Towards providing explanations for robot motion planning

Martim Brandão, Gerard Canal, Senka Krivić, Daniele Magazzeni

***Abstract*— Recent research in AI ethics has put forth explainability as an essential principle for AI algorithms. However, it is still unclear how this is to be implemented in practice for specific classes of algorithms—such as motion planners. In this paper we unpack the concept of explanation in the context of motion planning, introducing a new taxonomy of kinds and purposes of explanations in this context. We focus not only on explanations of failure (previously addressed in motion planning literature) but also on contrastive explanations—which explain why a trajectory A was returned by a planner, instead of a different trajectory B expected by the user. We develop two explainable motion planners, one based on optimization, the other on sampling, which are capable of answering failure and constrastive questions. We use simulation experiments and a user study to motivate a technical and social research agenda.**

I. Introduction

Motion planners are traditionally not self-explanatory about their output. The result of running a motion planner is typically either a trajectory or a failure notice, so users may have problems understanding why a planner failed or why a trajectory is different from what was expected. Typical motion planner output can thus be hard to understand, debug, and trust. Automatically-generated explanations for planner output may offer a way to alleviate this issue: by increasing users', developers', and stakeholders' understanding of planners and planning problems.

Notions of explanation in the existing motion planning literature are narrow. For example, research has focused on planner failure ("Why did you fail?") [1], [2] but not on answering trajectory-constrastive questions ("Why is the output trajectory A, rather than B which I expected?"). However, the latter form of contrastive question is most relevant to humans, as evidenced by work on the psychology of explanations [3], and it is also most important for human empowerment [4] and calibrating trust. Work in motion planning explanations has also focused on sampling-based [1], [5] but not optimization or search methods; and focused on environment design [1] (e.g. which furniture could be moved to make a problem solvable). However, as we will show, other important applications exist.

Motivated by these gaps, in this paper we explore the question of what the concept of explanation could mean and be useful for in the context of motion planning. We claim that we can use explanations as a unified way to algorithm design, mechanical design, environment design, human-guided planning, and calibrating trust. Based on a new taxonomy of the kinds of explanations related to motion planners, we propose and evaluate two explainable motion planning methods. Finally, we elaborate on how the study of explanation-computation should be informed by research on related problems in motion planning [1], [5], [6], optimization [7], visualization, communication [2], psychology and philosophy of explanation [3], as well as recent efforts for explainable machine learning [4] and explainable AI planning [8]. Our contributions are:

- We introduce a new taxonomy of explanations in the context of motion planning, and extend the concept to constrastive explanations and clarifications (Section III);
- We propose methods for generating explanations (Section IV) and evaluate them on a user study (Section V);
- We elaborate on a comprehensive research agenda for explainable motion planning (Section VI).

We provide related work throughout the whole paper.

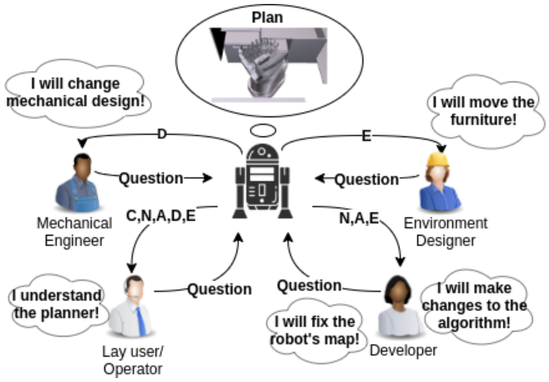

Plan
I will change mechanical design!
D
E
I will move the furniture!
Question
Question
Mechanical Engineer
C,N,A,D,E
N,A,E
Environment Designer
I understand the planner!
Question
Question
I will make changes to the algorithm!
I will fix the robot's map!
Lay user/ Operator
Developer

Fig. 1: Users may have different questions regarding motion planner output: "Why did you fail?", "Why trajectory A rather than B that I expected?". The planner should provide explanations based on various factors: **C**ost, constrai**N**ts, **A**lgorithm parameters, robot **D**esign, or **E**nvironment. These then serve multiple purposes (bubbles).

II. Preliminaries

Motion planning is concerned with finding trajectories of a system that respect certain start, goal, feasibility and other constraints. In the context of this paper, a trajectory $\xi(t) : \mathcal{T} \rightarrow \mathcal{Q}$ maps time $t \in \mathcal{T}$ to a robot configuration $q \in \mathcal{Q}$. A configuration typically includes a combination of joint angles, link pose, torques, or contact forces. Depending on the problem, time is an integer $\mathcal{T} = \mathbb{Z}$ or real number $\mathcal{T} = \mathbb{R}$. Trajectories lie in a space Ξ. A motion planning problem requires obtaining a trajectory $\xi(t)$ that starts at a configuration $q_s \in \mathcal{Q}$ and ends at a goal q_g in a goal set $\mathcal{G} \subset \mathcal{Q}$. It further requires ξ to minimize a cost $C[\xi] : \Xi \rightarrow \mathbb{R}_+$ and satisfy a set $\mathcal{H}$ of constraints $H_i[\xi] \leq 0$ (note that equality constraints can be modeled with two

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

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

inequalities). We compactly write a motion planning problem as $\min_\xi C[\xi]$ s.t. $\mathcal{H}$. This definition of a motion planning problem includes both 2/3D path planning, manipulation planning and loco-manipulation planning.

We denote by a motion planner any algorithm that can satisfy the above definitions. This includes optimal search-based (e.g. A* [9]), optimal sampling-based (e.g. RRT* [10]) and optimization-based planners (e.g. Trajopt [11]).

III. Taxonomy of motion planning explanations

A. Kinds of questions

Following the work of [3] we define explanations as answers to why questions. In this paper we focus on two kinds of questions: failure questions ("Why did you fail?") and trajectory-contrastive questions ("Why did you obtain trajectory A rather than B?").

1) Failure questions: Motion planners can fail for many reasons, such as reaching planning time limit (e.g. anytime search-based planners [12]), a poor initialization scheme in optimization, a poor search heuristic, or because a certain set of constraints could not be met. Answering such questions could potentially reveal issues with the planning method itself, clarify the part of the problem that is difficult to achieve, or inform future mechanical robot re-designs. On the other hand, motion planners could also fail when a problem has no solution. In such situations, explaining failure might require proving that there is no solution [13] and why.

2) Trajectory-contrastive questions: A trajectory obtained by a planner might be unexpected to a user: it might look unfeasible or sub-optimal, pass closer to an obstacle than expected, etc. This question contrasts obtained trajectory A with a trajectory B that was expected by the user. The question may concern: *i)* the full trajectory ξ_A; *ii)* the trajectory of a subset of the variables $\mathcal{Q}' \subset \mathcal{Q}$; *iii)* the trajectory of a link's pose; or *iv)* the trajectory of an arbitrary function $f[\xi]$ (e.g. power consumption). It may also concern the above on a portion of the trajectory $\mathcal{T}' \subset \mathcal{T}$ (e.g. a waypoint).

Such questions may arise when users have the wrong mental model of a problem, which leads them to believe that the optimal plan should be different. In this case, answering such questions could reveal gaps in knowledge (e.g. the robot cannot fit through this passage) and trigger an update towards a correct mental model. This is called model reconciliation [14] in the literature of "Explainable AI Planning" (XAIP).

In addition to knowledge gaps, trajectory-contrastive questions may arise when motion planners are sub-optimal or incomplete—and the user's expected trajectory is better than the planner's trajectory. Due to the non-convexity and high-dimensionality of many motion planning problems, there is often a need to use heuristic-driven, local, or anytime planners, and these could perform worse than humans in some situations. Answers to trajectory-contrastive questions could thus reveal issues with the planning algorithm or its parameters (e.g. initialization scheme that leads to sub-optimal solutions), and help developers debug and improve the algorithm.

B. Kinds of explanations

1) Cost-based explanations: One of the potential explanations for a question "Why A and not B?" is that the cost of A is lower than that of B. To be useful, such explanation might need to be accompanied by a description or visualization of the contributors to this cost difference. This kind of explanation serves the main purpose of updating the user's mental model of the system, as well as raising trust in the plan. In the case of multi-objective planning, it may be important to explain why the trajectory provides an optimal trade-off of the objectives [15].

Given two trajectories ξ_A and ξ_B, this explanation therefore compares $C[\xi_A]$ to $C[\xi_B]$. It only makes sense to provide these explanations when all constraints $\mathcal{H}$ are satisfied on both trajectories.

2) Constraint-based explanations: Another type of explanation relates to identifying the set of constraints that makes a plan infeasible. For example, "the solution is plan A, not B because B would collide with the environment at time t".

More specifically, constraint-based explanations are sets of constraints that, if removed or relaxed, lead to the expected result. This involves searching over the power set of constraint functions $\mathcal{P}(\mathcal{H}) = \{\mathcal{H}_1, ..., \mathcal{H}_{|\mathcal{P}(\mathcal{H})|}\}$ where $\mathcal{P}$ represents the power set and each $\mathcal{H}_i$ is a subset of constraint functions. For example, a problem with three constraints $\mathcal{H} = \{H_{\text{target}}, H_{\text{collision}}, H_{\text{dynamics}}\}$ may become feasible both if the collision constraint is removed $\mathcal{H}_1 = \{H_{\text{target}}, H_{\text{dynamics}}\}$, or the "target" constraint is removed $\mathcal{H}_2 = \{H_{\text{collision}}, H_{\text{dynamics}}\}$.

3) Algorithm-parameter-based explanations: Due to the non-convexity of many motion planning problems and the use of anytime algorithms, the reason for a planner obtaining plan A instead of B can be that it "did not find B" even if B was lower-cost. Generally, this can happen if a certain parameter of the algorithm did not have the appropriate value—"appropriate" in the sense that the algorithm would find B if the parameter had a different value. Example explanations of this kind are: *i)* the algorithm was not run for long enough; *ii)* the algorithm was initialized from a solution that is on the basin of attraction of A not B; or *iii)* the algorithm uses a non-admissible heuristic. This kind of explanation can inform the development process since explanations can suggest alternative parameter values (e.g. "I would have found B if you had let me run for 5 more seconds"), and they may help drive algorithm improvements (e.g. better initialization schemes). Providing an actionable explanation of this type requires a search in the space of algorithm parameters, either until a plan is found (explanation of failure) or a new solution $\xi_{A'} \approx \xi_B$ is found (contrastive explanation).

4) Design-based explanations: Another kind of explanation is related to the mechanical design of a robot. Plan B which was expected by the user could be infeasible because "the robot's arm is not long enough", "the robot's body is too heavy", etc. Such kind of explanation is useful in updating the user's mental model of the robot's capabilities and limitations. However, such explanations also provide useful routes for action at the level of design—they can inform subsequent mechanical design improvements that decrease failure, or better align with user's expectations and preferences.

Design-based explanations require finding a set of design parameter values p such that $C[\xi_B] < C[\xi_A]$ and constraints are still satisfied. Similarly, in the case of motion planner failure, it could be the case that for some p the problem becomes feasible. Computing (the existence of) these values requires searching over p, for example through gradient descent, random search, or evolutionary algorithms [16].

5) Environment-based explanations: Failure or unexpected trajectories could also result from characteristics of the environment. For example, the reason for a plan being unexpected could be "because area X is occupied". Such kind of explanation is tightly related to cost- and constraint-based explanations, but due to its focus on possible environment changes it can be useful in informing structural changes to make in the environment itself. It can suggest design actions to apply to the environment (e.g. a new door, moving furniture).

Similarly to design-based explanations, these explanations rely on a parameterization of the environment (e.g. heightfield [17], navigation mesh [18]) and searching over its parameters.

6) Clarifying explanations: Posing the question "why plan A instead of B?" involves proposing an alternative plan, but manually "drawing" feasible trajectories may be hard for certain problems. To answer users' questions, then, it might be necessary to compute alternative trajectories $\xi_C \approx \xi_B$ that are feasible. This would provide explanations such as "A was obtained rather than B because B is not feasible. Did you mean C? C is close to B and is lower cost than A. The reason for this is..." To obtain such clarifications, we can search for a trajectory around ξ_B that is feasible. In an optimization-based motion planning method, this could involve solving the original problem from multiple initializations around $\mathcal{B}$, and/or a cost to favor solutions close to ξ_B. In sampling-based methods, a bias towards ξ_B could be used.

C. Purposes of explanations

The previous taxonomy of explanations implicitly suggests multiple purposes for explanations in motion planning:

1) Developer-centered debugging and algorithm improvement: One of the purposes of explanations is to help developers find issues with the algorithms, such as why they fail or behave sub-optimally in certain situations, or suggest ways in which they could be improved. The "algorithm-parameter-based explanation" is motivated by such a purpose.

2) Informing robot design and environment changes: Explanations can inform the mechanical design of a system, or drive structural changes to the environment itself. Such explanations can thus serve as means to promote interdisciplinary work between algorithm developers, mechanical engineers, environment designers, and operators (Fig. 1).

3) User-centered understanding and responsibility: Explanations can also be used to increase the knowledge a lay user has over a system. This allows users to correctly decide whether to use and trust a system or not, as well as to take responsibility for the system's actions [19].

4) Cooperative motion planning: Explanations can be used as a tool to improve the performance of a collaborative planner. For example, through explanations, a user may learn in which contexts a motion planner usually fails, and can thus intervene to propose plan initializations in those situations.

IV. Explainable motion planning methods

In this section we present two proof-of-concept *explainable motion planners*. Our main purpose here is to show what such methods could look like, and to obtain experimental results that motivate our research agenda.

A. Explainable optimization-based planner

We designed a prototype explainable optimization-based motion planner based on Trajopt [11], focusing on constraint and algorithm-parameter-based (in particular initialization-based) explanations. The problems are solved using Sequential Quadratic Programming as implemented in Trajopt [11], and trajectories parameterized by waypoints, i.e. $\mathcal{T} = \{1, ..., T\}$.

In case of failure, the user specifies the kind of explanation they are interested in: constraint, or algorithm-initialization. The constraint-based method is as follows:

```
Algorithm 1 (constraint-based failure explanation):
1: for each $\mathcal{H}_i$ in $\mathcal{P}(\mathcal{H})$:
2:     $\xi_i = \text{argmin}_\xi \; C[\xi] + \mathbf{1}_{H_{\text{target}} \in \mathcal{H}_i} \alpha C_{\text{target}}[\xi]$ s.t. $\mathcal{H}_i$
3: $k = \text{argmin}_i \; C[\xi_i] - \beta|\mathcal{H}_i| + \gamma C_{\text{target}}[\xi_i]$ s.t. $H_{\text{collision}}$
4: return Message("Not all constraints could be satisfied.
   The problem would be feasible if $\mathcal{H} \backslash \mathcal{H}_k$ were dropped and
   the target was $C_{\text{target}}[\xi_k]$ meters away from the original.")
```

The method first obtains the power set $\mathcal{P}(\mathcal{H})$ of the constraints (i.e. all combinations of active constraints) and solves all combinations of problems, where each problem considers only a subset $\mathcal{H}_i$ of constraints (lines 1-2). When removing constraints over link-poses, represented by H_{target}, we add them as costs $C_{\text{target}} = |H_{\text{target}}|$ that promote smallest possible distances to satisfaction (line 2). α is a constant parameter. We assume the removal of collision constraints to be an environment- and not constraint-based explanation. Therefore, as an explanation we use the solution that respects all collision constraints ($H_{\text{collision}}$) and a good balance between low cost, high-number of constraints $\mathcal{H}_i$, and low distance to targets, the latter weighted by positive constants β and γ (line 3).

Our method for algorithm-initialization-explanation is:

```
Algorithm 2 (initialization-based failure explanation):
1: for $i = 1, ..., N_{\text{max}}$:
2:     Pick random initialization and use it below
3:     $\xi_i = \text{argmin}_\xi \; C[\xi]$ s.t. $\mathcal{H}$
4:     if $\xi_i$:
5:         return Message("The initialization was in the basin
           of attraction of an infeasible local minimum. The planner
           would succeed with initialization $\xi_i$.")
6:     else:
7:         return Message("Unfeasible or hard problem.")
```

The method re-solves the problem from multiple uniformly-sampled initializations until it finds a feasible solution or a maximum amount of attempts (N_{max}) is reached.

In case of trajectory-contrastive explanations, the user provides a trajectory ξ_B that they expected, manually in a user-interface by specifying waypoints in configuration space. Waypoints are interpolated to obtain a trajectory of the same size as the original trajectory, and then an explanation is computed based on ξ_A and ξ_B. The explanation-computation method is as in Algorithm 3. The method starts by checking whether any of the constraints is not satisfied (line 2), in which case we compute a closeby feasible trajectory (lines 3-4). Then, the method uses the cost of the alternative plan ξ_C to return an appropriate explanation.

```
Algorithm 3 (trajectory-contrastive explanation):
1: $\xi_C = \xi_B$
2: if $\exists_i \; H_i(\xi_B) > 0$:
3:     Use $\xi_B$ as an initialization below
4:     $\xi_C = \text{argmin}_{\xi} \; C[\xi] + \alpha \sum_{t=1}^{T} ||\xi(t) - \xi_B(t)||^2$ s.t. $\mathcal{H}$
5:     msg = "$\xi_B$ is not feasible, do you mean $\xi_C$?"
6: if $C[\xi_A] < C[\xi_C]$:
7:     return Message(msg+"$\xi_A$ has lower cost than $\xi_C$.")
8: else:
9:     return Message(msg+"To obtain $\xi_C$, the planner would require an initialization closer to $\xi_C$.")
```

B. Explainable sampling-based planner

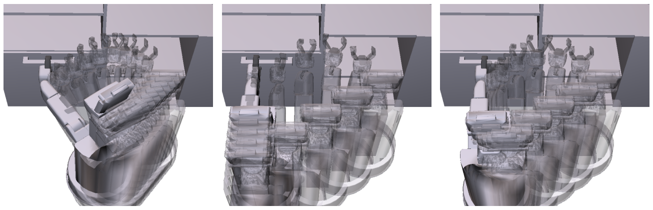
We designed a prototype explainable sampling-based motion planner, focusing on algorithm-parameter-based (in particular time-based) explanations. Users provide alternative (expected) trajectories in the same way as for the optimization-based planner. We use an off-the-shelf anytime motion planner: in particular, our experiments use "Batch Informed Trees" (BIT*) [20]. To obtain explanations for why the result was not the expected (was not ξ_B), we run the same planner with a large time limit and a new stopping criterion $||\xi - \xi_B|| \leq d_{\text{min}}$ which halts path refinement when the distance between ξ_B and the current solution ξ is small enough (d_{min} is a constant parameter). If such a solution is found, then the planner provides the explanation "The planner obtained A because of a low computation time budget, it would find B if the time budget was 30s".

V. Experiments

In this section we illustrate and evaluate the explanation methods just described. The experiments consist of four different types of explanation computed with the algorithms described in Section IV. We use the Toyota HSR robot [21] simulated in OpenRAVE [22] with the Open Dynamics Engine (ODE) [23] to compute collision and kinematics, the Trajopt library [11] to solve the trajectory optimization problems of Algorithm 1-3, and OMPL [24] for the BIT* algorithm.

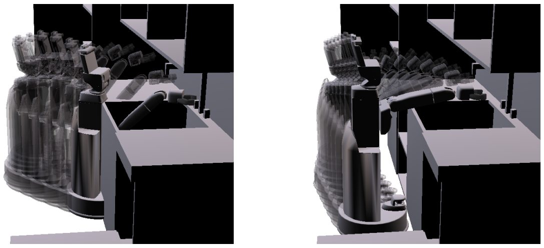
A. Experiment 1: clarification and cost explanations

We solved a motion planning task which involved reaching and grasping a handle (i.e. a hand-pose constraint at $t = T$), using Trajopt. The cost was squared velocities $C[\xi] = \sum_{t=1}^{T-1} ||\xi(t+1) - \xi(t)||^2$. The planner found the trajectory shown in Fig. 2a. We then assumed a hypothetical user asks why the trajectory does not approach the target frontally. To do this the user provides trajectory ξ_B by manually setting two waypoints in configuration space, one in front of the target, and one approximately grasping the target (Fig. 2b). We selected the waypoints in order to lead to a small collision with an object on the robot's back. We then used Algorithm 3 to compute a trajectory-contrastive explanation. Since the provided trajectory was infeasible, the algorithm used clarification (i.e. found a closeby feasible trajectory, shown in Fig. 2c), and provided Explanation 1.



(a) ξ_A (b) ξ_B (c) ξ_C

Fig. 2: Question clarification. The planner provides a solution ξ_A to grasp a handle (a). The user asks why the plan does not use a trajectory with a frontal approach ξ_B (b). Since the user-provided trajectory ξ_B is infeasible, the planner obtains the closest feasible solution ξ_C and returns Explanation 1.



(a) Original problem (b) Relaxed problem

Fig. 3: Constraint-based explanation. Planner fails in (a) because the target is unreachable. It finds a feasible solution by relaxing the constraints (b), and provides Explanation 2.

Explanation 1: The user-provided trajectory ξ_B (Fig. 2b) is infeasible, do you mean "why ξ_A (Fig. 2a) rather than ξ_C (Fig. 2c)"? The reason is the cost of ξ_A is lower.

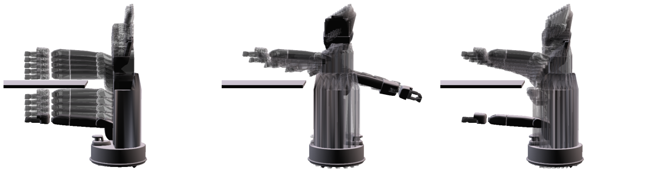
B. Experiment 2: constraint explanations

We solved a planning task to reach and grasp a water tap using Trajopt (i.e. hand-pose constraint at $t = T$). Compared to the previous problem, we added a new constraint H_{vel} which forces consecutive waypoint distances to be under a threshold. The planner failed (Fig. 3a) and we ran Algorithm 1 to obtain a constraint-based explanation. The algorithm found a solution that satisfies collision, ignores H_{vel}, and relaxes the target constraint—and provided Explanation 2.

Explanation 2: The planner failed because constraints H_{vel}, H_{target}, $H_{\text{collision}}$ were not satisfied. The problem would be feasible (Fig. 3b) if constraint H_{vel} was removed, and the target was 0.15 meters away from the original.

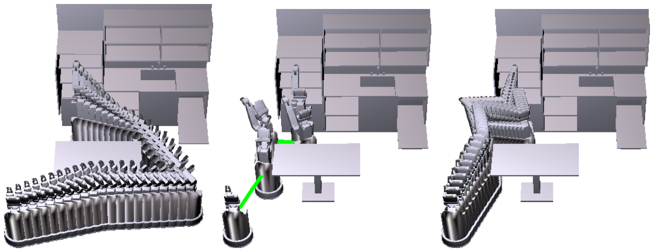
C. Experiment 3: initialization explanations

We solved a motion planning task which involved moving an arm in configuration-space from above to below a table (i.e. configuration-space constraint at $t = T$), using Trajopt. Since Trajopt uses a linearly interpolated initialization, it failed to find a feasible solution, as shown in Fig. 4a. We then used Algorithm 2 to compute an initialization-based



(a) Failure (b) New initialization (c) Success

Fig. 4: Initialization-based explanation. An optimization-based planner uses straight-line initialization and fails to find a trajectory (a). It then finds initialization (b) which leads to a feasible plan (c). It provides Explanation 3.



(a) Original plan ξ_A (10sec budget) (b) User's expected trajectory ξ_B (c) New plan, using 30sec budget

Fig. 5: Time-budget-based explanation. A sampling-based planner finds solution ξ_A (a). A user asks why the plan does not go through the left of the table, by providing trajectory ξ_B (b). The planner re-runs the search until a solution close to ξ_B is found. It provides Explanation 4.

explanation. The algorithm found an alternative initialization (Fig. 4b), which lead to a feasible plan (Fig. 4c).

Explanation 3: The planner failed because the initialization was in the basin of attraction of an infeasible local minimum. It would succeed using another initialization such as (Fig. 4b), obtaining solution (Fig. 4c).

D. Experiment 4: time-budget explanations

We solved a motion planning task which involved reaching a shelf (configuration-space constraint at $t = T$), using the sampling-based planner BIT*. We set the time budget to 10s. We show the solution found in Fig. 5a. We then simulated a user asking why the trajectory does not go around the table through the left. To do this we manually provided trajectory ξ_B using waypoints (Fig. 5b). We then used the algorithm described in Section IV-B to obtain a solution that is close to the manually-provided trajectory. The algorithm found such solution at around $t = 30$ seconds of refinement (Fig. 5c).

Explanation 4: The planner obtained ξ_A (Fig. 5a) because of low computation time budget, it would find $\xi_C \approx \xi_B$ (Fig. 5c) if the time budget was 30s.

E. User study

For each of the above explanations, we evaluated user satisfaction and criticism in order to estimate the usefulness and limitations of each explanation method. To do this we conducted a user study through an online questionnaire. The users were 29 experienced roboticists with more than 1 year of experience in motion planning (median 4 years). We first asked the users about whether the planners they currently use provide satisfactory reasons for failure when they fail. Then, each user was presented with the problem descriptions and explanations shown in Sections V-A-V-D above. For each explanation, users evaluated their satisfaction with the explanation (1-7 Likert scale from "very dissatisfied" to "very satisfied") and provided criticism in free text.

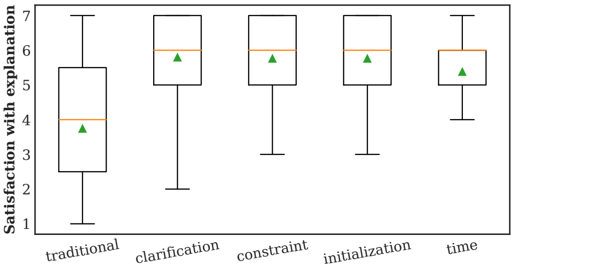


Fig. 6: Users' satisfaction with explanations provided by our and traditional methods. 1 is very dissatisfied, 7 very satisfied.

We show users' satisfaction with the explanations in Fig. 6. The median satisfaction was 6 for all our methods. On the other hand, the median user was neutral (score 4) about traditional planners' ability to explain failure. The *clarification experiment* had lower worst-case satisfaction due to a user who found the language used would be hard to understand for a lay user. Some users noted visualization of collisions would help the explanation, while another commented that this kind of explanation would be more suited to lay users, as it could be trivial to experts. For the *constraint-based explanation*, multiple users commented that the explanation could include a visualization of the feasibility region. One user questioned whether users should be shown a single explanation or all possible explanations (i.e. all possible constraint removals that lead to feasibility). Other users noted this was the most useful explanation to them. In the *initialization-based explanation*, three users commented that the method should have used multi-starts by default. One user noted that this explanation is more suitable to expert than lay users. For the *time-based explanation* three users commented that time-based explanations should also say why it takes longer to find path B (e.g. narrow passage).

VI. Discussion and Research agenda

The purpose of this paper has been to explore the concept of explanations for motion planning and outline a research agenda in that direction. So far we have introduced a taxonomy of explanations, the motivations for their use, and new proof-of-concept explainable planners that provide both *failure* and *contrastive* explanations. Expert motion planning users were consistently satisfied with the explanations computed by our methods, while they found traditional methods to be relatively unable to provide satisfactory reasons for failure. Based on the issues raised so far, we conclude with a research agenda.

A. Metrics of explanation quality

In Experiment 2 we assumed a specific metric of explanation quality (line 3 of Algorithm 1). Explanations that optimize this metric might not be those that satisfy human users the most, or those that are most effective at updating humans' mental model of the problem. Our study also highlighted that different users might want to look at all possible explanations, or provide their explanation preferences. One important research direction is, therefore, identifying good metrics of explanation quality. In the literature of explainable machine learning these are usually modelled as explanation length [25], or effectiveness at updating a user's mental model of the system [26], [8]. In motion planning, we will need to conduct user studies to understand what factors of explanations are most important to users.

B. Explanation methods

There is a need to identify and develop efficient ways to compute explanations. For example:

Cost and constraint-based explanations: Tools for explaining the influence of motion in the value of costs and in constraint satisfaction. Methods of sensitivity analysis and visualization of feasibility regions could be useful here (see user study), as well as methods for obtaining motion abstractions [27] and homotopy classes [6]. An interaction with the following literature is likely to be useful: culprit detection for task and motion planning [28], minimal constraint removal for motion planning [1], Irreducible Infeasible Set (IIS) methods for optimization [7], among others. Additionally, motion-planning-specialized methods to prove the inexistence of a solution to a problem are important [13], potentially building on analysis of the space of plans [29], [30], [6].

Heuristic-based explanations: Methods to automatically construct admissible heuristics are necessary here. If a method uses an inadmissible heuristic, then identifying whether that is the reason for the undesired result requires solving the problem with an admissible heuristic. Examples of methods that could be used or extended include those for admissible collision geometry based on sampling [27] or nesting [31].

Initialization-based explanations: In Experiment 3, we used random multi-starts to find an alternative initialization for the planner. Even though this is a common strategy in optimization-based planning [32], it may take large amounts of time to find a solution—and therefore an explanation. This asks for research on methods for efficient global optimization in motion planning, methods for visualization of local minima [33], and for computation of homotopy classes [6].

Design-based explanations: The challenge with design-based explanations is the high dimensionality of the space and the difficulty in finding design parameterizations. We need differentiable robot models [34], [35] so we can compute the impact of a change in link length (or mass, or joint positioning) on the cost and constraint functions. Additionally, methods for exploring the high dimensional search space are required, such as those used for design optimization [16].

Environment-based explanations: Differentiable environment models [36] are needed to provide environment-based explanations for optimization-based planners. Even though real-world robot motion planning typically relies on high-resolution, noisy maps, recent work on compressing large maps for fast planning [18] could be leveraged for efficiency.

C. Interfaces and communication

Asking "why plan A instead of B?" involves proposing an alternative plan, which is difficult to do precisely due to the difficulty in manually "drawing" trajectories. In our experiments, the user asked questions by manually defining waypoints in configuration space. This process would be unsuitable for fast-pace robot missions, or systems targeted at non-experts. Therefore, effective user interfaces will be an essential part of explainable planners. The interface design might also depend on the level of users' expertise and modality preferences (e.g. voice commands). In terms of natural language explanations, future research should try to identify what makes the communication of motion planning explanations effective, and use such insights to design interpretable [37], [2] communication methods.

D. Learning from the XAIP and XAI literature

Recent work on task planning explanations [8], [38] can inform research on motion planning. For example, [39] searches over different abstractions of a planning problem to provide explanations according to a user's expertise. Methods for providing environment-based explanations of planner failure have also been proposed in task planning [40]. On the other hand, the "model reconciliation" paradigm used by these methods may be hard to implement for motion planning in practice, since it assumes knowledge of the user's mental model of a planning problem. In motion planning, it may be difficult to know which robot collision-geometries/constraints the user is not aware of, and thus abstractions or probabilistic methods may have to be employed. Recent work on computing state-space abstractions for motion planning [27] could be useful here, as well as work on the human psychology of planning [41].

One kind of explanation we have omitted so far is "global" explanation, which is popular in machine learning [25]. This involves computing a simplified algorithm which is easier to interpret than the original. Inspiration could be drawn from these methods to identify which variables and problem abstractions account for most of the planned behavior. Another relevant research problem is that of mapping out a diverse set of situations (e.g. spatial configurations) that lead to failure—thus providing "summaries" [42] of planner failure types.

E. Learning from social and cognitive science

Research on the social and cognitive science of explanations will have to inform the design of motion planning explanations. For example, studies have shown that complete explanations are overwhelming to people's capabilities (e.g. providing all design-, environment-, constraint-, and initialization-explanations simultaneously) [3]. Finally, insights regarding user trust and reliance on algorithms [43] should be taken into account when designing explainable motion planners.

References

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

[2] M. Kwon, S. H. Huang, and A. D. Dragan, "Expressing robot incapability," in *Proceedings of the 2018 ACM/IEEE International Conference on Human-Robot Interaction*, 2018, pp. 87–95.

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

[4] S. Wachter, B. Mittelstadt, and C. Russell, "Counterfactual explanations without opening the black box: Automated decisions and the gdpr," *Harv. JL & Tech.*, vol. 31, p. 841, 2017.

[5] K. Molloy, L. Denarie, M. Vaisset, T. Simon, and J. Corts, "Simultaneous system design and path planning: A sampling-based algorithm," *IJRR*, vol. 38, no. 2-3, pp. 375–387, 2019.

[6] S. Bhattacharya, M. Likhachev, and V. Kumar, "Topological constraints in search-based robot path planning," *Autonomous Robots*, vol. 33, no. 3, pp. 273–290, 2012.

[7] J. W. Chinneck, "An effective polynomial-time heuristic for the minimum-cardinality iis set-covering problem," *Annals of Mathematics and Artificial Intelligence*, vol. 17, no. 1, pp. 127–144, 1996.

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

[9] P. Hart, N. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," *IEEE Transactions on Systems Science and Cybernetics*, vol. 4, no. 2, pp. 100–107, 1968.

[10] S. Karaman and E. Frazzoli, "Incremental sampling-based algorithms for optimal motion planning," *RSS VI*, vol. 104, no. 2, 2010.

[11] 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," *The International Journal of Robotics Research*, vol. 33, no. 9, pp. 1251–1270, 2014.

[12] M. Likhachev, G. J. Gordon, and S. Thrun, "Ara*: Anytime a* with provable bounds on sub-optimality," in *Advances in Neural Information Processing Systems*, 2003, pp. 767–774.

[13] J. Basch, L. J. Guibas, D. Hsu, and A. T. Nguyen, "Disconnection proofs for motion planning," in *ICRA*. IEEE, 2001, pp. 1765–1772.

[14] S. Sreedharan, T. Chakraborti, and S. Kambhampati, "Handling model uncertainty and multiplicity in explanations via model reconciliation," in *ICAPS*, 2018.

[15] R. Sukkerd, R. Simmons, and D. Garlan, "Toward explainable multi-objective probabilistic planning," in *2018 IEEE/ACM 4th International Workshop on SEsCPS*. IEEE, 2018, pp. 19–25.

[16] M. Brandao, R. Figueiredo, K. Takagi, A. Bernardino, K. Hashimoto, and A. Takanishi, "Placing and scheduling many depth sensors for wide coverage and efficient mapping in versatile legged robots," *IJRR*, Dec 2019.

[17] P. Fankhauser and M. Hutter, "A Universal Grid Map Library: Implementation and Use Case for Rough Terrain Navigation," in *Robot Operating System (ROS) The Complete Reference (Volume 1)*, A. Koubaa, Ed. Springer, 2016, ch. 5.

[18] M. Brandao, O. B. Aladag, and I. Havoutis, "Gaitmesh: controller-aware navigation meshes for long-range legged locomotion planning in multi-layered environments," *IEEE RAL*, 2020.

[19] M. Coeckelbergh, "Artificial intelligence, responsibility attribution, and a relational justification of explainability," *Science and engineering ethics*, pp. 1–18, 2019.

[20] J. D. Gammell, S. S. Srinivasa, and T. D. Barfoot, "Batch informed trees (bit*): Sampling-based optimal planning via the heuristically guided search of implicit random geometric graphs," in *IEEE ICRA*, May 2015, pp. 3067–3074.

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

[22] R. Diankov, "Automated construction of robotic manipulation programs," Ph.D. dissertation, Carnegie Mellon University, Robotics Institute, August 2010.

[23] Open dynamics engine. [Online]. Available: http://www.ode.org/

[24] I. A. Şucan, M. Moll, and L. E. Kavraki, "The Open Motion Planning Library," *IEEE Robotics & Automation Magazine*, vol. 19, no. 4, pp. 72–82, December 2012, http://ompl.kavrakilab.org.

[25] M. T. Ribeiro, S. Singh, and C. Guestrin, ""why should i trust you?" explaining the predictions of any classifier," in *Proceedings of the 22nd ACM SIGKDD*, 2016, pp. 1135–1144.

[26] I. Lage, A. Ross, S. J. Gershman, B. Kim, and F. Doshi-Velez, "Human-in-the-loop interpretability prior," in *NIPS*, 2018, pp. 10 159–10 168.

[27] M. Brandao and I. Havoutis, "Learning sequences of approximations for hierarchical motion planning," in *ICAPS*, Jun 2020.

[28] F. Lagriffoul and B. Andres, "Combining task and motion planning: A culprit detection problem," *The International Journal of Robotics Research*, vol. 35, no. 8, pp. 890–927, 2016.

[29] J. Hoffmann, J. Porteous, and L. Sebastia, "Ordered landmarks in planning," *JAIR*, vol. 22, pp. 215–278, 2004.

[30] O. Brock and Y. Yang, "Efficient motion planning based on disassembly," in *Robotics: Science and Systems*, 2005.

[31] A. Orthey, A. Escande, and E. Yoshida, "Quotient-space motion planning," in *IEEE/RSJ IROS*. IEEE, 2018, pp. 8089–8096.

[32] 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 *17th IEEE-RAS Humanoids*, Nov 2017.

[33] A. Orthey, B. Frész, and M. Toussaint, "Motion planning explorer: Visualizing local minima using a local-minima tree," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 346–353, 2019.

[34] F. de Avila Belbute-Peres, K. Smith, K. Allen, J. Tenenbaum, and J. Z. Kolter, "End-to-end differentiable physics for learning and control," in *NIPS*, 2018, pp. 7178–7189.

[35] J. Carpentier and N. Mansard, "Analytical derivatives of rigid body dynamics algorithms," in *RSS 2018*, 2018.

[36] S. T. O'Callaghan and F. T. Ramos, "Gaussian process occupancy maps," *IJRR*, vol. 31, no. 1, pp. 42–62, 2012.

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

[38] M. Cashmore, A. Collins, B. Krarup, S. Krivic, D. Magazzeni, and D. Smith, "Towards explainable ai planning as a service," *arXiv preprint arXiv:1908.05059*, 2019.

[39] S. Sreedharan, S. Srivastava, and S. Kambhampati, "Hierarchical expertise level modeling for user specific contrastive explanations." in *IJCAI*, 2018, pp. 4829–4836.

[40] S. Sreedharan, S. Srivastava, D. Smith, and S. Kambhampati, "Why cant you do that hal? explaining unsolvability of planning tasks," in *IJCAI*, 2019.

[41] R. Morris and G. Ward, *The cognitive psychology of planning*. Psychology Press, 2004.

[42] 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.

[43] B. Green and Y. Chen, "The principles and limits of algorithm-in-the-loop decision making," *Proceedings of the ACM on Human-Computer Interaction*, vol. 3, no. CSCW, pp. 1–24, 2019.