understand the misconceptions. In the 3rd and final stage, experts look at the lessons to further approve or reject them. The approved lessons are annotated and further edited to remove extraneous details. It's important to note that except in the final annotation process, minimal manual intervention is required.

Lessons/insights generated and their impact will be discussed in the results and discussion section.

4.0 RESULTS AND DISCUSSION

4.1 Strategy Detection

Examples of the results of the proposed strategy detection algorithm applied to Mozak data are shown in Figure 4. The proposed algorithm is able to identify several classes of users, which are identified using a high-performance version of the Hierarchical Density-based Spatial Clustering of Applications with Noise (HDBSCAN) algorithm (McInnes et al. 2017) applied to the encoded Q-pairs described above. The proposed approach identifies several groups of users, including “superconnectors” and “superdeleters” (users who create or delete large numbers of annotations without taking other actions); “hunt and peck” annotators who selectively create or delete single annotations as a cleanup and editing strategy to refine existing solutions; “cleanup” users who strictly remove annotations in order to improve existing traces; and “power users” who employ combinations of advanced features. The proposed algorithm is also able to identify sub-strategies based on subtle situational characteristics; for example, it identifies “hunt and peck in soma” vs. “hunt and peck in noise” (lower right of left panel in Figure 4), which are a similar strategy applied in different structural parts of the neuron and require different approaches due to the unique imaging artifacts which can be introduced in these different areas.

The right-hand panel of Figure 4 also shows the hierarchical relationships which the proposed approach learns; for example, it learns a high-level “cleanup” region of the pyramid which collectively contains several lower-level editing, refinement, and improvement strategies. Other groupings are annotated in the dendrogram shown in Figure 4. Figure 5 also shows these strategies embedded in the same latent space as Figure 4 (left), but partitioned by dataset (each neuron is a different “dataset”, and thus a different problem). One may see evidence of different mixes of strategies used in different datasets. Figure 4 shows both organized groupings of coherent emergent sub-role creation, as well as the hierarchical composition of such behaviors. It is noted that, while prior work has explored techniques such as self-organizing maps (SOM) for sensemaking of player behavior (Yannakis and Togelius 2018), techniques based on Q-pairs via an autoencoder are novel, to our knowledge.

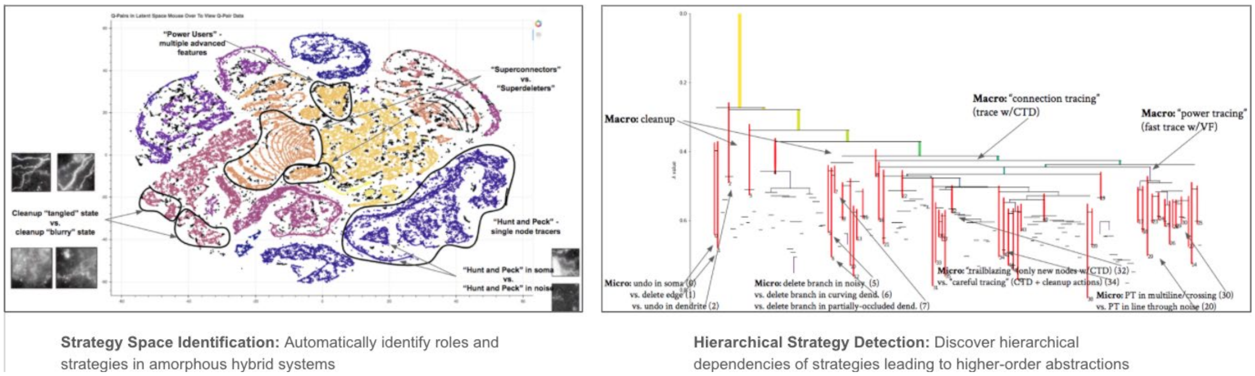

Figure 4. Strategy Detection in Mozak

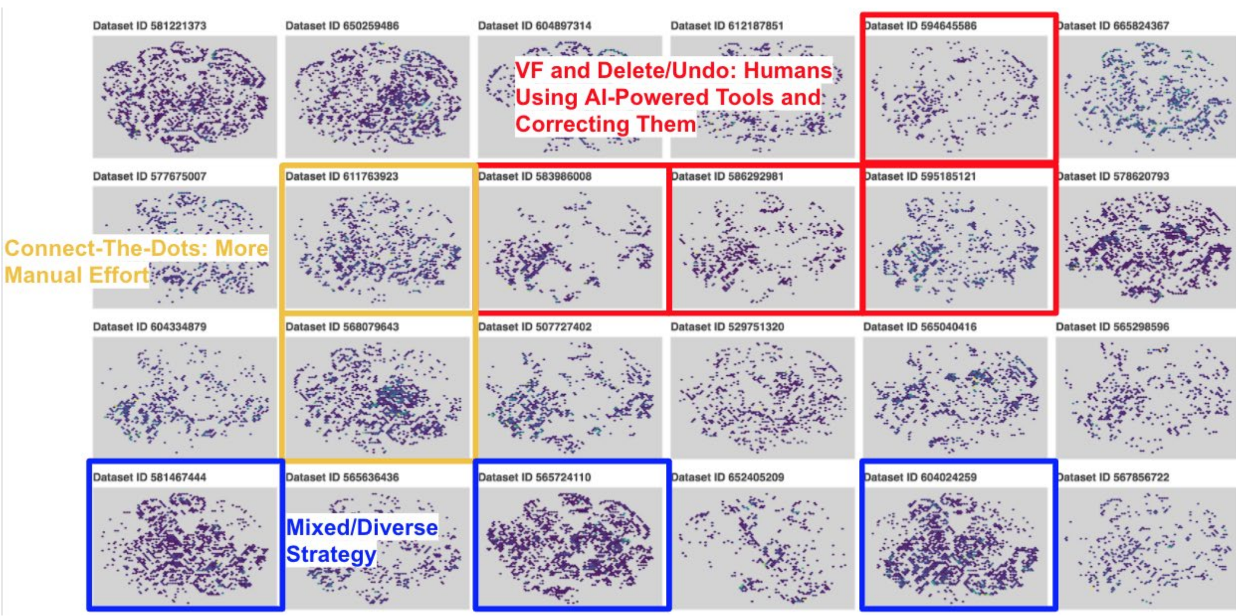

Figure 5. Comparison of Strategy Use on Different Problems

4.2 Contrasting Cases for Player Improvement

Figure 6 shows an example of the contrasting cases approach, which were developed and deployed to the live platform.

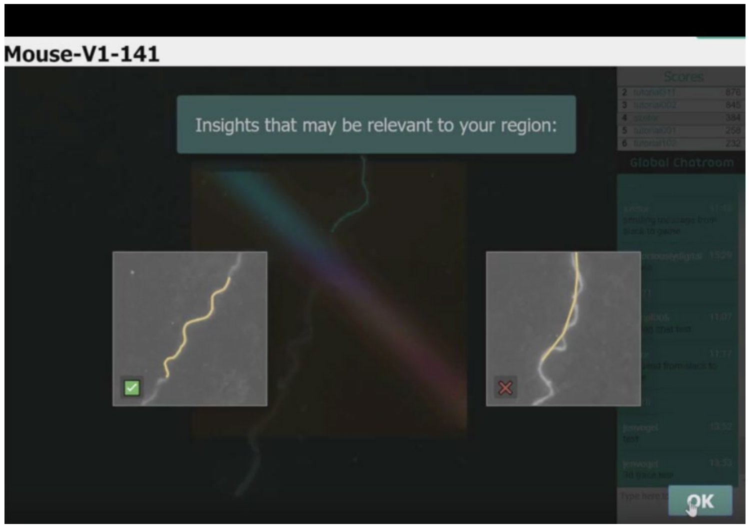

Figure 6. Examples of Contrasting Cases Generated in a Live Play Environment

This approach showed the potential to uncover useful sets of contrasting cases. For example, in the case shown in Figure 6, the left-hand case (positive case) shows a good example of an expert trace which follows a wavy neuronal fiber; the right-hand case shows an example of a common mistake on a similar case, which misses the subtle curves and traces a line that is too straight and does not capture the shape of the neuron. This methodology is likely to be useful in other complex problem-solving environments, particularly those which use visual inputs.

4.3 Imitation Learning from Expert Behavior

The research explored and extensively tested two models, described above, for learning from expert behavior; a pix2pix-based model and a FFN-based model. Results from both approaches are shown in Figure 7.

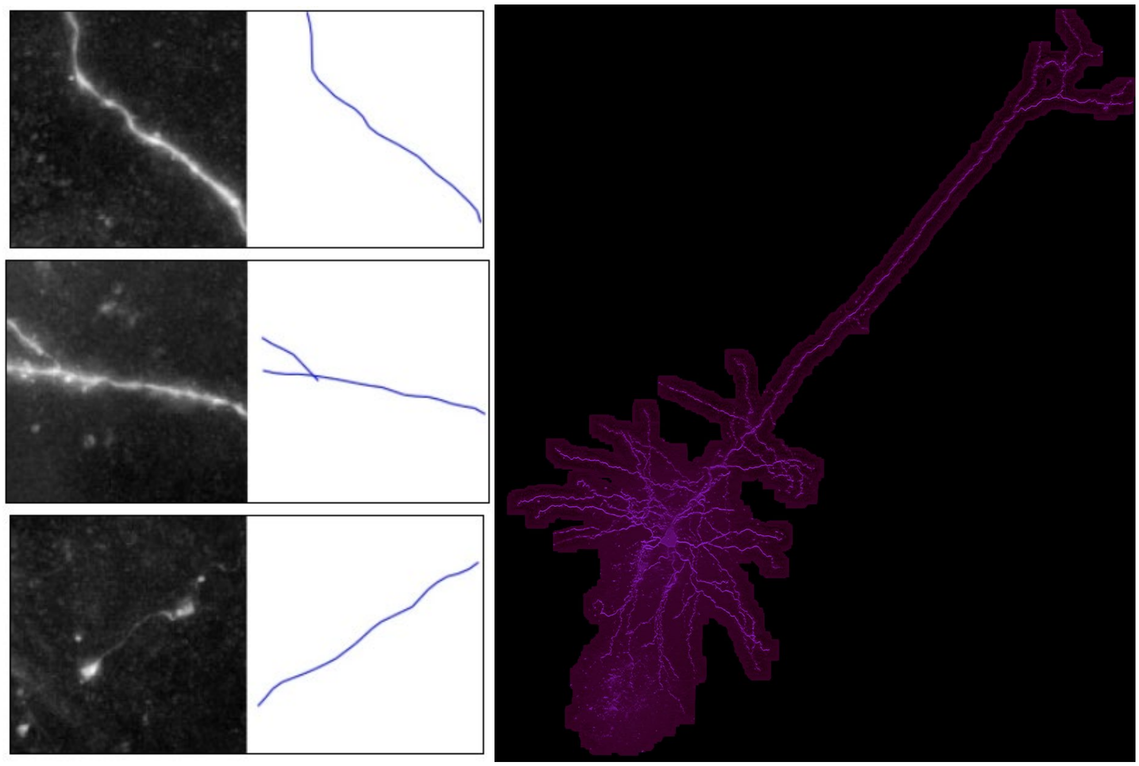

Figure 7. Automated Tracing Results from GAN-based Models. Left: Pix2Pix Right: Flood-Filling Networks (FFN)

Figure 7 shows that both approaches were promising potential techniques, and that they were able to learn high-quality automated solutions to the tracing problem. These results suggest that further refinement of these models could result in a model of sufficient quality to allow export annotators to transition from a “worker” role, where they manually generate all annotations, to an “editor” or “teacher” role, where they inspect AI-generated annotations and correct them, optionally providing feedback to the model so that it can avoid making similar mistakes in future annotations.

4.4 Generating Insights from Players’ Misconceptions

Figures 8 and 9 are examples of the generated insights/lessons with expert annotations. An important thing to note, when these lessons are shown to Mozak players, the marked traces blinkingly appear and disappear so players could understand the underlying neurite structure.

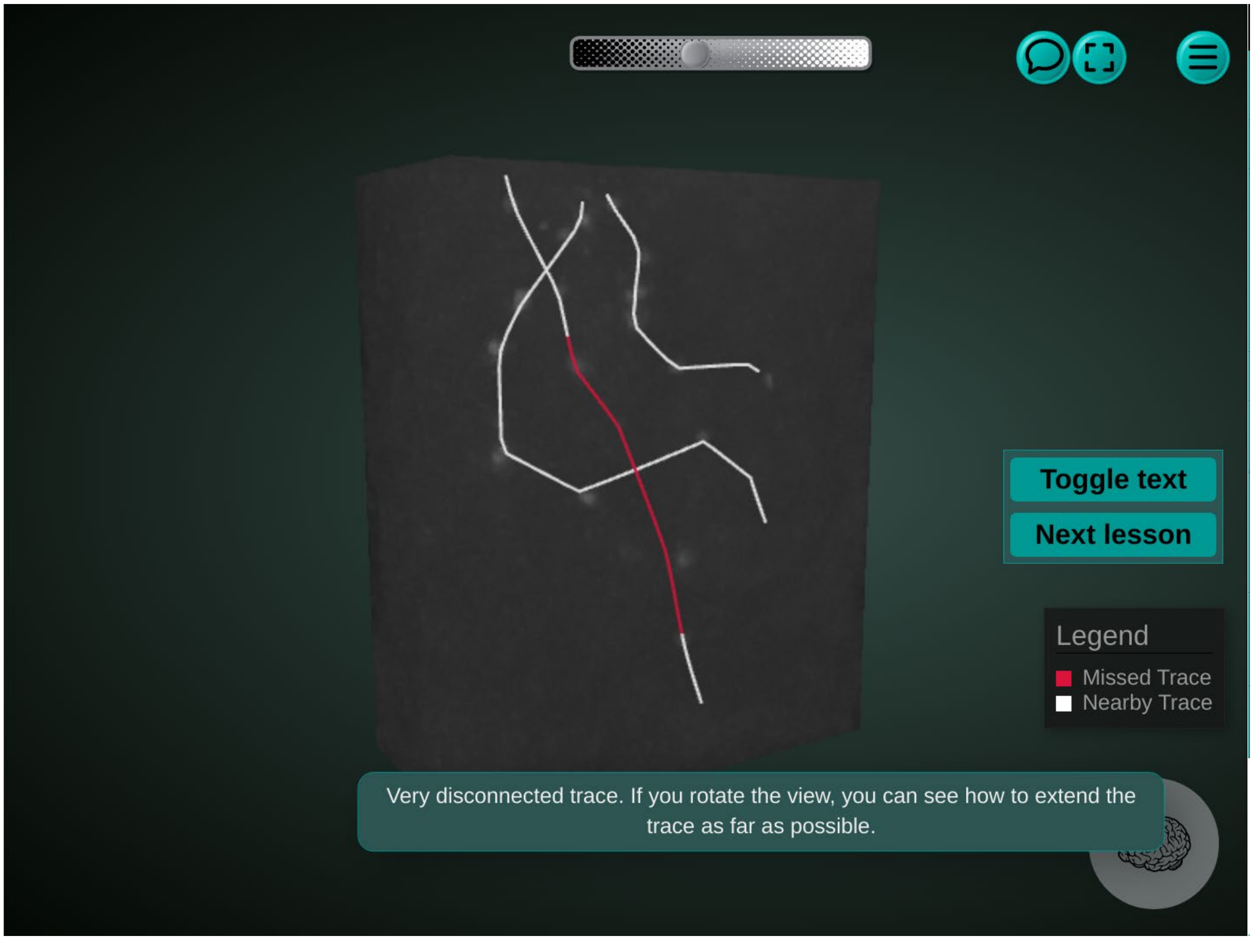

Figure 8. Insight Generated by Processing Regions of Gold Standard where Players did Low or Zero Quality Tracing

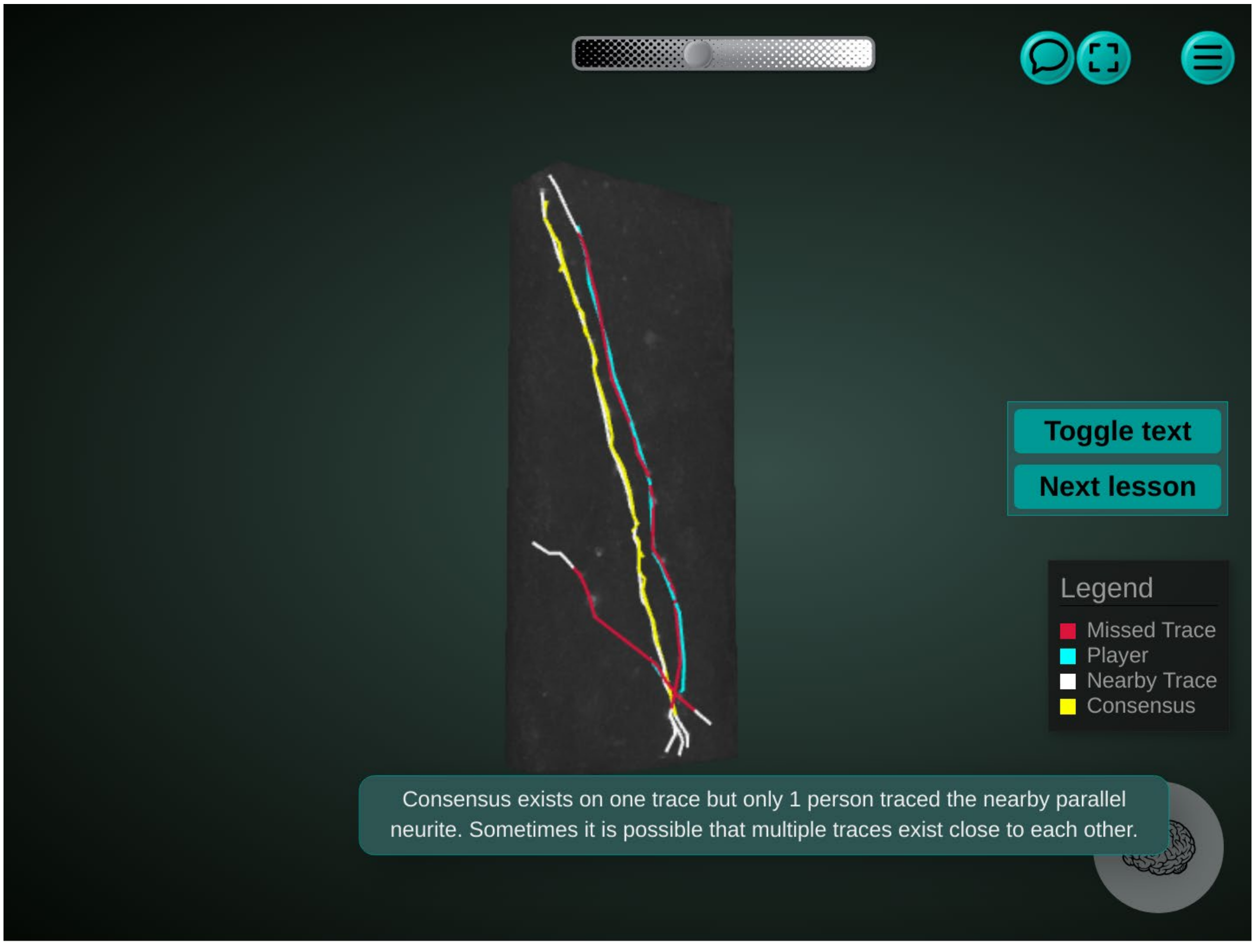

Figure 9. Insight Generated by Processing Low Quality Player Traces

To evaluate if these lessons have any behavioral change in players, a randomized controlled trial was performed. The pilot study helped to understand some key features which need to be improved on Mozak. For example, players start tracing from soma (brightest region in a neuron) and then further extend their traces. While the collective quality of traces near soma are high, their quality drastically reduces near the tips of the neurite. This is probably because the neurite isn't as bright near the tips and because the neurite breaks into blobs in those regions (unlike bright connected threads near soma). Additionally, it was noticed some players were doing high quality tracing but their traces were not leading to consensus. This was because of less work done by other players in those regions.

From these findings, further investigation, as to if there are opportunities to train players, such that they focus on regions which are harder to trace, i.e. near the tips of the neuron, may be pursued. And for the high quality tracers, perhaps increase their vote count such that their tracing directly leads to consensus.

4.5 Using Cohorts to Improve Understanding

Working toward metaplanner and diagnosis capability, the research needed to develop approaches that boost the signal of key factors contributing to success and failure. Considering the entire population of solvers en masse can dampen or obscure all but the most obvious factors (i.e., time), so subdividing the population can control for these obvious factors. Comparisons between subdivisions, or cohorts, can accelerate discovery of additional factors. The research defined four cohorts, see Figure 10, by time and performance and found that high performers have a similar range of time compared to other cohorts. The research also found a new highly salient feature: action rate. Disparities in action density appear in first hour of puzzle. This research effort verified that differences in action density are not due to parallel work (i.e., users have multiple clients open at once). The research also looked at which actions were getting used more by higher performers. The research found that higher performers make more use of actions related to constraints and big-picture reorganization suggests that they have a clearer vision of their approach. At this point, results were still too coarse for high-quality diagnosis and planner interventions. There is a compelling role for automation in exploring space of action-related features.

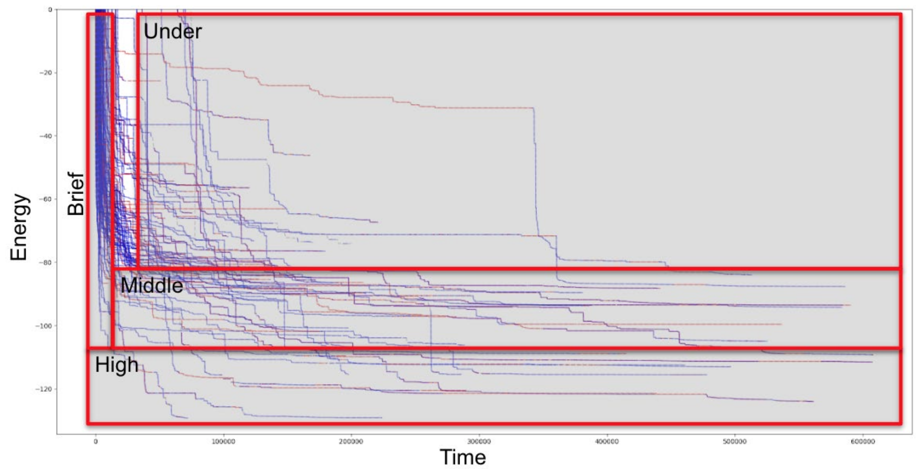

Figure 10. The Four Cohorts Defined, as Shown on the Energy vs. Time Trajectories for a Single Puzzle

4.6 Toeplitz Inverse Covariance-based Clustering (TICC) Applied to Foldit Actions

The approach constructed a time series from actions each solver took along the path to their best solution. The time series were then clustered using the TICC algorithm, varying the number of clusters. Leveraging the interpretable Markov Random Field (MRF) output TICC generates for each cluster, the relationships between actions and between cohorts were examined. Because each cluster is based on a corresponding MRF, the approach applied the betweenness-centrality measure to the graph to estimate the relative importance of each feature for each cluster. It was found that greater use of constraints relative to optimization may separate high, Figure 11, and middle cohorts, Figure 12.

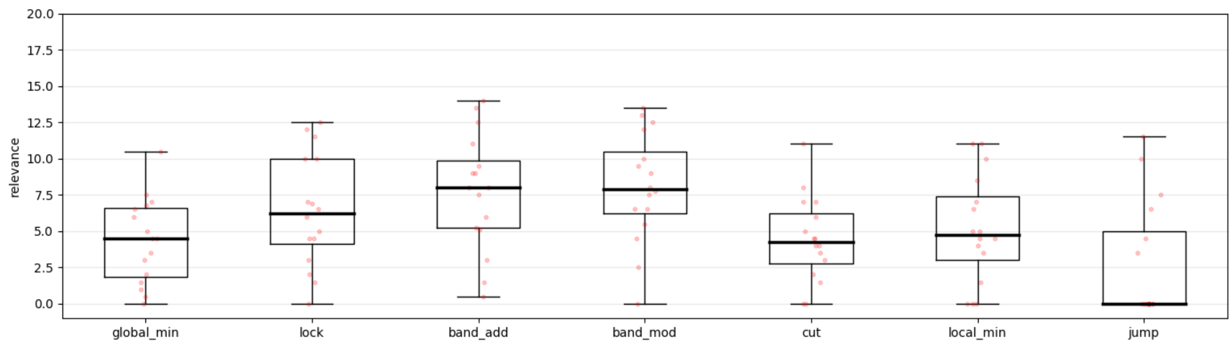

Figure 11. Relevance of Different Actions to TICC Clusters for the High Cohort

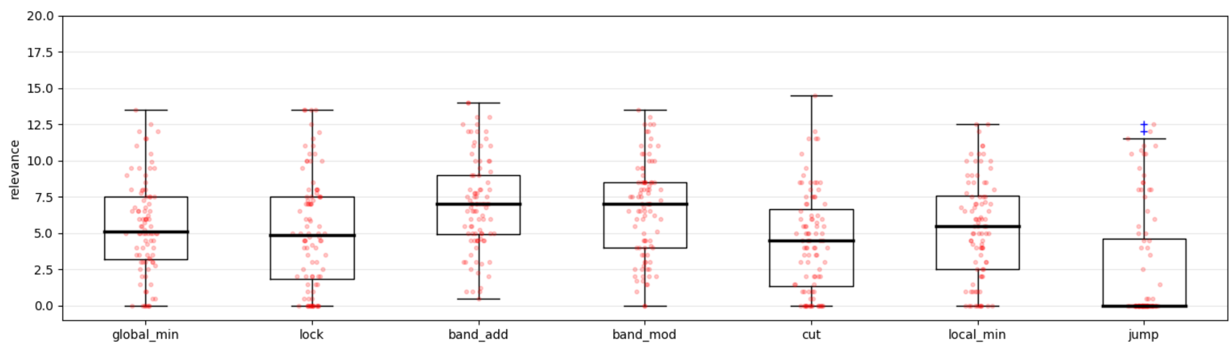

Figure 12. Relevance of Different Actions to TICC Clusters for the Middle Cohort

The research team felt that these results of clustering on centrality metrics were promising. When solvers were clustered into two groups, see Figure 13, according to the centrality metrics of their TICC clusters (using k-means), the best performers belonged to the same group (yellow) and each group has a more and less active cluster. The yellow group uses more locks and bands relative to optimization.

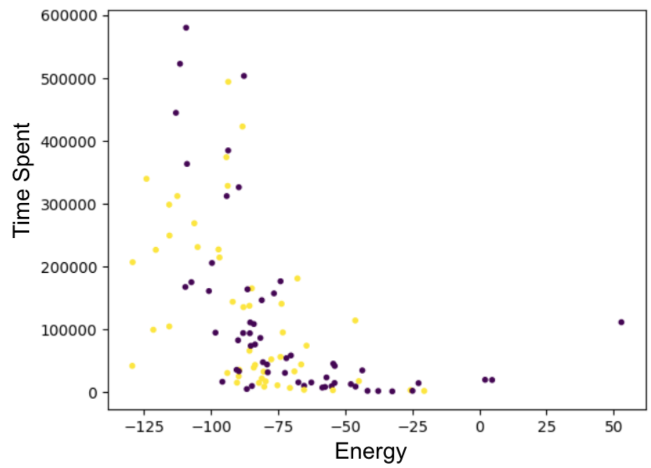

Figure 13. Time Spent vs. Energy

In October, diagnosis via pattern extraction and a proposed model discovery and system adaptation were reported. Research focus in November and December was on model discovery and adaptation. The research is conducting an evaluation of proposed network model/algorithm, contrastive "Red Zone" policy learning, and global metaplanner. The approach had a paradigm shift in optimal policy methods, as it was found that there was no value functions across entire spaces and no policies with full coverage. Instead, the research has focused on human improvement through contrasting-case exemplars, finding red zones, and best interventions in those areas. These new problems need new algorithms.

4.7 Contrastive Learning in the "Red Zone"

This is a new methodological approach in which one learns key areas of importance -- the "red zone" where expert users achieve substantially different outcomes from novices. By learning policies for action in these areas, it was found that these policies include optimal individual user actions, task assignments, and teaming.

4.8 Mozak: Meta-Planning in Complex Action Spaces

4.8.1 The "Network-Nudging" Algorithm

The research team has also been investigating the Networking-Nudging algorithm by evaluating 2 key properties: convergence and outcomes, see Figure 14. The research set to find out: Does the method learn stable network representations of the user population? How long does it take to reach convergence (if reached at all)? How sensitive is convergence to network size? Are next-action recommendations produced by this method tied to outcomes? Can recommendations improve over expert player actions? How sensitive is recommendation quality to network size?

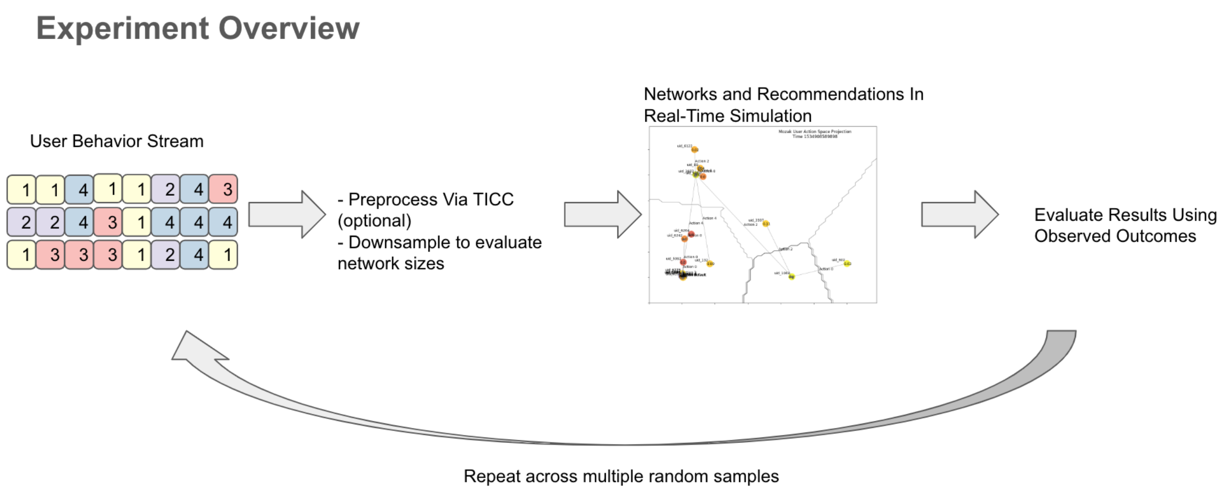

Figure 14. Experiment Overview

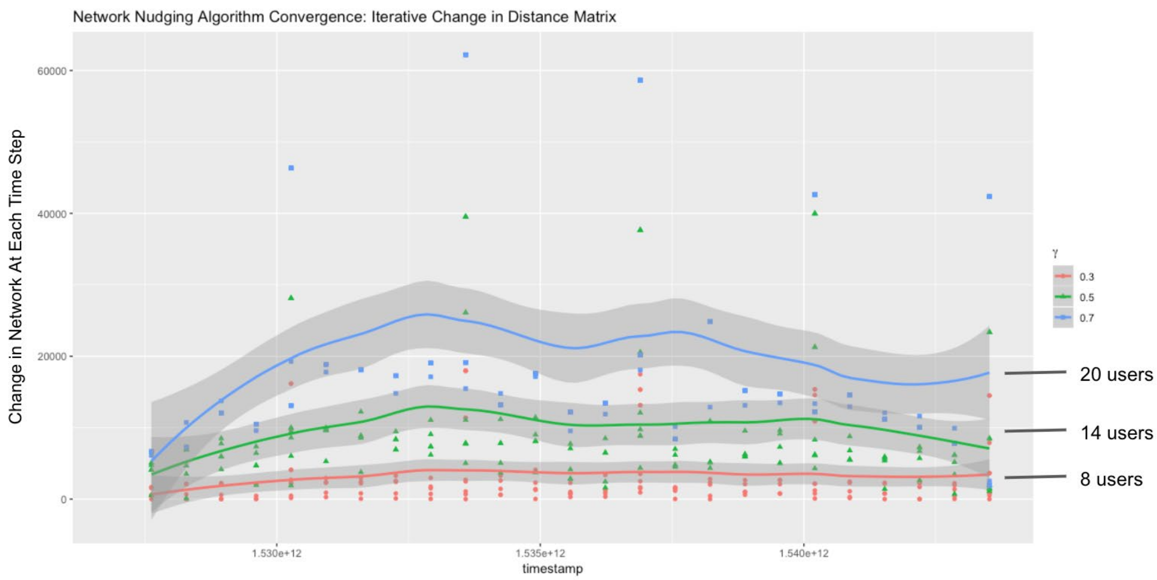

Figure 15. Results of the Networking-Nudging Experiment

Figure 15 indicates a good, stable network structure even on very small (9 users!) networks. Next, the team wanted to find out how the quality of nudges perform under low-data scenarios. The data was analyzed for the following things: Are next-action recommendations produced by this method tied to outcomes? Can recommendations improve over expert player actions? How sensitive is recommendation quality to network size?

The graph of Figure 16 depicts the results of this experiment. It was found that users who take recommended actions improve significantly more, while also finding out that adding second recommendations do not further improve outcomes.

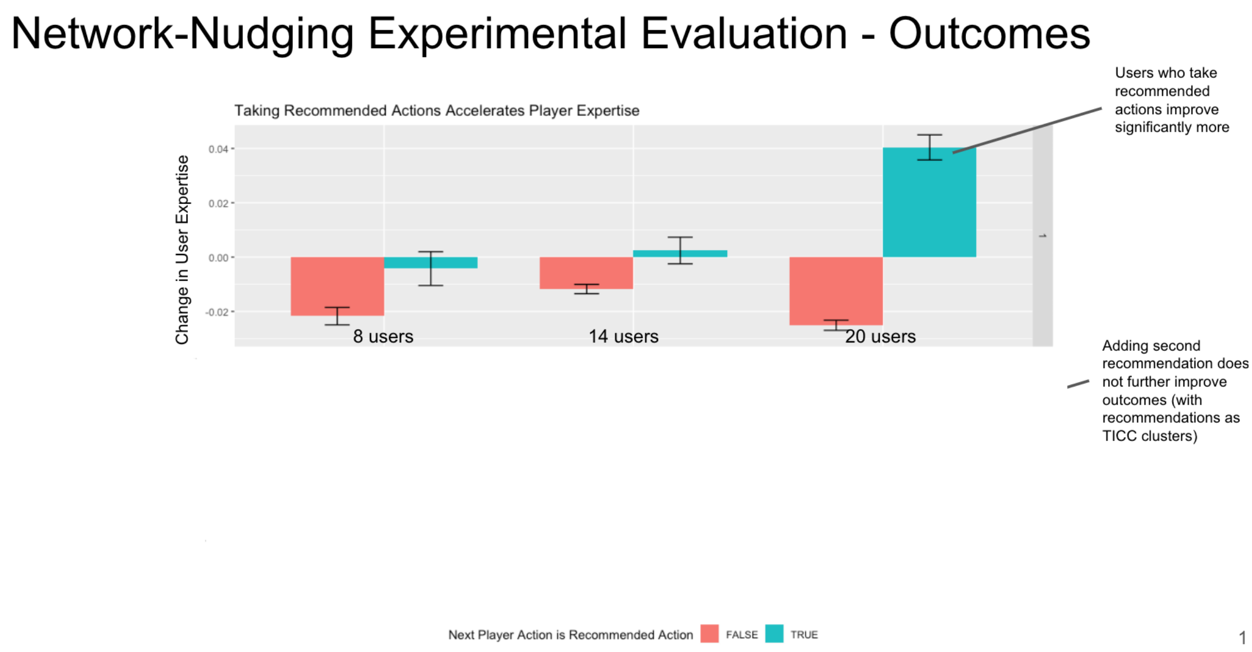

Figure 16. Network-Nudging Outcomes

4.8.2 State Space Representation for Complex Environments

Next, the research team wanted to create context-aware models, see Figure 17, to support Red Zone Contrastive Policy-Learning. This provides support for many different tasks, including: Identifying key problems for expert support (the “red zones”); characterizing expert behavior (the “contrastive policies”); supporting dynamic, data-driven decisions for teaming, task assignment, next-action recommendations; providing interpretable output for domain experts; and changing objectives.

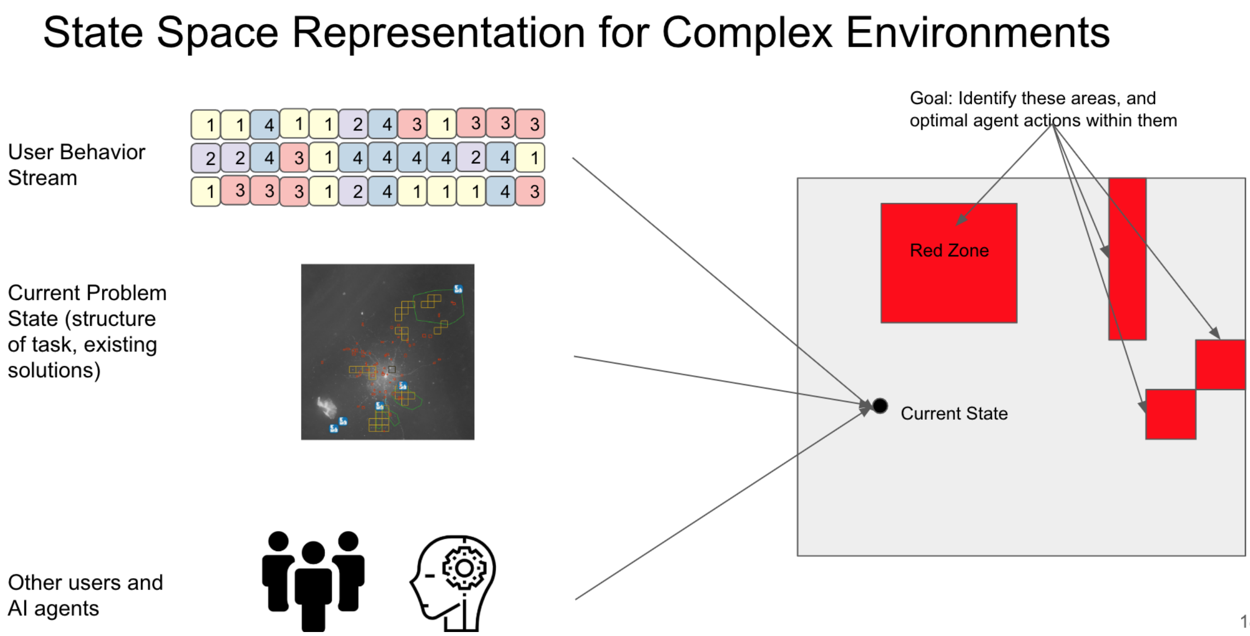

Figure 17. State Space Representation for Complex Environments

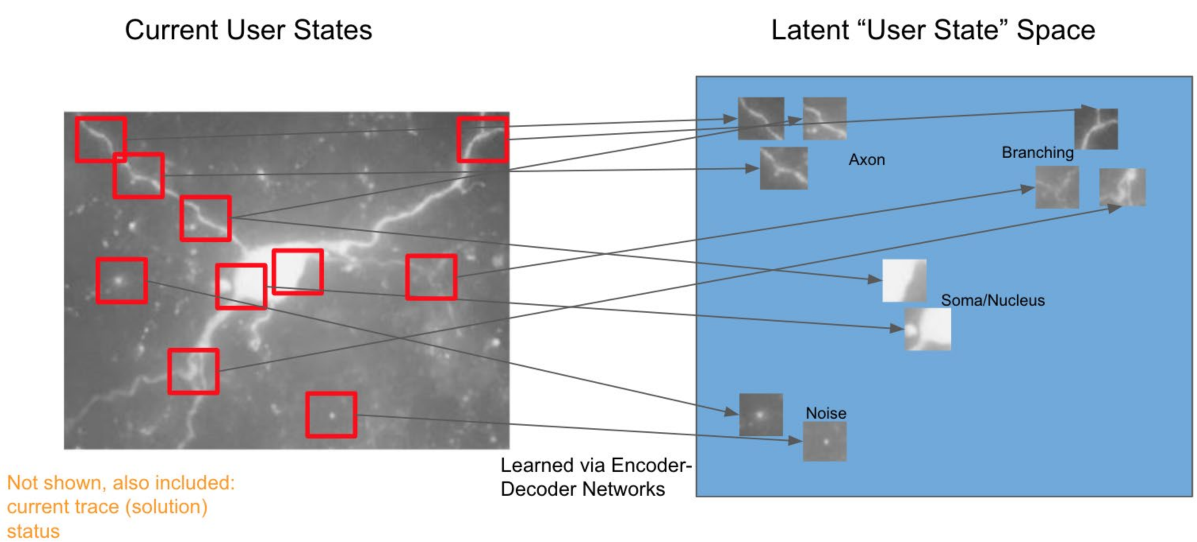

Figure 18. Current User State and Latent User State Space Example

Using the results of these experiments, as shown in Figure 18, the research team developed data-driven representations of user states (context), including images (what user currently sees), position and behavior of adversaries/teammates, and user behavior. These representations are task-specific, learned directly from data, and low-dimensional (useful for modeling). Representations of many modalities are combined as input to a metaplanner, which optimizes for a given objective.

4.9 Pattern Extraction

Previously, the research team worked on diagnosis via pattern extraction and proposed a model of discovery and system adaptation. Here the research would consider future refinements for the pattern extraction, the intent to create more sophisticated model selection which would account for similarity and the separation of high- and low-performers. The research team also looked to create richer representation of patterns which would represent attributes such as action ubiquity, duration, cadence with the goal of improving interpretability. The research team would further explore multi-level possibilities, such as sequence model at macro-level (e.g., when should a user switch patterns) and at a micro level: How many layers of subpatterns are useful. Lastly, the research team also worked on predicting performance of the approach.

First, the models were trained on differing amounts of data from 1 puzzle up to 5 puzzles, evaluated mean squared error on a test set of 2 other puzzles, examined a linear regression model (Ridge regression) and two models capable of fitting arbitrary functions Generalized Additive Model (GAM) and Gradient Boosted Regression Trees (GBRT), as shown in Figure 19. The baseline models use total time to predict performance. Pattern-based models use time spent on each pattern to predict performance. In pattern-based models, it was found that one could train the pattern extraction model, use it to predict patterns on a test set, fit performance model based on pattern times in training data, and test using predicted patterns for test data. Then, the approach searched over large sets of possible pattern-time features for the best model (different number of patterns and then different subsets of those patterns). It was found that the best GBRT model used 8 patterns from a 13-pattern model. It was also found that GAM baseline accuracy was good, with limited benefits from additional data.

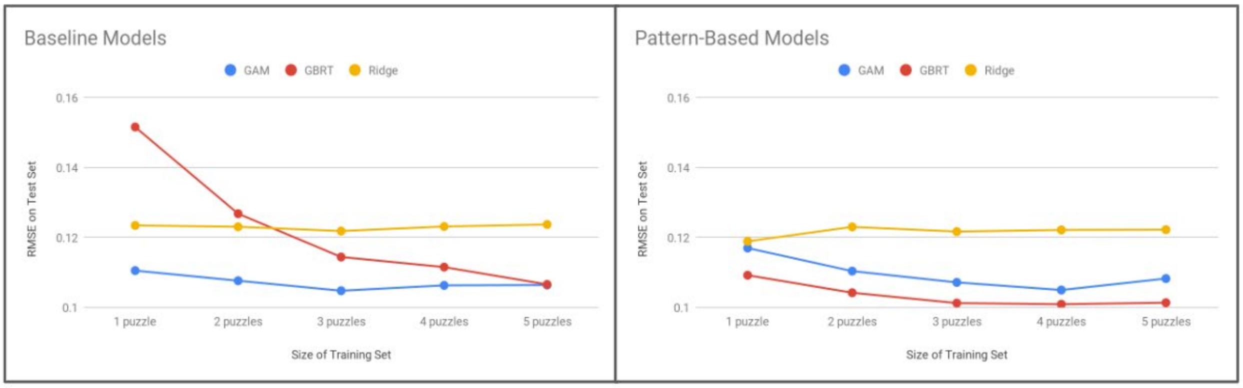

Figure 19. Baseline vs. Pattern-Based Models

4.10 Adaptation

The research team wanted to determine how Foldit users responded to environmental differences (new data, new objective, new constraints). The research attempted to identify adaptation by examining changes in behavior between same-protein puzzles but encountered a serious wrinkle: Foldit users show large variation in behavior puzzle-to-puzzle. Next, the hypothesis was tested with run-of-the-mill Foldit puzzles, which resist a standardized approach and force players to adapt. The adaptation was analyzed via efficiency of approach. The key question was: what prompts this variation? How do Foldit users benefit from these frequent changes in approach? One possibility is using prior approach fails to make good progress. If progress on a Foldit puzzle is represented as an energy frontier, and each time a new best solution is reached, the frontier is extended, one can evaluate how efficiently each player reaches each of these frontier milestones. The data was analyzed with the goal of quantifying the efficiency of a user's approach to help understand when and how adaptation is beneficial. This is also useful as an outcome distinct from final solution quality.

5.0 CONCLUSIONS

This effort's research along three main streams -- strategy detection, contrasting cases generation, and imitation learning from expert behavior -- shows that viable techniques exist for each problem in the context of complex problem-solving environments. Research developments provide a foundation for future work which leverages these techniques to build more complete systems for human-AI interaction; for example generating artificial agents which use previously-detected strategies to generate their actions, building player development flows which utilize contrasting cases for training human players; or creating human-in-the-loop flows for continual refinement of machine-learned models trained via imitation learning.

6.0 RECOMMENDATIONS

The primary conclusion of this effort is that it is possible to infer complex actions of multi-person groups joined with multiple computer agents without any a priori knowledge of the problem strategies, and structures. This has deep implications for rapidly created new problems that have no theory, no known best solutions, and sometimes even no existing experts. The research team needs to push this further to ensure broad flexibility for target problem variation. In addition, the research team continues to discover and demonstrate with implementation that it is possible to automatically and from scratch create instructions that lead individuals to improve their task-specific skillset. This means that manuals, curricula, and training can be automatically created on the fly as group problem solving strategies and behavior are discovered through analysis.

7.0 REFERENCES

Alfieri, Louis, Timothy J. Nokes-Malach, and Christian D. Schunn. "Learning through case comparisons: A meta-analytic review." Educational Psychologist 48.2 (2013): 87-113.

Hochreiter, Sepp, and Jürgen Schmidhuber. "Long short-term memory." Neural computation 9.8 (1997): 1735-1780.

Sutton, Richard S., and Andrew G. Barto. “Reinforcement learning: An introduction.” MIT press, 2018.

David Arthur and Sergei Vassilvitskii. 2007. “K-means++: the advantages of careful seeding.” In Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms (SODA '07). Society for Industrial and Applied Mathematics, USA, 1027–1035.

Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, Alexei A. Efros. “Image-to-Image Translation with Conditional Adversarial Networks.” CVPR, 2017.

Hallac, David, et al. "Toeplitz inverse covariance-based clustering of multivariate time series data." Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. 2017.

Januszewski, M., Maitin-Shepard, J., Li, P., Kornfeld, J., Denk, W., & Jain, V. (2016). “Flood-filling networks.” arXiv preprint arXiv:1611.00421.

Januszewski, Michał, Jörgen Kornfeld, Peter H. Li, Art Pope, Tim Blakely, Larry Lindsey, Jeremy Maitin-Shepard, Mike Tyka, Winfried Denk, and Viren Jain. "High-precision automated reconstruction of neurons with flood-filling networks." Nature methods 15, no. 8 (2018): 605-610.

McInnes, Leland, John Healy, and Steve Astels. "hdbscan: Hierarchical density based clustering." Journal of Open Source Software 2.11 (2017): 205.

Tucker, Aaron, Adam Gleave, and Stuart Russell. "Inverse reinforcement learning for video games." arXiv preprint arXiv:1810.10593 (2018).

Tastan, Bulent, and Gita Sukthankar. "Learning policies for first person shooter games using inverse reinforcement learning." Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment. Vol. 6. No. 1. 2011.

Ng, Andrew Y., and Stuart J. Russell. "Algorithms for inverse reinforcement learning." Icml. Vol. 1. 2000.

Yannakakis, Georgios N., and Julian Togelius. “Artificial intelligence and games.” Vol. 2. New York: Springer, 2018.

Ziebart, B. D., Maas, A. L., Bagnell, J. A., & Dey, A. K. (2008, July). “Maximum entropy inverse reinforcement learning.” In Aaai (Vol. 8, pp. 1433-1438).

8.0 APPENDICES

Our Publications

Bauer, Aaron. Understanding Problem Solving and Collaboration in Open-Ended Environments. Dissertation. https://digital.lib.washington.edu/researchworks/handle/1773/44764 (2019)

Roskams J, Popović Z. Power to the People: Addressing Big Data Challenges in Neuroscience by Creating a New Cadre of Citizen Neuroscientists. Neuron. 2016 Nov 2; 92(3):658-664. doi: 10.1016/j.neuron.2016.10.045. PMID: 27810012.

Gouwens, N.W., Sorensen, S.A., Berg, J. et al. Classification of electrophysiological and morphological neuron types in the mouse visual cortex. Nat Neurosci 22, 1182–1195 (2019). https://doi.org/10.1038/s41593-019-0417-0

Josh A. Miller, Firas Khatib, Seth Cooper, Scott Horowitz. Introducing Foldit Education Mode Nature Structural & Molecular Biology (2020).

Firas Khatib, Ambroise Desfosses, Foldit Players, Brian Koepnick, Jeff Flatten, Zoran Popović, David Baker, Seth Cooper, Irina Gutsche, Scott Horowitz. Building de novo cryo-electron microscopy structures collaboratively with citizen scientists PLOS Biology (2019).

Brian Koepnick, Jeff Flatten, Tamir Husain, Alex Ford, Daniel-Adriano Silva, Matthew J. Bick, Aaron Bauer, Gaohua Liu, Yojiro Ishida, Alexander Boykov, Roger D. Estep, Susan Kleinfelter, Toke Nørgård-Solano, Linda Wei, Foldit Players, Gaetano T. Montelione, Frank DiMaio, Zoran Popović, Firas Khatib, Seth Cooper and David Baker. De novo protein design by citizen scientists Nature (2019).

9.0 LIST OF SYMBOLS, ABBREVIATIONS, AND ACRONYMS

3D	3-Dimensional
AI	Artificial Intelligence
API	Application Programming Interface
DARPA	Defense Advanced Research Projects Agency
FFN	Flood-Filling Networks
Foldit	not an acronym
GAM	Generalized Additive Model
GANs	Generative Adversarial Networks
GBRT	Gradient Boosted Regression Trees
HDBSCAN	Hierarchical Density-based Spatial Clustering of Applications with Noise
KNN	K-Nearest Neighbors
LSTM	Long Short-Term Memory
Mozak	not an acronym
MRF	Markov Random Field
SOM	Self-Organizing Maps
TICC	Toeplitz Inverse Covariance-based Clustering