How experts explain motion planner output: a preliminary user-study to inform the design of explainable planners

Martim Brandão, Gerard Canal, Senka Krivić, Paul Luff, Amanda Coles

***Abstract*— Motion planning is a hard problem that can often overwhelm both users and designers: due to the difficulty in understanding the optimality of a solution, or reasons for a planner to fail to find any solution. Inspired by recent work in machine learning and task planning, in this paper we are guided by a vision of developing motion planners that can provide reasons for their output—thus potentially contributing to better user interfaces, debugging tools, and algorithm trustworthiness. Towards this end, we propose a preliminary taxonomy and a set of important considerations for the design of explainable motion planners, based on the analysis of a comprehensive user study of motion planning experts. We identify the kinds of things that need to be explained by motion planners ("explanation objects"), types of explanation, and several procedures required to arrive at explanations. We also elaborate on a set of qualifications and design considerations that should be taken into account when designing explainable methods. These insights contribute to bringing the vision of explainable motion planners closer to reality, and can serve as a resource for researchers and developers interested in designing such technology.**

I. Introduction

There is a growing interest in research and development of AI algorithms that are capable of generating explanations for their output [1], [2], [3]. Explanations could play an essential role in robot systems, as a way to improve predictability, user-friendliness, debugging effectiveness and overall trustworthiness of robots.

While efforts have been made to characterize explanation in general [4] and to develop explanation algorithms for task [5], [3] and path planning [6], [7], explainable motion planning has received much less attention. However, motion planners are typically black boxes that either return a robot-trajectory (e.g. a sequence of positions and joint angles to execute) or a failure message without explanation. Lay users of mobile robots may thus struggle to understand why a robot is failing to complete its task, or why it is performing it in what seems to be an inefficient way. Similarly, expert users and developers often struggle to debug planning methods, as the source of failure or trajectory properties relies on many factors—from algorithm parameters and heuristics, to inherent problem properties such as world geometry and robot kinematics. Explainable motion planners—planners that provide algorithmically-generated explanations for their output—can thus help improve user and developer understanding of these methods' outputs.

Motion planning algorithms are significantly different from task-planning algorithms due to their use of continuous sampling and optimization methods, the exhaustive use of non-linear cost and constraint functions, and various specializations to the robotics domain. Therefore, it is unclear if the kind of explanations studied in the task planning literature [3], [8] would be useful in the motion planning domain. More fundamentally, it is not even clear at the current state of research what kind of questions an explainable motion planner should be expected to answer, and what those explanations would look like.

The main idea of this paper is to use insights from experts in motion planning algorithms to obtain a preliminary taxonomy of such questions, explanations, and also methods for arriving at explanations. Due to their experience in the use or development of such algorithms, experts have in-depth knowledge of typical failures, undesired output and, importantly, *reasons* for failure and trajectory output. Experts may also have developed appropriate strategies for probing and analyzing a planner in order to identify such reasons. Therefore, we also look to elicit procedures used by experts to arrive at explanations.

This paper reports on a comprehensive user study of motion planning experts that elicits the *objects* of explanation (what things need to be explained), *types* of explanation (kinds of reasons provided), *templates* or prototypes of explanation (specific examples), and the *procedures* used by experts to arrive at explanations. The paper provides a preliminary characterization of explanations in motion planning, in the form of a taxonomy, a large set of examples, and relevant themes that one should pay attention to when developing explanation-generation algorithms in motion planning—thus considerably extending previous work [9]. This paper also paves the way for the development of explainable planners and the further refinement of the taxonomy with complementary (e.g. reflexive, lay-user-based) user studies.

The contributions of the paper are the following:

- We conduct and use a comprehensive user study of motion-planning experts to propose a preliminary taxonomy of motion planning explanations: in terms of the objects, types and procedures of explanation, as well as concrete examples of explanations;
- We gather potential issues and design considerations that should be addressed during the development of explainable motion planners, so as to produce interpretable and useful explanations.

II. Related work

This paper's goal of characterizing explanations in the context of motion planning is related to recent efforts

*This work was supported by the Air Force Office of Scientific Research (FA9550-18-1-0245), EPSRC grant THuMP (EP/R033722/1), and UKRI Trustworthy Autonomous Systems Hub (EP/V00784X/1).

All authors are with King's College London, UK.

throughout the AI community [1], [2], [3], [4]. Explainability is seen as a way to provide users, designers, and other stakeholders with tools to better understand a system's behavior, as well as to know what to do in order to obtain desired behavior [10]. Explanations have been studied from cognitive, philosophical and social perspectives [4], and they have recently been modeled in technical terms for the purpose of generating them automatically.

In the context of machine learning, several methods have been proposed to improve the understanding of neural networks [11] and general black-box classifiers [12]. In the context of task planning, methods have been proposed to automatically generate explanations for plans. For example, "model reconciliation" methods [5] assume differences between a user's mental model and a planner's model of a problem—and use these differences to generate explanations that correct the user's model. A community of eXplainable AI Planning (XAIP) has emerged from these and other efforts [8], and has led to attempts of relating similar concepts of explicability, legibility, predictability, and others [13]. We refer the interested reader to surveys of XAIP [3]. The concept of explainability has also been recently applied to the problem of single-agent [6], [14] and multi-agent path finding [7]. Few of these methods translate directly to the motion planning problem, however, since motion planning fundamentally deals with continuous instead of discrete spaces; and it is typically solved with different methods—e.g. optimization and sampling based methods instead of search.

In terms of motion planning explanations, there is currently limited work on the field. Rare exceptions include Hauser's [15] explanations for algorithm failure based on environment changes (e.g. which furniture could be moved to make a problem solvable), and Kwon et al.'s work [16] on planning motion that conveys the reason for failure. However, such works do not elaborate on what the range of possible kinds of explanations is, and which approaches could exist to generating such explanations. They also do not provide insights into the considerations and difficulties that designers should have in mind when designing algorithms to generate explanations for motion planner output. It is these crucial points that this paper will focus on. While in a previous publication [9] we have used literature review and self-reflection to characterize explanations for motion planning, the output of this paper is more comprehensive due to the use of various expert elicitation strategies. Compared to that work, here we identify new types and objects of explanation, new methods, and new design considerations not considered previously. However, [9] includes a comprehensive overview of relevant technical work and we refer the curious technically-minded reader to that publication for such insights.

III. Motion planning background

This section is an introduction to the motion planning problem and its algorithms, targeted at readers that are not familiar with the topic. Robot motion planning algorithms [17] are concerned with computing trajectories that take a robot from a start to a goal configuration, where a "configuration" typically refers to the position of the robot in the world, as well as the angles (and/or positions, velocities, torques, etc.) of its joints. These trajectories can be represented in different ways: for example, a finite sequence of configurations, or the parameters of a continuous curve. For these trajectories to be feasible, they need satisfy a certain set of constraints, such as avoiding collision with objects, respecting joint and motor limits, dynamics constraints, etc. There is also a notion of optimality associated with a trajectory—related for example to smoothness or energy consumption—and it is modeled by a "cost" function that needs to be minimized.

There are multiple approaches to solve motion planning problems. One approach is "sampling-based" and uses random sampling to build a tree of feasible configurations, until a feasible trajectory to the goal is found [18]. Optimal versions of sampling-based algorithms further refine these trees in order to asymptotically arrive at an optimal trajectory [19]. Another approach to the problem is to discretize the space of configurations, and then use graph-search algorithms to find a trajectory from start to goal where configurations take only discrete values [20]. Finally, optimization-based algorithms model motion planning as a numerical optimization problem, and use off-the-shelf optimization algorithms such as sequential quadratic programming methods to find locally-optimal solutions to the problem [21], [22], [23].

Each approach has its advantages and disadvantages. Sampling methods can be complete and optimal, thus being able to find a globally optimal solution. They struggle with narrow passages and can take potentially very high computation times. Optimization methods are relatively fast when properly initialized and can easily handle many constraints, however they rely on a good initial "guess" of what the trajectory should look like, and their output is only locally optimal [23], [21]. Search methods can be very fast at low dimensions, but need to rely heavily on heuristics at high dimensions, and the discrete nature of configurations leads to a lack of completeness and smoothness.

New machine learning-based methods are also beginning to be developed, that typically work by either trying to copy the behavior of classical motion planners in a training set of problems [24], or by trial-and-error in reinforcement-learning-based methods [25]. In this paper we focus on the classical methods above, as these are more established, and are most commonly used in production.

IV. User study methodology

A. Overview

As we have described in Section I, the goal of conducting the present user study was to characterize explanations of motion planner output, and to identify potential issues and important aspects of the development of explainable motion planners. More specifically, we were interested in answering the following research questions. **RQ1:** which events may motion planning experts want an explanation for, and what types of explanation for planner output are there? **RQ2:** which procedures do experts use to arrive at an explanation for motion planner output? **RQ3:** what issues and considerations

should developers have in mind when designing explainable motion planners?

We used the research questions above to motivate the organization of an online questionnaire targeted at motion planning experts. We specifically chose the questionnaire format for high turnout, but used a variety of elicitation approaches—from general open-ended text questions to example-driven probes—in order to allow in-depth analysis.

The questionnaire starts by asking participants' occupation, years of experience in motion planning, as well as the names of (family of) methods participants were familiar with. We excluded participants with less than 1 year of experience in motion planning.

Our questionnaire used multiple elicitation approaches targeted at answering RQ1. First it used open-ended questions that asked participants for examples of typical reasons for planner failure in their experience (i.e. failure to find a feasible trajectory), as well as reasons for obtaining unexpected motion. We specifically chose the word "reasons" in order to trigger replies that are in the form of explanations. In a second approach the questionnaire asked participants to provide examples of explanations that would be useful if automatically provided by a motion planner algorithm. The third approach tried to better understand the objects of explanation, and hence explicitly asked participants for the kinds of things that would need explanation (other than failure and unexpected motion). The last approach used realistic examples of planner output in specific planning problems, as a probe to gather specific explanations (more detail in Section IV-B).

To answer RQ2 we used direct open-ended questions that asked participants to describe the kinds of procedures and analyses they would do in order to identify the reasons for motion planner failure or unexpected motion. Finally for RQ3, we again used probes based on realistic examples of automatically generated explanations for specific problems (Section V-C), from which we gathered criticism; we also used a feedback question on the overall "explainable motion planner" enterprise; and a thematic analysis of comments raised by participants throughout the questionnaire (Section V-D). Importantly, some of the answers to sections of the questionnaire related to RQ2 and RQ3 contained insights into the characterization of explanations, and we therefore used them to complete the analysis of RQ1.

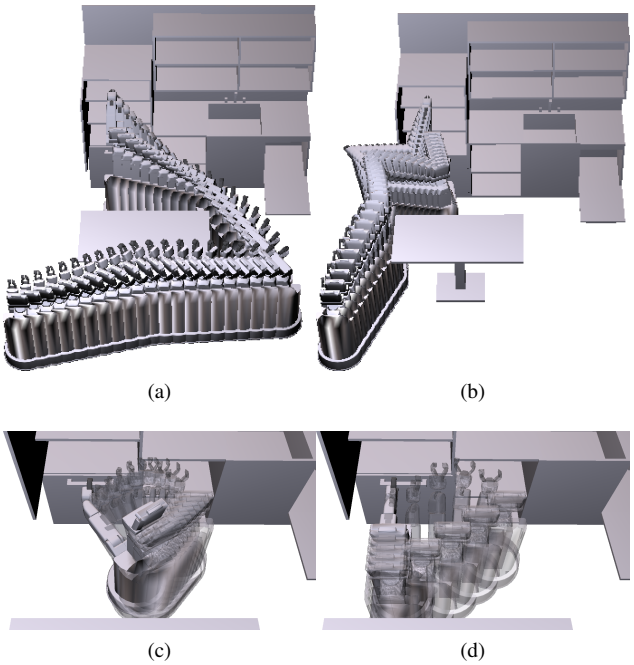

Fig. 1: Two motion planning problems: reaching for a shelf (a-b), and reaching for a washing machine handle (c-d). In each problem, the experts were asked to provide reasons why the planner would return path A (a,c) instead of path B (b,d).

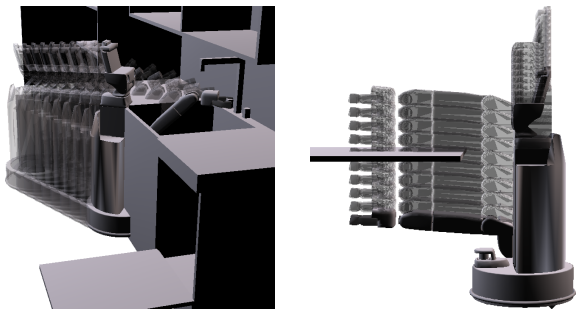

Fig. 2: Two extra problems (besides those in Fig. 1) for which example explanations were provided to experts for feedback. (a): failure due to incompatible constraints (reaching, velocity and collision). (b): failure due to an initialization that is nearby an infeasible local minimum.

B. Example-based probes

We used realistic examples of planner output to obtain specific explanations the participants would use to justify that output. All examples used a simulated model of the Toyota HSR robot [26]. First we designed two motion planning problems where two visually different paths would have similar costs, and asked users to say why the planner would return one path instead of the other. Fig. 1 shows the two problems. In one of the problems (a-b), the robot needs to avoid colliding with a table (around one side or the other) and then reach for a shelf. The path (b) is more constrained (though shorter) than (a). In the second problem (c-d), the robot needs to reach for a washing machine handle, either with a frontal path (d) or smoother curved path (c). The questionnaire described the motion planning problems ("a robot avoiding a table and reaching for a shelf", "a robot reaching for a washing machine handle on the left") and asked participants to "Assume a motion planner returns path A, but B is the actual optimal path. Why do you think the planner could return A instead of B?".

For eliciting criticism of particular explanations, we used simple methods to generate explanations for four motion planning problems. We generated explanations for both the problems in Fig. 1 (of the unexpected-motion type "why did

the planner obtain path A rather than B?"), as well as two extra problems shown in Fig. 2 (of the failure type "why did the planner fail to obtain a solution?"). The methods used to generate the explanations were *counterfactual*: they computed a modification of the original problem or the planner's parameters that would lead to the desired outcome (i.e. expected motion B and planner success respectively). We describe each explanation method together with the respective experts' criticism in Section V-C.

C. Data collection and participants

We used a snowballing strategy for gathering participants: we first sent the questionnaire personally to motion planning experts in the research network of the authors, who in turn disseminated it to their own networks. We stopped recruiting more participants once we reached saturation, i.e. the amount of new insights generated per participant became substantially low.

Out of 31 subjects that filled in the questionnaire, a total of 29 subjects had over 1-year experience and were thus allowed to participate fully (i.e. the other 2 subjects only filled in the initial background section). Fig. 3 shows the participants' occupation, years of experience in motion planning, and familiarity with four families of methods (sampling-based, optimal sampling-based, search-based, optimization-based). The figure shows that most participants were PhD, post-doc and faculty, while two were BSc and one worked in industry. The median number of years of experience was 4. Method expertise was balanced across the four families.

The families of methods participants could choose from were predefined, but we also allowed them to provide specific names of methods or software in addition to this question. Participants were familiar with varied software such as "MoveIt", "OpenRAVE", "SBPL", "STOMP", "EXOTica", "Humanoid Path Planner", among others.

V. User study results

A. Experts' explanations for planner output (RQ1)

In this section we identify the objects and types of explanation that experts provided for motion planner output (RQ1). Table I ("explanation templates" column) shows a list of all examples of explanations that we collected from participants throughout the whole questionnaire.

We categorized the explanation templates by open coding of the templates and surrounding text provided by participants, thus arriving at four "types" of explanations: method-centered explanations, problem-centered explanations, visualization-centered explanations, and unknown reasons (i.e. when no explanation can be found). Table I therefore groups the templates by type of explanation (first column).

Most method-centered explanation examples were provided by participants in the first elicitation approach to RQ1 ("what are the typical reasons for planner-failure / unexpected-motion that you encounter?"). The second elicitation approach ("what kind of planner-generated explanations would be useful?"), on the other hand, lead participants to provide many explanations that were problem-centered—referring to "how close" an object is, whether the environment is "cluttered", etc. Most visualization-centered explanations were also provided within this elicitation approach. Interestingly, this difference in responses once the focus is on the *usefulness* of explanations (i.e. difference between first and second elicitation responses), seems to indicate that problem-centeredness, interpretable abstractions (e.g. of "proximity" or "clutter") and visualizations are key factors to make explanations useful. We discuss these ideas further in Sections V-C and V-D.

The third elicitation approach to RQ1 asked participants for other type of motion planner behavior that they would like to have explanations for (other than failure and unexpected motion which had already been referred before this point). Here participants provided two extra objects of explanation: 1) computation time (i.e. why it took so long for the planner to obtain this solution); and 2) obtained cost or task-performance (i.e. why the planner did not obtain lower cost, for example because the arm was too heavy, or because a planner hyper-parameter was not well tuned). Table II summarizes the objects of explanations we have identified from the questionnaire. The last object in the table (constraint conflict) was actually inferred from RQ3-related questions. There, multiple users suggested that explanations should not only say failure happened because of a conflict between certain constraints, but also specify why those constraints conflicted with each other.

In the last elicitation approach to RQ1, where we used realistic visually-grounded explanation probes (previous described in Section IV-B), participants provided a considerable variety of explanations. Particularly, the following had not been brought up in the previous elicitation approaches: "[the planner returned trajectory A rather than B] because of the algorithm's initialization (in optimization algorithms)", "because of the chosen cost weights", "because of object x which cannot be seen from this perspective", "because of chance", "path A is lower-cost/safer", "B is out of the workspace", "the algorithm did not obtain enough samples" (in sampling-based methods). Interestingly, the diversity and novelty of explanations obtained in this elicitation approach suggests that explanations can become very specific to particular problems and applications. This observation has important consequences for the design of explainable motion planners—namely that they might require application-specific user-studies in the early stages of design, so as to better identify the kinds of explanations that will be triggered within the context at hand.

Compared to the taxonomy of [9], our elicitation strategies thus identified new categories of explanation objects ("Computation time", "Cost", and "Constraint conflict"), and explanation templates (all templates in Table I except "occupied space", "time budget", "object closeness", "conflicting constraints") that had not been considered previously.

B. Experts' methods for obtaining explanations (RQ2)

Experts that participated in the questionnaire provided multiple examples of procedures they would use to arrive at justifications for motion planner output. We categorized

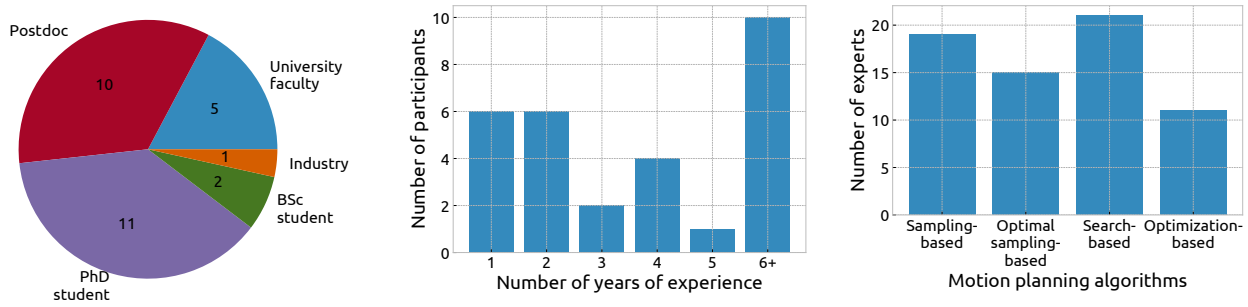

Fig. 3: Participants' occupation, years of experience, and algorithm expertise.

TABLE I: Explanation types, templates, and procedures

Explanation type	Explanation template	Explanation-generation procedure
Method-centered explanation	Because of the use of approximate collision checking	Analysis of motion
	Because of the use of gradient methods on an unsmooth problem	Analysis of motion
	Because the objective promotes solutions of following types	Compute families of solutions (e.g. local minima clusters)
	Because of the computation time budget used	Re-solving with method changes
	Because hyper-parameter x is not y	Re-solving with method changes
	Because of the algorithm's initialization scheme	Re-solving with method changes
	Because the methods got stuck at an infeasible local minimum	Analysis of motion
	Because cost weights are x not y	Re-solving with method changes
	Because of chance	Re-solving
	Because the algorithm did not obtain enough samples	Re-solving with method changes
	Because of incorrect pruning of the search tree in state x	Analysis of search tree
	Because of the choice of planner	Re-solving with other method
	Because of a software bug	Code analysis
Problem-centered explanation	Because of conflicting constraints (x conflicts with y)	Re-solving with problem changes
	Because constraint x cannot be satisfied even by itself	Re-solving with problem changes
	Because of occupied space in region x (obstacle y)	Analysis of motion / problem
	Because of free space in region x	Analysis of motion / problem
	Because the robot dynamics has effect x	Analysis of motion / problem
	Because of the volume of robot part x	Re-solving with problem changes
	Because the problem has no solution	Re-solving with problem changes; unfeasibility certification
	Because object x is not close enough	Re-solving with problem changes
	Because the start/waypoint/goal cannot be satisfied	Analysis of problem; Re-solving with problem changes
	Because the environment is cluttered	Analysis of problem
	Because of a singularity in region x	Analysis of motion / problem
	Because path A is lower-cost	Analysis of motion
	Because path A is safer/more-efficient	Analysis of motion
	Because B is out of the workspace	Analysis of motion
	Because path B crosses unmapped space	Analysis of motion
	Because of a bug-trap in region x	Analysis of function
	Because of problem difficulty	Analysis of search tree / problem
Visualization-centered explanation	Visualize explored actions and their feasibility regions	Analysis of motion / search tree
	Visualize where expected paths become infeasible	Analysis of motion
	Visualize which part/link leads to not finding a plan and where	Re-solving with problem changes
	Visualize map/plan areas that are problematic/bottlenecks	Analysis of motion / search tree / problem
	Visualize with colours why the robot moved the way it did	Analysis of motion; Re-solving with problem changes
	Visualize families of good solutions coloured by performance	Compute families of solutions (e.g. local minima clusters)
Unknown	I dont know the reasons for x	

these explanation-generation procedures into "analysis of motion" (e.g. analyzing the presence of collisions or the costs along a computed trajectory), "analysis of the problem" (e.g. analyzing the incompatibility between constraints), "re-solving" (i.e. simply solving a problem multiple times to see if the output is related to random factors), "re-solving with problem changes" (e.g. solving a problem with a relaxed or removed constraint), "re-solving with method changes" (e.g. solving the same problem with a different computation time budget, hyper-parameter value, initialization scheme, etc.), "analysis of the search tree", "code analysis" (e.g. to identify bugs), and "computing families of solutions". We added these types of procedures to Table I (column "explanation-generation procedure") according to the types of explanation these procedures could potentially be applied to.

Participants offered multiple qualifications of such procedures. For example for "analysis of the problem", one participant suggested it could involve identifying "bug-traps" (i.e. regions of the space that are hard to get out of once the planner begins exploring them), and another

TABLE II: Explanation objects (explanations of what)

Explanation object
Failure (why planner failed)
Unexpected motion (why motion A not B)
Computation time (why planner took longer than x)
Cost (why cost isn't lower)
Constraint conflict (why constraint x conflicts with y)

mentioned analysis of convergence rates for characterizing problem difficulty (for optimization-based algorithms). Other participants also suggested empirical methods of "[breaking] the problem into sub-problems that isolate different aspects", or coming up with simpler problems that re-create the issue of concern. Such procedures could then provide explanations of the type "this problem is not solvable because sub-problem x is not solvable" or "for the same reason this simple problem is not feasible". The suggestion of identifying and visualizing feasibility regions was also brought up by multiple participants.

For "re-solving" procedures, participants suggested they would try to solve slightly different problems to identify the reasons for planner output. For example, changing problem start/goal/waypoints/cost-weights until the planner output is as desired—in order to say that the output was not as desired because of the start/goal/waypoint/cost-weights in the problem. One participant also suggested it would be useful to investigate whether the problem is feasible with a single enforced constraint (e.g. only kinematics). For explanations of failure, re-solving with relaxed or removed constraints was a strategy brought up by a large proportion of the experts.

For procedures that "re-solve with method changes", participants provided multiple examples, such as solving the same problem with a different planner, initialization scheme, or hyper-parameters such as maximum number of samples (in sampling-based planners).

The methods provided for explanation of unexpected motion were similar to those of failure, except the former elicited extra procedures related to changing cost-weights, choice of planner, and visualization of solutions.

C. Issues and considerations in the design of explainable motion planners (RQ3 part 1)

In order to answer RQ3 (what kind of issues and considerations should developers have in mind when developing explainable motion planners?), we used four particular examples of explanations in realistic motion planning problems. We will call these "Problems" 1-4.

In Problem 1, shown in Fig. 1 (a-b), we assumed the use of a sampling-based planner. To the question "why did the planner obtain path (a) rather than (b)?", we automatically generated the following explanation: *"The planner obtained (a) because of a low computation time budget. It would have found (b) if the time budget had been larger (>30s)"*. We obtained this explanation by re-running the anytime sampling-based planner with a longer time-budget and stopping once a solution close to (b) was found [9]. The participants reacted to the explanation with varied criticism. Three participants commented that the explanation should specify why one path takes longer to discover (e.g. one constraint is hard to meet), since computation time is more a symptom than a cause of not finding path (b). One participant expressed concerns about whether and how we can trust the explanation to be valid. And one participant suggested the planner could provide examples of similar situations. We infer that such suggestions could help users better understand and predict patterns that lead to unexpected (sub-optimal) motion, from which users could extrapolate reasons for the longer computation time—or ideas for improving the planning algorithm.

For Problem 2, shown in Fig. 1 (c-d), we provided the explanation *"[the planner obtained path (c) rather than (d)] because the cost of (c) is lower than the cost of (d)"*. We generated this explanation by computing the difference in cost of the two motions. Participants commented that the planner should also provide information on the cost trade-offs to improve the explanation (e.g. that turning the base is cheaper than moving it). One participant also found the explanation would be useful to a new user but trivial to an expert.

For Problem 3 (Fig. 2 (a)) we assumed the use of an optimization-based solver (TrajOpt [21]). To answer the question "why did the planner fail to obtain a solution?" we re-solved the problem multiple times, each time enforcing only one of the possible subsets of constraints ("re-solving with problem changes" procedure). Then we selected the problem for which collision constraints were respected, a low number of constraints was removed, and a low distance to target pose achieved [9]. By comparing the active constraints of the feasible and original problem we generated the explanation: *"I failed because constraints on movement velocity, target location, and collision could not be simultaneously satisfied. The problem would be feasible if the constraint on velocity was removed, and the target was 0.15 meters away from the original"*. Three participants commented the explanation should specify why one constraint conflicts with the other (as discussed in Section V-A). Another participant suggested the explanation should use some kind of abstraction about the maximum reaching length of the arm (e.g. "the target exceeds arm's maximum reach by y meters"). Two participants also expressed concern regarding the degree of information the explanation should reveal. For example, since there are many unsolvable constraint sets, should the explanation only reveal one set (as in the explanation we provide), a subset, or all (i.e. all ways to make the problem feasible). Indeed, how to choose meaningful subsets of explanations without overwhelming the user could be a difficult problem in itself.

For Problem 4 (Fig. 2 (b)) we again assumed the use of an optimization-based planner. To answer the question "why did the planner fail to obtain a solution?" we re-solved the problem from multiple randomly generated initializations until the planner succeeded. We generated the following explanation: *"I failed because my algorithm relies on an initial guess of the solution (linear interpolation between start and goal), and I could not find a feasible solution locally*

around the initial guess. I would have succeeded if I had used another initialization strategy. For example, I could get the following solution using multiple random initializations". The majority of the criticism focused on the fact that such kind of explanation should not be necessary, as such functionality (to solve the problem with a different initialization) could be used as part of the algorithm and avoid failure. However, such explanation could make sense in contexts of post-hoc explanation (e.g. accident investigation or debugging). It would also make sense to use such kind of explanation as a way to determine if hard problems are solvable given enough computation time, or if they are seemingly unsolvable (e.g. "we could not find any initialization that would make this problem feasible, even after X tries"). Another participant commented the explanation seemed more suited to expert than new users.

Finally, we asked participants to reflect on and criticize the general enterprise of building explainable planners. Participants were overall receptive to a research direction towards explainable motion planning, focusing on its ability to "accelerate the debug process" and they were enthusiastic about embedding "debugging capability" to the planner itself. The focus on debugging as the main application of interest is a clear manifestation of the background of the participants (i.e. experts in the use and development of motion planners), though some participants also indicated they would imagine lay users using the technology. A few participants commented that many of these ideas would be challenging to achieve in practice. Namely they manifested concern that they would require large amounts of computation, though they suggested this could be alleviated by "drawing on planning experience and plan representations that have already been computed/learned". One participant also suggested that experience accumulated by the planner through explanation computation could be used to improve the planner over time (i.e. improve its hyper-parameters, initialization scheme, etc.). Particularly, the strategies used to identify reasons for failure could then be exploited to "try different approaches or know when to give up."

D. The themes of design consideration for explainable motion planners (RQ3 part 2)

Next, we qualitatively analyze the recurrent *themes* brought up by experts throughout the questionnaire, which can be used together with the insights above to qualify important aspects of explainable-planner development.

Visualization theme. Participants considered visualizations to be a good way of explaining failure and unexpected trajectories, with ideas like highlighting problematic areas or feasibility regions, or displaying occupancy and robot constraints. Some participants suggested it is important to find ways to "visualize the cost of the expected solution", for example by "[highlighting] in different colors why the robot had to move like it did (because of its volume, occupancies, etc.)". The focus of visualization was on making explanations interpretable and intuitive, e.g. by "showing where intuitively expected paths collide/invalidate constraints", which could be important for making sense of very abstract or complex constraints, as well as "small collisions [that] can be difficult for humans to spot".

Abstraction theme. Automatically finding problem abstractions was a functionality that was implicitly present in many of the explanations provided by participants. Explanations referring to "cluttered" environments, or an object not being "close enough" implicitly require methods for automatically recognizing such situations. Some participants suggested an explanation could rely on comparisons to a simpler problem, which again would rely on methods to identify the important features of a problem and produce new problems with similar features. The production of natural language descriptions of the environment and of objective and constraint-function behavior is also an important area of research related to this theme.

Problem hardness theme. Multiple participants commented on the need to quantify problem difficulty as a cause for some events—such as failure to find motion within a computation time budget. Participants suggested the need to use convergence rate statistics (optimization-based planners) and sample-validity statistics (sampling-based planners) as way to "measure the difficulty in solving a particular problem".

Deeper explanation theme. Another recurrent theme was the need to generate explanations that go deep in the causal process of failure, unexpected motion, constraint-conflict, etc. When referring to explanations of failure related to computation time budgets, participants expressed their belief that explanations should include reasons "why it took so long to find a solution"—rather than stating only that the method needed more time. One participant commented that "runtime is usually a symptom, not a cause of the problem", thus showing an interest of experts in deep causal explanations. One participant suggested "the explanation could say why it needs more time" by specifying which "of the constraints is hard to meet" (see problem hardness theme). Similarly regarding constraint conflicts, experts stressed explanations should say why a constraint conflicts with another, potentially through the help of visualization or abstraction. One participant even suggested explanations could help experts better understand the inner workings of motion planners, for example by helping "understand more about how the solver balances priorities of constraint satisfaction and cost minimization, and how this can cause an infeasible initial guess to converge to a much more stylistically different solution than expected".

Actionability theme. When asked for useful explanations and desired functionality of explainable planners, 7 of the 29 experts suggested that explanations should provide information that is actionable—they should provide hints about the changes that would have to take place in order to obtain the desired planner output. Participants said it would be "even better to suggest any hyper parameter changes [that] will result in better local minima", to provide "suggestions for improvement", or "examples on what to change to (possibly) make it work" (i.e. to make the planner not fail). Participants commented that such explanations could

even suggest "robot model/morphology/actuation capability changes that can improve the performance of completing a task" (i.e. when the explanation is for why the cost is not lower). And that such functionality would be an "interesting technique when combining with modularized robot design". This focus on actionability is aligned with similar trends in machine learning explainability, that have started arguing for counterfactual explanations that can empower users, e.g. to know what they need to do to get a loan [10]. Some of the actionable explanations suggested by our participants also relied on abstraction, e.g. "need the object 2 cm closer/to the left to be able to grasp it".

VI. Conclusion

In this paper we have described and analyzed a new user study designed to unpack the concept of explanations in motion planning. We used multiple elicitation approaches to construct a preliminary taxonomy of explanation objects (Table II), types, and procedures (Table I), along with specific examples and qualifications of each (Sections V-B-V-D). The data was based on expert input, thus providing deep insights into the challenges and considerations that should be paid attention to when venturing into the development of explainable motion planners in practice.

The motion-planning experts who participated in the survey agreed that explainability is an important research direction in motion planning, and were especially optimistic about its capability to improve the debugging process of motion planners. Important take-home messages from the paper include the observation that for explanations to be *useful* they will often have to be problem- and visualization-centered, and make use of intuitive abstractions. They will have to be actionable: provide hints into the required changes to make to the problem/planner in order to obtain the desired planner output. And they will have to provide *deep* explanations, in the sense that they capture causes of events at multiple levels in the causal chain (e.g. to say that a planner failed because it was not given enough time, and that it took too much time because a specific sub-problem is difficult to solve). Another observation was that concrete example-driven elicitation helps identify a large number of examples of specific explanations of planner output that were not predicted through other elicitation methods. This observation suggests that user-studies might have to be conducted with target audiences of explainable planners, in order to identify potential context-specific explanation templates.

The preliminary taxonomy and design considerations proposed in this paper can be used as a resource in the design of explainable motion planners. However, they can also be used as a basis for further research on the topic. In the future, the taxonomy should be extended based on user studies with lay users and over multiple applications of robot motion planning. Other research directions include more realistic (e.g. interactive) user study setups, and a reflexive analysis of the taxonomy.

References

[1] A. Abdul, J. Vermeulen, D. Wang, B. Y. Lim, and M. Kankanhalli, "Trends and trajectories for explainable, accountable and intelligible systems: An hci research agenda," in *ACM CHI*, 2018, pp. 1–18.

[2] A. B. Arrieta, N. Díaz-Rodríguez, J. Del Ser, A. Bennetot, S. Tabik, A. Barbado, S. García, S. Gil-López, D. Molina, R. Benjamins, *et al.*, "Explainable artificial intelligence (xai): Concepts, taxonomies, opportunities and challenges toward responsible ai," *Information Fusion*, vol. 58, pp. 82–115, 2020.

[3] T. Chakraborti, S. Sreedharan, and S. Kambhampati, "The emerging landscape of explainable automated planning & decision making," in *IJCAI*, 2020, pp. 4803–4811.

[4] T. Miller, "Explanation in artificial intelligence: Insights from the social sciences," *Artificial Intelligence*, vol. 267, 2019.

[5] T. Chakraborti, S. Sreedharan, Y. Zhang, and S. Kambhampati, "Plan explanations as model reconciliation: Moving beyond explanation as soliloquy," in *IJCAI*, 2017.

[6] M. Brandao, A. Coles, and D. Magazzeni, "Explaining path plan optimality: Fast explanation methods for navigation meshes using full and incremental inverse optimization," in *ICAPS*, 2021.

[7] S. Almagor and M. Lahijanian, "Explainable multi agent path finding," in *AAMAS*, 2020.

[8] M. Fox, D. Long, and D. Magazzeni, "Explainable planning," *arXiv preprint arXiv:1709.10256*, 2017.

[9] M. Brandao, G. Canal, S. Krivic, and D. Magazzeni, "Towards providing explanations for robot motion planning," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2021.

[10] S. Wachter, B. Mittelstadt, and C. Russell, "Counterfactual explanations without opening the black box: Automated decisions and the gdpr," *Harvard Journal of Law & Technology*, vol. 31, 2017.

[11] C. Rudin, "Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead," *Nature Machine Intelligence*, vol. 1, no. 5, pp. 206–215, 2019.

[12] M. T. Ribeiro, S. Singh, and C. Guestrin, "'why should i trust you?' explaining the predictions of any classifier," in *KDD*, 2016.

[13] T. Chakraborti, A. Kulkarni, S. Sreedharan, D. E. Smith, and S. Kambhampati, "Explicability? legibility? predictability? transparency? privacy? security? the emerging landscape of interpretable agent behavior," in *ICAPS*, 2019, pp. 86–96.

[14] M. Brandao and D. Magazzeni, "Explaining plans at scale: scalable path planning explanations in navigation meshes using inverse optimization," in *IJCAI 2020 XAI Workshop*, 2020.

[15] K. Hauser, "The minimum constraint removal problem with three robotics applications," *IJRR*, vol. 33, no. 1, pp. 5–17, 2014.

[16] M. Kwon, S. H. Huang, and A. D. Dragan, "Expressing robot incapability," in *ACM/IEEE HRI*, 2018, pp. 87–95.

[17] S. M. LaValle, *Planning algorithms*. Cambridge university press, 2006.

[18] S. M. Lavalle, J. J. Kuffner, and Jr., "Rapidly-exploring random trees: Progress and prospects," in *Algorithmic and Computational Robotics: New Directions*, 2000, pp. 293–308.

[19] S. Karaman and E. Frazzoli, "Sampling-based algorithms for optimal motion planning," *IJRR*, vol. 30, no. 7, pp. 846–894, 2011.

[20] M. Likhachev, G. J. Gordon, and S. Thrun, "Ara*: Anytime a* with provable bounds on sub-optimality," in *NeurIPS*, 2003, pp. 767–774.

[21] J. Schulman, Y. Duan, J. Ho, A. Lee, I. Awwal, H. Bradlow, J. Pan, S. Patil, K. Goldberg, and P. Abbeel, "Motion planning with sequential convex optimization and convex collision checking," *IJRR*, vol. 33, no. 9, pp. 1251–1270, 2014.

[22] M. Zucker, N. Ratliff, A. D. Dragan, M. Pivtoraiko, M. Klingensmith, C. M. Dellin, J. A. Bagnell, and S. S. Srinivasa, "Chomp: Covariant hamiltonian optimization for motion planning," *IJRR*, vol. 32, no. 9-10, pp. 1164–1193, 2013.

[23] M. Brandao, K. Hashimoto, and A. Takanishi, "Sgd for robot motion? the effectiveness of stochastic optimization on a new benchmark for biped locomotion tasks," in *IEEE-RAS Humanoids*, 2017.

[24] A. H. Qureshi, A. Simeonov, M. J. Bency, and M. C. Yip, "Motion planning networks," in *IEEE ICRA*, 2019, pp. 2118–2124.

[25] S. Levine, C. Finn, T. Darrell, and P. Abbeel, "End-to-end training of deep visuomotor policies," *JMLR*, vol. 17, no. 1, pp. 1334–1373, 2016.

[26] Toyota hsr robot description. [Online]. Available: https://github.com/ToyotaResearchInstitute/hsr_description