Deeply Explainable Artificial Intelligence

Profs. Trevor Darrell, Dan Klein, Pieter Abbeel, John Canny, Tom Griffiths,
Anca Dragan, UC Berkeley; Prof. Kate Saenko, BU;
Prof. Zeynep Akata, U. Amsterdam; Dr. Anthony Hoogs, Kitware

Our proposed effort toward deeply explainable AI (DEXAI) involves two key challenges: (1) generating accurate explanations of model behavior, and (2) selecting which of these explanations are most useful to a human. We will address the first challenge by creating explanation models which can be either **implicit** or **explicit**: they can implicitly present complex latent representations in understandable ways, or they can build explicit structures that are inherently understandable. Our implicit and explicit DEXAI models will create a *repertoire* of possible explanatory actions. Because these actions are generated without any model of the user (apart from providing reference explanations), we call them **reflexive**. We will address the second challenge by proposing **rational** explanations that take a model of the end-user's beliefs into account when deciding which explanatory actions to select, in a way tailored to the specific individual the system is interacting with. We will develop an explanation interface based on these innovations informed by iterative design principles.

1 Implicit and Explicit Explanations

Standard deep neural networks and other machine learning approaches are good at predicting answers or taking actions, but are typically unable to explain their decisions. To make such base models explainable we advocate and will develop XAI systems both with and without access to

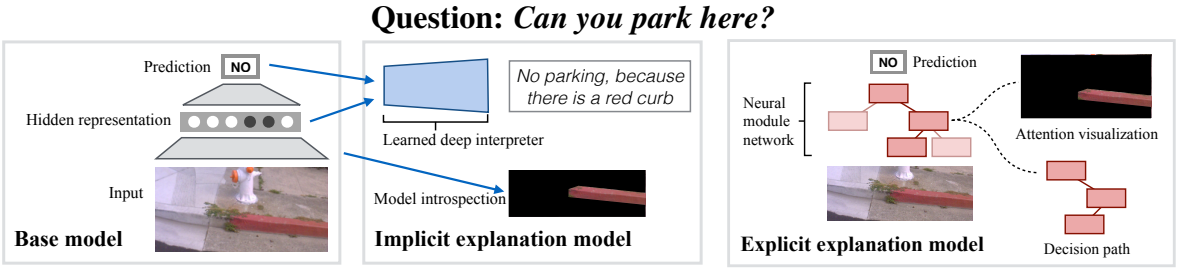


Figure 1: Example implicit and explicit explanations of a model's decision to avoid parking in the region depicted. The implicit model explains its decision by using an auxiliary interpreter model to generate a textual explanation and model introspection to extract implicit attention visualizations. The explicit model exposes interpretable intermediate states (e.g. explicit attention) and a discrete reasoning process.

structured variables: our implicit explanation approach will interpret hidden or output representations of a model and predict textual or visual explanations, while our explicit models will have interpretable components that allow one to understand their decision process (Fig. 1).

1.1 Implicit Explanations

Implicit explanation models present complex latent structures in understandable terms with the assistance of a learned interpreter model that translates states of the base model into natural textual explanations, visualizations of the evidence, or sample roll-outs (trajectory samples from the current autonomous agent policy). Modern deep neural networks (DNNs) perform reasoning directly from raw data without any hand-crafted domain knowledge by learning multiple hidden representation vectors based on end-to-end task loss(es). While these high dimensional embeddings are powerful and allow reasoning in the model, they are not immediately interpretable by humans. Our implicit DEXAI models can either work with black-box base models or can exploit access to internal hidden representations.

Generating Textual Explanations. Our near-term models will be based on generated textual explanations, which take the hidden representation and prediction of a model as input and will output natural language sentences justifying the model's decision. Textual models will learn to generate explanations from either ground truth explanations provided at training time or from captions/descriptions.

Our current textual explanation model [18] optimizes for explanatory text by explicitly providing a reward for generating text which performs well on a task-specific side task (e.g., if the goal is to generate the explanation of why a bird is classified in a certain way, we reward our model when the generated sentence has enough information to classify that bird). We have demonstrated that this method works well for generating justification explanations on a fine-grained bird species classification task. In the near term, we will explore this method on other datasets and tasks, including activity recognition and visual question answering models. Explaining actions can potentially be applied to automatic video surveillance systems where actors of suspicious actions would be automatically traced. Explaining the answer of questions would enable us to model the complex thought process required to answer vague questions, e.g. whether or not the available space is enough to safely park a car (Fig. 1).

Textual Explanations for Autonomy. In addition to providing textual explanations for vision tasks, we will generalize textual explanation to autonomy by building on the technologies discussed above. Deep neural network policies can be augmented with an extra action space to generate textual explanation for the actions being taken. In the near term, we will investigate defining a side task that encourages generated text to carry relevant information. However, since deep policies usually operate at a high frequency to output low-level actions, generated text here is expected to contain information about a sequence of future actions rather than only single-step decision like in our past work [18]. The side task will provide intrinsic reward [33, 11] for being a good explainer, which will modulate external reward. An autonomous agent that's successful at optimizing both reward functions will perform well in the specified environment, prefer temporally coherent behavior and generate textual description that's consistent with its internal "plan".

Model Introspection via Attention. We will further understand a model's decision through model introspection, which examines how different inputs cause the model to make different predictions. In the near term, we will use *activation maximization*, which backtraces the hidden decision process within a model. This can be used to recover the model's implicit attention to the most salient parts of the inputs, for example, highlighting the pixels in a bird image that are most discriminative for classifying it as a particular species. This information will be extracted from trained models via gradient backpropagation, which has been used to show that CNNs learn to localize objects despite being trained for image-level classification [40]. We will also use input dropout, which observes the neurons' role on the prediction process by masking them out [39]. We will build on these techniques to enable introspection of far more complex models, such as video captioning networks that encode frames of video and decode them into sentences using RNNs [37]. Such end-to-end neural models are often criticized for being highly non-transparent, as they provide no clear insight into the internal mapping between the pixels in a frame and the predicted words in the generated caption. In preliminary experiments, we showed that introspection techniques can uncover the implicit attention in video captioning. In Fig. 2 we show such a result, highlighting temporal segments and spatial regions that the model is using to predict each word in the caption.

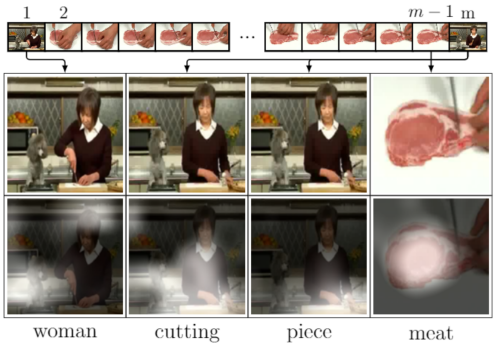


Figure 2: Model introspection for predicted video caption 'A woman is cutting a piece of meat'.

Synthetic Introspection. In the longer term, we plan to perform *synthetic introspection* by probing the network with synthetically modified input data, extending our prior efforts [25]. We will test a model's invariance to certain input properties in a controlled manner by rendering an input image without those properties and observing if similar neurons will activate. For example, if a network trained to detect objects is invariant to texture, then it will have similar top-layer activations on objects with and without texture, i.e. it will "hallucinate" the right texture when given a textureless object shape. Explanations here will include showing model activations for one example, or comparing distributions of examples, e.g., using t-SNE plots [20]. The advantage of this approach is that it is independent of the underlying machine learning model, which can be treated as a black box. This will be equally applied to analytics and autonomy.

Introspection for Autonomy via Rollouts. We will also investigate other alternatives to text-based explanations in the context of autonomy. An autonomous agent learned through reinforcement learning acquires its current behavior by trial-and-error learning and hence a reinforcement learning agent's learning process has the potential to explain the agent's current behavior. If the agent also acquires a dynamics model, the agent can also before execution illustrate anticipated behavior to the human. A human operator that's equipped with the essential knowledge of an agent's learning process and/or gets to see representative anticipated behavior will be able to have a basic understanding of an agent's behavior even before it is deployed.

As the first step towards leveraging the learning process, somewhat similar to attention models

in our supervised learning work, we seek to explain an agent's actions by tracing back the past roll-outs that strongly influence those actions: for instance, an autonomous driving agent may explain why it is following the lane by showing a training roll-out in which the agent explores not following the car lane and the decision results in a crash.

In the state-of-the-art reinforcement learning approaches [22, 30, 21, 31] it is feasible to attribute the change in policy to specific trajectories but it remains technically challenging to develop techniques that can select most relevant trajectories from the vast amount of past experience that's necessary for learning a successful DRL agent. After developing techniques for finding concise trajectories to explain a certain action, we will investigate extending those techniques to summarize all the training trajectories that an autonomous agent has experienced.

Integrating Language and Introspection. In the mid-term, we will integrate textual explanations and introspection to ground explanatory phrases with features important for a classification decision. For an explanation like "This is a cardinal because it is red with a black cheek patch" which explains a fine-grained classification, we will ground the phrase "black cheek patch" in the image. We can then compare which features are important for the network decision and for generating the textual explanation. We expect a rough alignment between features important for network decision-making and for text generation, even though we do not explicitly enforce this in our current models. However, if features do not align well, we will explore methods to align them. By aligning features, we will force generated text to accurately reflect a network's decision process.

1.2 Explicit explanations

Explicit explanation models are structured around variables with explicit semantics from which model beliefs and intentions can be transparently inferred. We propose to handle higher-level reasoning with explicit models. While implicit DNNs are irreplaceable and powerful tools for explaining low-level processing in deep models (e.g. edge structures in images or dynamic patterns in motion understanding), the phenomena they capture are likely too fine-grained to help the user understand more abstract model behaviors (e.g. reasoning processes and plans). We thus plan to couple implicit DNN representations with higher-level structured models that explicitly articulate the intended structure of computation. Examples range from broadly-applicable attention mechanisms [15, 38] to rich task-specific abstractions [3, 2], intermediate semantic attribute layers [29], and deep networks that

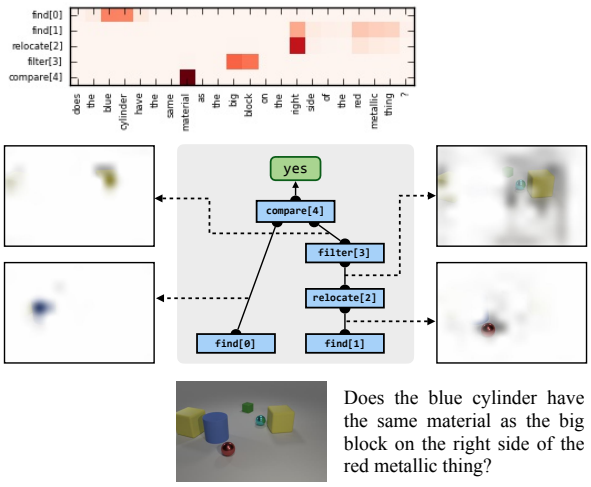


Figure 3: Neural Module Network solving a VQA task. The model automatically predicts a computation graph built from interpretable primitive components. The intermediate results of this computation have interpretable attentional semantics.

explicitly encode (end-to-end differentiable) planning procedures [35] or state estimators [16]. At their best, such models are self-interpreting: users can directly observe which parts of the input they deem relevant and which named computational primitives they choose to invoke.

A large portion of our proposed work on explicit explanation models focuses on the Neural Module Network (NMN) framework [3]. An NMN is a neural network tailored to an individual input datum (e.g. a question from a human user or an objective for an autonomous agent). NMNs are automatically constructed on the fly from a collection of *modules*, shallow network fragments that are jointly trained to be freely composable. Examples are shown in Fig. 3. NMNs may be thought of as functional programs in which each function call is implemented as a neural network. They are thus naturally interpretable: they provide an explainable model decision path via both the high-level structure and intermediate representations involved in the computation of a prediction. The discrete structure of each (input-specific) program can be easily examined by humans, and individual modules inspected or diagnosed to determine sources of model error.

Neural module networks for planning and reasoning. Following our initial success with NMNs for perception problems like visual question answering, we will extend the same machinery to similarly complex problems in autonomy. Just as the existing NMN framework provides a mechanism for assembling feed-forward networks compositionally, we expect that similar techniques can be used to assemble and reason about complex policies built from modular components. In the near-term, we plan to use modular networks to share fragments of policies across both robots and tasks. We will compose these fragments together based on directly observable aspects of the environment (e.g. the kind of robot being controlled) or high-level policy "sketches" that relate new tasks to old ones. In the mid-term, we plan to extend these techniques to focus particularly on tasks that demand reasoning over long time horizons. We will develop methods that allow us to view neural modules not only as concrete instantiations of low-level controllers, but also as high-level planning operators, providing a general mechanism for hierarchical reinforcement learning built around interpretable high-level actions.

Directly generating interpretable visual output. While our past work on modular networks has focused on problems with text or low-level controls as the output, we will extend these techniques to generate *visual* output as well. We specifically plan to generate input-conditional visual explanations by predicting where the evidence for an explanation can be found in the input. One line of near-term effort will incorporate explicit attention weight layers into visual question answering networks. Our past work [38, 15] visualized the spatial evidence, in the form of attention maps, that the network collects from the image prior to predicting the answer. This demonstrates the network's decision process at each recurrent step. We plan to extend this by also including an attention layer on the question and answer words, so we can see which words the model is using to attend to the image.

Model-based Control. For autonomy, in addition to the currently most popular model-free approaches (policy gradients and Q-learning), we will study model-based learning, where at execution time planning will be performed using the learned model. A model-based agent's behavior

is self-explaining since the human can inspect what the agent anticipates to see and achieve. In addition to studying model-based reinforcement learning for low-level, high-frequency control, we will also investigate learning with hierarchical models [34, 24, 26, 19, 12] that include high-level, abstract actions and states. Plans at the highest levels of abstraction will be forced to match up with explicit commands such as "land at location X." To make such traditional planners work with raw sensory inputs as needed for realistic settings, we can set up end-to-end differentiable computation graphs that combine such fixed, interpretable planning modules with neural nets (as done in, e.g., value iteration networks [35] and BackpropKF [16]). An RL agent capable of planning in an abstract space will then be able to present concise plans of longer duration, explaining an agent's long-term decisions in addition to short-term behavior.

2 From Reflexive to Rational Explanations

The methods discussed in the previous section provide actions that an AI system *can* take to generate explanations, in the form of language, visualization, and examples (describing the state of a computation, using policy roll-outs, etc.). An important question remaining is which explanatory actions the AI system *should* take. For explanations to be useful to humans, they cannot simply recapitulate the entire decision-making process undertaken by the AI system: they must pick out diagnostic elements relevant to the current situation and in accordance with human intuitions.

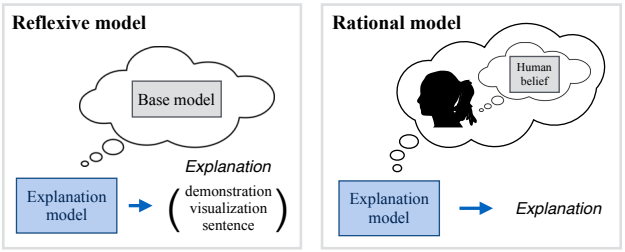


Figure 4: Reflexive and rational models. Reflexive models generate explanations directly, while rational models select explanations by reasoning about their effect on a human's beliefs. Explanations might include not only natural language descriptions, but also demonstrated behaviors, images, etc.

One way to achieve this is to learn explanations from human examples. We call agents trained in this way *reflexive*. Reflexive agents' explanations are accurate, but are not guaranteed to be most relevant or intuitive. We will thus couple the explanation algorithms described above with a learning framework inspired by our prior work on pragmatics [1], legibility of actions [13], and computational models of explanations using human judgments [23]. Our key insight is that actions (whether task-oriented or exclusively communicative) better convey information when they take into account a model of the human's beliefs. We call agents that act in this way *rational* (Fig. 4).

Modeling the Effects of Explanatory Actions. In order to support the development of rational explanatory agents, we propose to build a model of the *effects* the AI system's explanatory actions have on human beliefs, and leverage that model to optimize over the space of explanatory actions to generate those that are the most useful or informative to the human.

In our *Rational Explanations Model*, we characterize people as performing Bayesian updates over some hypothesis space taking the explanation that the system provides as evidence. Formally, let $\mathcal{H}$ denote the set of hypotheses that the human could believe in about the learned model, e.g.

the possible feature combinations the AI system might use to make its prediction (we discuss $\mathcal{H}$ in further detail below). One hypothesis $h^* \in \mathcal{H}$ is the true hypothesis, which the human does not know and the system wishes to convey. This hypothesis will be conveyed through explanatory actions e generated from the set $\mathcal{E}$, e.g. textual explanations, visualizations, policy rollouts, or alternative trajectories. Each action e has a probability $P(e|h)$ given by the observation model for the person – the probability they would assign to the system producing explanation e if hypothesis h were correct.

Given this formulation, we can characterize how the person's beliefs should change via Bayesian inference. Every explanatory action e updates the person's belief over $\mathcal{H}$ according to Bayes' rule, with $b'(h) \propto P(e|h)b(h)$ The rational DEXAI system needs to generate a sequence of explanatory actions $\mathbf{e}^* = (e_1, .., e_n)$ that best convey to the person the correct hypothesis h^*: $\mathbf{e}^* = \text{argmax}_{(e_1,..,e_n)}\, b(h^*|e_1, .., e_n)$. It is important to note that the definition of $\mathbf{e}^*$ is different from producing the explanations most probable *given* h^*. This is because b normalizes over the space of possible hypotheses, $\mathcal{H}$: producing e^* importantly takes into account what alternatives the person might believe in. Therefore, if an explanation is what would arise from the right hypothesis (e.g. the learned model that uses all the correct features to make its predictions) but also from a wrong hypothesis (e.g. an alternative model that gets some features wrong), then that explanation would not be considered as useful.

Characterizing the space of human hypotheses. If the goal is to give the user a good mental model of *how* the learner makes predictions in general, then the space of hypotheses might be different learned models that the user might assume. It is an open research question what this space might be. It will likely not be the set of all possible neural networks – it would be difficult for people to consider such a set and make inferences about it. We will start by experimenting with a hypothesis space consisting of *simpler, more interpretable* learned models, e.g. the space of all decision trees of some limited depth d, $\mathcal{DT}^d$. Actions ought to guide more probability mass onto the model closest to actual learned model M, i.e. h^* is the decision tree of depth d that best approximates the learned neural network. Here, the explanation would be of the type "I generally decide on a landing site by looking at…(these features)…".

Finally, if the goal is for the user to understand why the learner made a particular prediction $M(x)$ for a particular input x (which is different from explaining that the prediction is correct), then the space of hypotheses might be local models around that input – for instance, we would experiment with decision trees of depth d that could approximate M locally around x. This builds on [27] but uses our explanatory actions to convey these local models as opposed to assuming that they are directly interpretable. An explanation here might be. e.g. "I mistakenly decided that was the right landing site because….".

Likelihood models for explanation Having defined a hypothesis space $\mathcal{H}$, the next critical component of the model is the likelihood function $p(e|h)$ linking explanatory actions e to hypotheses h. Developing a good likelihood model requires capturing human intuitions about explanations: if h were true, what explanatory actions e would be generated?

In defining $p(e|h)$ we can draw on formal models of explanation that have been evaluated

against human cognition. We have previously explored a number of models of explanation that have been proposed in the AI literature, comparing the predictions of these models against human judgments [23]. The results provide insights about the factors that people view as relevant to forming good explanations, e.g., that a hypothesis that simply has high posterior probability given the observed data is less explanatory than a hypothesis that has high posterior probability given the observed data relative to other possible explanations. These models provide a starting point for defining $p(e|h)$ that can be refined through further experimentation. Concrete instantiations of the likelihood function $p(e|h)$ as a distribution over utterances, visuals, or controls makes it possible for an AI system to unambiguously convey its learned or intended behavior using language, visualization, or robot motion respectively.

Rational Explanation Scenarios Our model will support explaining learned policies via example roll-outs in autonomously (or semi-autonomously) behaving agents. Our formalism will enable robots to convey the policies they have learned by giving the end-user examples in simulation of what the robot would do in different scenarios. Here, the space of hypotheses $\mathcal{H}$ is the space of possible objective function the robot's policy might be optimizing, h^* would be the learned objective function, and the space of explanations $\mathcal{E}$ corresponds to policy rollouts from different starting conditions. The observation model $P(e|h)$ characterizes how likely a policy rollout would be *given* a particular objective function, and can thus be evaluated as proportional to the utility of the rollout under that objective. With this structure, a robot carefully chooses *informative* initial conditions that help the end-user best tease out the underlying learned objective.

3 Interfaces for Explanation

Our project emphasizes the importance of explanations that are interpretable and salient. Thus we will conduct a formal needs analysis and cognitive mapping for target users. We will perform contextual interviews [6] to elicit needs. Data will be collected in aggregated form to serve as the ground truth for our engineered models, and we do not plan to directly model, inquire about, or form scientific hypotheses about any individual or population in our proposed work. Contextual interviews track user behavior on realistic tasks, to better extract user behaviors that are situated in their practices [5]. We will also perform conceptual modeling with these users, extracting the concepts (roughly entities) and relations that users employ to think about their tasks. Users and experimenters will jointly construct graphical concept maps on paper for each domain [10]. Conceptual mapping is particularly important for expert users, who typically use specialized terminology that

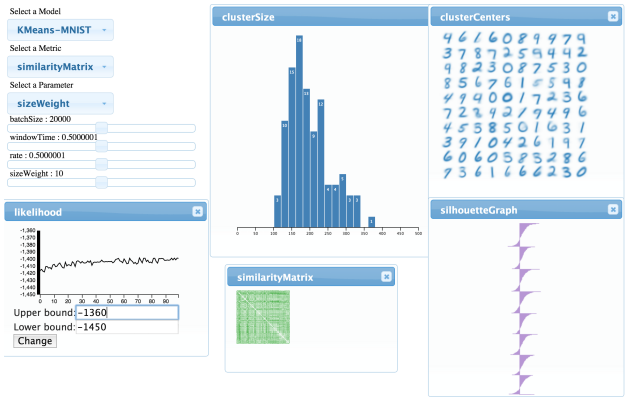


Figure 5: An interactive learning interface. Users select secondary loss functions to control and visualize their effects via matching display elements.

is inaccessible to novices. These models will then inform the construction of explanations discussed above.

Initial Interface Designs A sample implicit interface is presented in Fig. 2. The system's explanation for the generated caption is presented as an activation pattern on the image. The video described as 'A woman is cutting a piece of meat'; middle row shows temporally most important frames corresponding to the words at the bottom (arrows show positions of the frames in the video); bottom spatial heatmaps indicating significant regions for these words. Such explanations are relevant to neural systems analyzing images or video, whether or not there is an associated textual explanation.

A sample explicit interface is presented in Fig. 3. Each colored box represents a "module" or shallow network fragment and is shown with its associated intermediate result. For question answering tasks, these network layouts can be predicted based on natural language syntax and logical reasoning, and the associated modules trained end-to-end from (input, answer) pairs without direct supervision of module behaviors. Our existing experiments show that the modules acquire interpretable attentional semantics, e.g. with the `find[tie]` module implementing a tie detector for natural images and the `relate[in]` module computing containment relations.

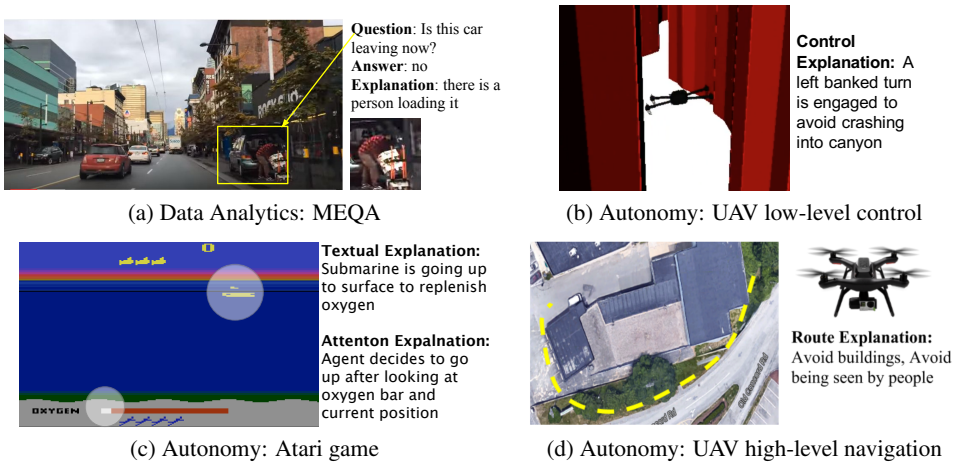
For the data analysis challenge, we will build on our previous work on *interactive machine learning interfaces* and extend it to neural models (Fig. 5). The "explanations" in our approach take the form of interactive controls of secondary loss functions, and visualization of their effects on these losses. The interaction is live during the training of a model. We are currently studying clustering tasks with a user group of statisticians, using datasets they study in their own practices.

We also propose to develop *natural language interfaces* for explanation. Human language is characterized by conversations, not just isolated utterances, and we expect that effective language-based DEXAI will require the ability to generate dialogues that allow humans to interactively refine or drill down their questions about the system's behavior. To this end, we can use our experience with machine learning models of discourse phenomena like coreference resolution [14] and content selection [4].

4 Evaluation

Our team proposes test problems for both challenge areas. They are selected in a way to evaluate our explicit and implicit explanations approaches including how useful they are to humans as discussed in the previous sections.

Data analytics For our data analytics effort, we propose to generalize the visual question answering (VQA) paradigm into an interactive dialogue multimedia event question answering (MEQA) system that enables the analyst to submit a natural language question and receive a natural language answer. The question is in the form of an event, or a query describing an event, and the system must answer it based on an archive of images and videos with associated metadata as well as audio and textual descriptions. The analyst will be able to iterate on the results and associated natural



(a) Data Analytics: MEQA

(b) Autonomy: UAV low-level control

(c) Autonomy: Atari game

(d) Autonomy: UAV high-level navigation

Figure 6: Sample challenge problems in Analytics (a) and Autonomy (b-d) efforts. UAV tasks (b) and (d) will be developed in simulation for evaluation in both Phase 1 and 2; real-world versions of these scenarios will also be developed in Phase 2 as capstone demonstrations.

language and visual explanations by asking more detailed queries referring to semantic elements within them.

We will extend a recent dataset of 100K hours of driving video data available at the Berkeley DeepDrive center; this data contains multi-media in the form of video, GPS data, and other IMU sensor data (e.g. acceleration). We will augment this data with rationales to explain driver behavior including obstacle avoidance, e.g. *"I slowed down because children are playing close to the road"* or *"I turned right because the road is blocked ahead"* and corresponding groundings, here potentially bounding boxes of the playing children or the blocked road and relevant segments of the acceleration data and steering control.

We will additionally build upon the following existing datasets: The Charades Dataset [32], the Large Scale Movie Description Challenge (LSMDC) [28], and the MovieQA dataset [36]. All these datasets contain multi-media data in form of video, audio, and textual descriptions. Some also contain dialogue and associated meta data, e.g. in form of Wikipedia articles. These datasets and existing VQA datasets and challenges do not include language explanations or ground truth localization annotations to support training and evaluation of our implicit and explicit DEXAI.

For both types of data we will generalize the VQA paradigm and collect open-ended questions and queries from annotators (see e.g. Fig. 6a), and then open answers from other humans for the multi-media data, focusing on question-answer pairs, which require one to understand multiple or all of the available modalities, so that one cannot answer the questions, e.g. with either video or text alone. Given question-answer pairs, we will collect textual rationales from another human answer provider as well as groundings (e.g. temporal segments, bounding boxes, and segmentation), which explain a given answer.

Autonomy For our autonomy effort, we will focus on demonstration-based and reinforcement-learning control of autonomous vehicles in simulated environments, involving both low-level (motor control) and high-level (route planning) tasks. Our method will then train agents with rein-

forcement learning and/or imitation learning to complete tasks in given environments and generate meaningful explanations for both low-level and high-level actions. Agents will explain their low-level actions (high-frequency control) in two ways: short-term roll-outs will be shown from a learned dynamics model to show what an agent anticipates to see, and visualization of the agent's attention of input space will demonstrate on which part of the input the agent's decision is based on. For instance, if a self-driving car starts to change lanes because it is trying to take an exit, a short-term roll-out will show the car is anticipating to take an exit and the attention visualization will indicate that the agent is focusing on the highway exit sign. Our method will also be able to generate textual rationales for an agent's high-level actions or complete trajectories, which will be relevant for a human operator to understand an agent's long-term behavior pattern. For example, an UAV agent will state it is flying back to the starting point without reaching its destination because it has discovered that the destination is indoors but all doors are locked.

In addition to evaluating our explainable agents in the context of autonomous vehicles, we also plan to use games as a richer set of environments where there will naturally be a larger vocabulary and variation in explanations. In the near term, we can use Atari games from OpenAI Gym since recent advances have made it possible to train effective policies on Atari games from raw pixels and we can directly focus on making policies explainable (Fig. 6c). In the long term, we seek to also learn explainable agents on Starcraft, where a starting point will be Berkeley Overmind [9, 8, 17, 7] (who won the inaugural AIIDE competition), and which we will also integrate into OpenAI Gym. Starcraft will provide rich environments where high-level strategy and low-level control are both important and complex, making explanations particularly useful and challenging. For our proposed domains we will collect both human demonstrations and explanations for roll-outs from either human or machine.

References

[1] Jacob Andreas and Dan Klein. Reasoning about pragmatics with neural listeners and speakers. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2016.

[2] Jacob Andreas, Marcus Rohrbach, Trevor Darrell, and Dan Klein. Learning to compose neural networks for question answering. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, 2016.

[3] Jacob Andreas, Marcus Rohrbach, Trevor Darrell, and Dan Klein. Neural module networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[4] Gabor Angeli, Percy Liang, and Dan Klein. A simple domain-independent probabilistic approach to generation. 2010.

[5] M. Aydede and P. Robbins. *The Cambridge handbook of situated cognition*. Cambridge University Press, 2009.

[6] Hugh Beyer and Karen Holzblatt. *Contextual Design: Defining Customer-Centered Systems*. Morgan Kaufmann, 1987.

[7] David Burkett, David Hall, Taylor Berg-Kickpatrick, John Blitzer, John DeNero, Haomia Huang, Eugene Ma, Yewen Pu, Jie Tang, Nihcolas Hay, Oriol Vinyals, Jason Wolfe, and Dan Klein. The berkeley overmind starcraft bot. `http://overmind.cs.berkeley.edu`. Winner of the AIIDE 2010 Starcraft Competition.

[8] David Burkett, David Leo Wright Hall, and Dan Klein. Optimal graph search with iterated graph cuts. In *Proceedings of the Conference on Artificial Intelligence (AAAI)*, 2011.

[9] MacGregor Campbell. Machine intelligence put to test in alien world. *New Scientist*, 208(2784):24–25, 2010.

[10] A. J. Cañas, R. Carff, G. Hill, M. Carvalho, M. Arguedas, and T. et al. Eskridge. Concept maps: Integrating knowledge and information visualization. In Tergan S. O. and Keller T., editors, *Knowledge and information visualization: Searching for synergies*. Springer, 2006.

[11] Nuttapong Chentanez, Andrew G Barto, and Satinder P Singh. Intrinsically motivated reinforcement learning. In *Advances in neural information processing systems*, pages 1281–1288, 2004.

[12] Peter Dayan and Geoffrey E Hinton. Feudal reinforcement learning. In *Advances in neural information processing systems*, pages 271–271. Morgan Kaufmann Publishers, 1993.

[13] Anca Dragan and Siddhartha Srinivasa. Generating legible motion. In *Robotics: Science and Systems*, June 2013.

[14] Greg Durrett and Dan Klein. Easy victories and uphill battles in coreference resolution. 2013.

[15] Akira Fukui, Dong Huk Park, Daylen Yang, Anna Rohrbach, Trevor Darrell, and Marcus Rohrbach. Multimodal compact bilinear pooling for visual question answering and visual grounding. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2016.

[16] Tuomas Haarnoja, Anurag Ajay, Sergey Levine, and Pieter Abbeel. Backprop kf: Learning discriminative deterministic state estimators. In *Neural Information Processing Systems (NIPS)*, 2016.

[17] David Leo Wright Hall, Alon Cohen, David Burkett, and Dan Klein. Faster optimal planning with partial-order pruning. In *ICAPS*, 2013.

[18] Lisa Anne Hendricks, Zeynep Akata, Marcus Rohrbach, Jeff Donahue, Bernt Schiele, and Trevor Darrell. Generating visual explanations. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2016.

[19] Leslie Pack Kaelbling and Tomás Lozano-Pérez. Hierarchical task and motion planning in the now. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 1470–1477. IEEE, 2011.

[20] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(Nov):2579–2605, 2008.

[21] Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy P Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. Asynchronous methods for deep reinforcement learning. *arXiv preprint arXiv:1602.01783*, 2016.

[22] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 2015.

[23] Michael Pacer, Joseph Williams, Xi Chen, Tania Lombrozo, and Thomas Griffiths. Evaluating computational models of explanation using human judgments. *Proceedings of the Twenty-Ninth Conference on Uncertainty in Artificial Intelligence*, 2013.

[24] Ronald Parr and Stuart Russell. Reinforcement learning with hierarchies of machines. *Advances in neural information processing systems*, pages 1043–1049, 1998.

[25] Xingchao Peng, Baochen Sun, Karim Ali, and Kate Saenko. What do deep cnns learn about objects? *ICLR Workshops*, 2015.

[26] Doina Precup, Richard S Sutton, and Satinder Singh. Theoretical results on reinforcement learning with temporally abstract options. In *European conference on machine learning*, pages 382–393. Springer, 1998.

[27] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. " why should i trust you?": Explaining the predictions of any classifier. *arXiv preprint arXiv:1602.04938*, 2016.

[28] Anna Rohrbach, Atousa Torabi, Marcus Rohrbach, Niket Tandon, Christopher Pal, Hugo Larochelle, Aaron Courville, and Bernt Schiele. Movie description. *arXiv:1605.03705*, 2016.

[29] Marcus Rohrbach. Attributes as semantic units between natural language and visual recognition. *arXiv:1604.03249*, 2016. Book chapter in preparation.

[30] John Schulman, Sergey Levine, Philipp Moritz, Michael I Jordan, and Pieter Abbeel. Trust region policy optimization. *CoRR, abs/1502.05477*, 2015.

[31] John Schulman, Philipp Moritz, Sergey Levine, Michael Jordan, and Pieter Abbeel. High-dimensional continuous control using generalized advantage estimation. In *International Conference on Learning Representations (ICLR)*, 2016.

[32] Gunnar A. Sigurdsson, Gül Varol, Xiaolong Wang, Ali Farhadi, Ivan Laptev, and Abhinav Gupta. Hollywood in homes: Crowdsourcing data collection for activity understanding. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2016.

[33] Satinder Singh, Richard L Lewis, Andrew G Barto, and Jonathan Sorg. Intrinsically motivated reinforcement learning: An evolutionary perspective. *IEEE Transactions on Autonomous Mental Development*, 2(2):70–82, 2010.

[34] Siddharth Srivastava, Eugene Fang, Lorenzo Riano, Rohan Chitnis, Stuart Russell, and Pieter Abbeel. Combined task and motion planning through an extensible planner-independent interface layer. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 639–646. IEEE, 2014.

[35] Aviv Tamar, Sergey Levine, and Pieter Abbeel. Value iteration networks. In *Neural Information Processing Systems (NIPS)*, 2016.

[36] Makarand Tapaswi, Yukun Zhu, Rainer Stiefelhagen, Antonio Torralba, Raquel Urtasun, and Sanja Fidler. Movieqa: Understanding stories in movies through question-answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[37] Subhashini Venugopalan, Marcus Rohrbach, Jeff Donahue, Raymond Mooney, Trevor Darrell, and Kate Saenko. Sequence to sequence – video to text. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2015.

[38] Huijuan Xu and Kate Saenko. Ask, attend and answer: Exploring question-guided spatial attention for visual question answering. *Proceedings of the European Conference on Computer Vision (ECCV)*, 2016.

[39] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *Computer vision–ECCV 2014*, pages 818–833. Springer International Publishing, 2014.

[40] B. Zhou, A. Khosla, Lapedriza. A., A. Oliva, and A. Torralba. Learning Deep Features for Discriminative Localization. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.