Constraint-driven abductive explanations as a mechanism for reflective normative reasoning

Nathan Lloyd[1][0000−0002−7127−2500] and Peter R. Lewis[1][0000−0003−4271−8611]

Ontario Tech University, Oshawa, Ontario, Canada
{fname.lname}@ontariotechu.ca

Abstract. Normative systems require reflective capacities if agents are to interpret social behavior in a structured manner. In social contexts, agents must not only register outcomes such as praise or punishment, but also understand their justification. When sanctioned, the central issue is not simply what happened, but why it was warranted: which expectations were in force, whether the conditions for a norm were present, and how a violation is recognised. In this work, we extend the Abductive Event Calculus to support justification-driven explanation generation over both states and events. While the standard Abductive Event Calculus explains why fluents hold at particular times, our extension also explains the occurrence of events (i.e., sanctions). We first demonstrate these modifications using an adaptation of Shanahan's canonical chemical plant example, showing that standard explanatory behaviour is preserved. We then apply the extended framework to a simple normative scenario grounded in Bicchieri's formalisation of norms and the Expectation Event Calculus. Within this setting, empirical and normative expectations are operationalised to model conditional norm compliance, violation, and sanctioning. As such, this work contributes a method to generate causally grounded explanations about behavior, an essential capability for socially reflective reasoning. We conclude by outlining directions for future work, including multi-agent extensions and explanation selection.

Keywords: Abduction · Event Calculus · Normative Justification · Social Expectations · Reflection

1 Introduction

Norms are the informal rules that guide behavior, arising from shared beliefs and sustained through perceived adherence or enforcement. They rest on conditional, interdependent, and self-reinforcing expectations. Bicchieri's formalization of norms [6] can account for behaviors driven by one-way beliefs, such as descriptive norms like fashion trends or other imitative practices, that may eventually solidify into stable conventions within a group [22]. They also encompass more intricate behavioral patterns, including social norms, which additionally depend on expectations about what individuals believe other's believe they should

or should not do. The latter is sometimes accompanied by supporting behaviors that directly reinforce that norm; sanctions that may encourage or dissuade action. In prior work we address the issue that sanctions (rewards and punishments) are often assumed to clearly signal normative expectations [24], describing how delayed responses and blind revenge [31], misperception and accidental transgression [5], and strategic counter-punishment [29] may muddy normative interpretations.

Rather than reacting to such instances, we argue that causal explanations are required to attempt to make sense of enforcement. Actors confronted with a sanction naturally ask: "Why was I sanctioned?" "Was it because I violated a norm—and if so, was I unaware of it?" "Or was the response driven by something else, such as misinterpretation, bias, or strategic retaliation?". These questions reveal that sanctions do not automatically convey normative information. For a sanction to clarify a norm, the recipient and relevant observers must be able to attribute it to a specific violation within a shared rule framework. When that attribution is uncertain, the sanction's meaning becomes ambiguous. In prior work [23], we have explored contextualizing explanations in subjective context (self-other distinctions) using the Event Calculus (EC) [18] and Abductive Event Calculus (AEC) [47]. That work enabled us to generate perspective-relative explanations of fluent states, for example, accounts of the form "from their point of view X holds, therefore their plan will be Y". In that work, we were able to explore such extensions to contextualize explanations for some fluent state, however, explaining actions via abduction remained incomplete.

In this work, we extend the framework to generate explanations for actions, that is, the inclusion of actions as abductive goals whose realization is governed by explicitly represented preconditions encoding both causal and temporal structure. By evaluating the preconditions of these actions first, the resulting explanations are causally and temporally justified. We demonstrate our extensions to the Abductive Event Calculus through Shanahan's canonical chemical plant example [47], where we show how causal and temporal preconditions constrain abductive explanations of action. Then we apply this extension to a normative domain to address questions such as "Why was I sanctioned?", performing abduction over expectations (underpinning norms) to explain causally supported actions.

2 Background

2.1 Reflection through Abduction

Initially coined by Peirce [37], abduction was conceived as a third mode of inference, distinct from both deduction and induction, and central to explaining how original ideas arise [28]. Seeking to move beyond the traditional Aristotelian dichotomy of reasoning types [4], Peirce variously described: "hypothesis," "reasoning a posteriori," "abduction," "presumption," and "retroduction" [4]. Throughout his writings, the versatility of Peirce's language allowed for multiple interpretations of the term [32], indeed, Peirce himself revised and refined the concept

across decades. Peirce's "mature theory" positioned abduction as not merely another kind of reasoning, but as a distinct step in scientific inquiry [4][1]. The latter conceptualization of abduction sees it as a distinct stage of the scientific method, when faced with surprising facts, an explanation is required such that a proposition would lead (certainly or probably) to the observed facts [39, pp. 94-95]. A hypothesis counts as explanatory insofar as it provides a causal account of the phenomenon at issue [50]. As Bochman observes [7], the close link between abduction and causation is especially evident in domains such as diagnosis. Abduction is commonly expressed in the following form [38, pp.231], whose structure resembles that of a syllogism with additions around "surprise" and "matter of course" [34].

> The surprising fact, C, is observed;
> But if A were true, C would be a matter of course.
> Hence, there is reason to suspect that A is true.

For this, abduction is core to the 'logic of discovery' [34], a philosophical concept concerned with how new ideas, hypotheses, or theories are generated—that is, how discovery happens, and hypothetical thinking, a methodological process of inquiry that seeks to resolve surprise, anomaly, or ambiguity through explanations that are both plausible and empirically verifiable [12]. The emphasis on plausibility and subsequent verification through deductive and inductive methods suggests that abduction constitutes a weaker form of inference, insofar as it yields merely candidate hypotheses rather than necessarily true explanations [33].

In another line of inquiry, Harman introduced "Inference to the best explanation" [14] (IBE). IBE is frequently treated as the dominant contemporary interpretation of "abduction" [11]. This tendency is especially pronounced in artificial intelligence research, where abduction is commonly conceptualized in explicitly IBE-oriented terms [34]; some examples [35,9,49,17]. While some scholars have argued for a clear conceptual separation between the two [33], others maintain that such a distinction cannot be drawn with complete rigor [44]. Despite this, abduction is commonly distinguished by *creative* and *selective* abduction [25], or others, who avoid the term abduction, focus on the processes of *creative explanatory inference* and *evaluative explanatory inferences* [51], or *hypotheses generation* and methods for finding the best explanations [36], generating and justifying [11], respectively. Indeed, one may envision, like Thagard [50], abduction as a pipeline of processes where the type of evaluation seen in IBE follows the generation of a hypothesis.

We characterize abductive inference as a reflective mechanism because it enables agents to move from the recognition of surprising or anomalous observations to the construction of plausible explanatory hypotheses. Abductive reasoning, from both philosophical and psychological perspectives, is triggered by an agent's encounter with surprising data that cannot be assimilated within

[1] We condense the contributions and various perspectives around abduction, see [26] for a comprehensive treatment.

existing information, and it involves not only the logical structure of forming hypotheses but also the *normalization* of surprise and the adjustment of belief states in response to it [28]. In doing so, abduction equips agents not merely to register discrepancies in experience, but to reorganize their belief systems in light of them. It thereby enables the diagnosis of anomalies, the generation of candidate explanations, and the guidance of subsequent reasoning and action under conditions of uncertainty.

2.2 Event Calculus

The Event Calculus (EC) [18] is a narrative-based logical formalism that enables the specification of actions and their effects. It supports non-deterministic and concurrent actions, models indirect effects, and addresses the frame problem through the principle of inertia [46], whereby fluents persist unless affected by an event. The formalism is based on circumscription and is therefore non-monotonic, allowing newly acquired information to retract previously derived conclusions. The EC consists of three major components:

1. A set of general domain-independent axioms that provide general rules for temporal reasoning.
2. A set of specific domain-dependent axioms that provide contextual rules, defining how actions affect fluents.
3. A problem specification defining the world's initial state, a narrative, or goals.

As a formalism centered on actions and their effects, it may described as causally expressive, insofar one may specify an event initiating or terminating some property at a particular time. The Event Calculus is action-centered and explicitly temporal. Fundamentally, it is a temporal logic rather than a dedicated causal formalism: causality is not built in as a primitive, but instead encoded indirectly via *causal constraints*; domain-specific relationships specifying how events initiate or terminate fluents [48]. This 'causality' therefore differs from the full structural theories of causation common in modern causal modeling: there have, however, been extensions that would include such capacities, i.e., *what-if* reasoning [45].

2.3 Abductive Event Calculus

Shanahan demonstrates how Event Calculus axioms may be compiled into a meta-interpreter that closely mimics Prolog's object-level execution strategy to achieve temporal explanation, postdiction, diagnosis and planning (via abduction) [48]. Conceptually, this reverses the typical direction of inference with the Event Calculus which we may describe as forward, deductive, or as temporal projection. In the EC, temporal projection proceeds from (1) knowledge of action effects and (2) a history of events, to deduce (3) what holds at a given time. In contrast, abductive reasoning gives (1) knowledge of action effects, (3) an

outcome to satisfy, and aims to infer (2) the sequence of events that could have produced it.

In their work extending the abductive event calculus, Bracciali and Kakas [8] argue that the framework naturally supports rich causal explanations. Their work highlights how abductive explanations can be generated to recover from frame inconsistency. That is, when a fluent is observed to hold (or not) in a way that cannot be accounted for by the recorded sequence of events, the framework does not simply yield an inconsistency. Instead, it addresses the discrepancy by postulating additional, previously unobserved events, thereby constructing a plausible causal account in the presence of incomplete information. These explanations are not arbitrary: they must be temporally ordered, causally adequate (i.e., capable of initiating or terminating the relevant fluents), and minimally sufficient to restore consistency. Recent work by Lloyd and Lewis [23] resonates with this perspective. However, instead of introducing hypothesised events to resolve frame inconsistencies, they take a different approach: observed events are used to constrain, refine, and anchor the space of candidate explanations, effectively filtering and grounding the explanatory process in the available evidence.

This explanatory role of abduction and temporal reasoning is not confined to post hoc reconstruction of events, but also underpins models of deliberative agency. The **K**nowledge **G**oals **P**lan model (KGP) [15] provides a transition-based agent architecture that embeds abductive logic programming and the Event Calculus within an operational framework for planning, reactivity, sensing, and goal revision. In KGP, abduction is used not only to explain past observations but also to generate prospective courses of action,reflecting a common approach in the literature in which planning problems are addressed using abductive reasoning [13,47]. The KGP model was later augmented to include normative modalities [42,41], introducing deontic concepts such as obligations and prohibitions to further enrich the architecture. The KGP model served as a foundational framework for integrating abduction and temporal reasoning in agent systems, even though other agent programming paradigms ultimately achieved wider adoption. Subsequent developments, most notably the SCIFF framework [1], utilized abduction over expectations to specify what "admissible interactions" among agents are. This yields a semantics closely aligned with normative systems, where compliance and violation are evaluated relative to temporally structured obligations. In our work, we extend these ideas by integrating Bicchieri's formalization of norms [6] with Cranefield's Expectation Event Calculus [10], using abductive reasoning to ground explanations within a normative domain.

3 Embedding Causal and Temporal Constraints in the Abductive Event Calculus

Before turning to the normative domain, we first outline the modifications made to Shanahan's original abductive Event Calculus. Shanahan uses the chemical plant example to illustrate how partial-order plans may be constructed through

abductive inference in the Abductive Event Calculus (AEC) [47, p.13]. Following Shanahan's version 1.9 of the AEC [2], Shanahan describes how actions and their effects, now re-represented as metal-level predicates within the AEC, drive transitions between fluent states, such that a chemical plant's safety can be determined through a series of state constraints. The chemical plant is rendered safe by executing a sequence of actions to bring about those states, which in turn give way for subsequent actions. A corresponding flowchart is given in Figure 1.

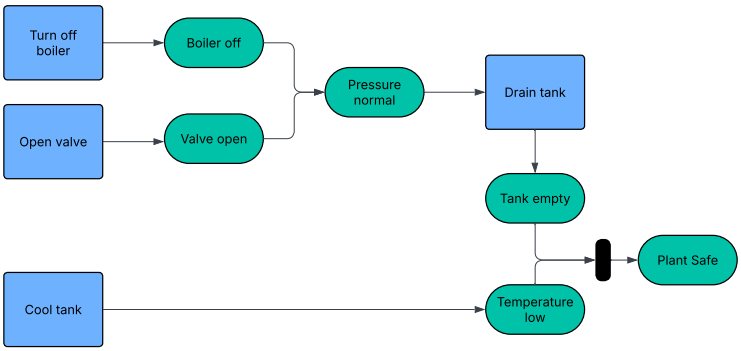

Fig. 1. Flow chart for the Chemical Plant Planning problem where *Plant safe* is given as a terminal state, requiring both the *tank empty* **AND** the *temperature low*. The fluent state *pressure normal* is dependent on either the *boiler off* **OR** the *valve open*. If the goal was to make the chemical plant safe, one may query $abdemo([holdsAt(plant\_safe, t)], Residue)$ where $Residue$ returns solutions for the satisfaction of the goal. A solution consists of a set of event occurrences together with a corresponding set of ordering constraints sufficient to achieve the fluent $Plant\_safe$. We may call each solution a plan or an explanation.

In Shanahan's Chemical Plant example [47], conditional dependencies are largely unidirectional: actions give rise to fluents through initiation and termination axioms, and fluents persist via inertia to support subsequent reasoning and actions. While actions may be enabled by the presence of certain fluents (implicitly via the requirements of the partial-order plan), the framework did not explicitly and directly encode action-to-action preconditions or enforce structured causal links between action occurrences, although this may be achieved through intermediary fluents. As a consequence, abductive explanations are constructed by hypothesizing event occurrences sufficient to establish desired fluent states, without requiring those events to be embedded within an explicitly ordered network of enabling actions. The emphasis lies on satisfying the declarative conditions of the Event Calculus theory (including the domain axioms) rather than enforcing a structured account in which each action must itself be

[2] Code Available here: https://www.doc.ic.ac.uk/~mpsha/planners.html

justified by prior enabling actions. Furthermore, this version of the AEC did not permit a goal to be specified as a narrative event. Consequently, a behavior could not itself be explained by prior behaviors or fluents as part of a larger narrative structure. For clarity, in Figure 1, one would not be able to explain why $abdemo([happens(drain\_tank, t)], R)$ but they could explain $abdemo([holdsAt(tank\_empty, t)], R)$. Shanahan acknowledged, however, that the AEC was extensible and could in principle support a wider range of goal representations.

3.1 Technical Contribution

This work extends the Abductive Event Calculus (AEC) [47] to support explanations of event occurrences in addition to fluent states. By incorporating abductive reasoning over the additional dependencies illustrated in Figure 2, the framework extends the conditions governing action occurrence. In particular, the calculus is enriched to represent not only fluents as preconditions for actions, but also actions as prerequisites for subsequent actions. This increases the representational capacity of the planner by enabling more expressive temporal and dependency structures within partial-order plans.

While these preconditions are similar in purpose to those in the KGP agent model [15], specifying requirements for actions that constrain plans (KGP's preconditions generally specify requirements for actions), they differ in that our preconditions are enforced as explicit abductive constraints during explanation/plan generation, directly shaping the abduction of event occurrences, rather than operationally checked during goal selection. This affords a tighter control over plan correctness (determined by preconditions) thereby pruning some sequences early.

Our approach to this extension is to include explicit action- and event-level preconditions, and to enforce their satisfaction prior to the abduction of event occurrences. Rather than defining *happens*/2 deductively:

```
axiom(happens(celebrate, T),
      [holdsAt(plant_safe, Tb), before(Tb, T)]).
```

preconditions are interpreted at the meta-level as constraints that must be satisfied prior to the hypothesis of an action instance. We do not introduce object-level clauses defining *happens*/2, which in prior work was a source of incompleteness [23, p.5]. This preserves the abductive status of event occurrence while ensuring that actions are introduced into a plan only when their supporting preconditions (fluents and events) are already established. This justification-first strategy propagates temporal ordering constraints through the partial-order plan. It ensures that each action occurrence is embedded within an appropriate supporting context. As a result, event introduction remains abductive, but is constrained by explicitly represented state and ordering dependencies. Unlike the KGP, our approach does not require a broader agent architecture, only a domain specification represented at the AEC meta-level is required, support by preconditions and temporal constraints that guide abductive explanation and plan generation.

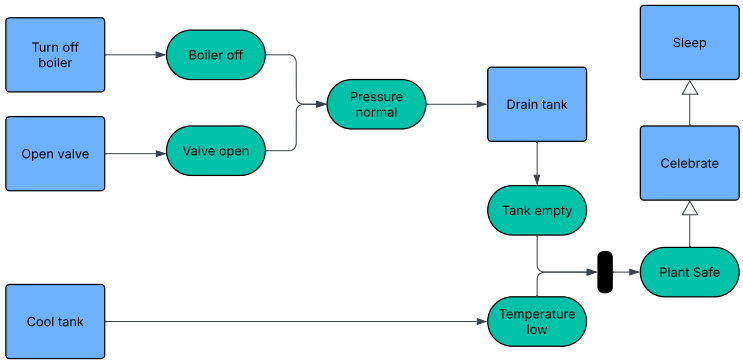

Fig. 2. A flowchart of the extended Chemical Plant Planning problem introduces two additional conditional dependencies: the action *celebrate* is contingent upon the fluent *plant safe* being true, and the action *sleep* is contingent upon *celebrate* having occurred in a preceding time step. This contingency is shown by the white-headed arrow which corresponds to a *precondition*/2 predicate

Operationally, if *happens*/2 is encountered as a goal, its associated preconditions are first retrieved and expanded into temporally constrained subgoals. For each fluent precondition F, the goals $holdsAt(F, T_p)$ and $before(T_p, T)$ are generated; for each action precondition A', the goals $happens(A', T_p)$ and $before(T_p, T)$ are introduced. These constraints are resolved prior to the abduction of the event instance itself. Only once the preconditions are satisfied is *happens*/2 hypothesized. We do this by introducing precondition axioms to the AEC:

```
axiom(precondition(A,F)).
axiom(action_precondition(A,A2)).
```

Operationally, preconditions are resolved before an event is abduced. In our implementation, *abdemo*/5 is modified to first expand preconditions of each event goal into temporally constrained subgoals. The relevant code snippet illustrates this:

```
abdemo([happens(A,T)|Gs],R1,R3,N1,N4) :-
  executable(A),
  % Preconditions are resolved first
  (event_preconditions(A,T,Ps) -> true ; Ps = [] ),
  abdemo(Ps, R1, R2, N1, N2),
  enforce_diff_times(A,T,R2),
  abresolve(happens(A,T), R2, [], R3, _),
  abdemo(Gs, R3, R4, N2, N4).
```

Importantly, causal and temporal preconditions are established before abducing an event occurrence, and temporal ordering is managed as a constraint

system. This ensures temporal consistency throughout the evolving partial-order plan. Temporal ordering constraints are treated as first-class abductive constraints rather than procedural predicates: goals of the form $before(T_p, T)$ introduce ordering requirements instead of merely querying existing temporal relations, allowing the plan to maintain declarative temporal structure. Furthermore, we utilize $diff\_time/2$ in the abductive reasoning process to enforce separation between action occurrences. After resolving an event's preconditions, $diff\_time(A, T)$ constrains the time point T of action A so that it does not coincide with already- abduced events. This prevents events that are declared to occur at different times from collapsing into a single time instant, ensuring concurrency arises only when justified in the domain.

In the revised formulation, existentially quantified time points are preserved as shared logical variables throughout abduction, avoiding the introduction of Skolem constants and thereby maintaining relational and temporal dependencies and partial-order integrity. When a time variable T is unbound at abduction, a fresh symbolic constant is introduced via $fresh\_time/1$ solely to instantiate the event occurrence; unlike Skolem functions these constants do not depend on other variables, so relational constraints remain intact. Existing events are reused when present in the residue, and events with ground time points are directly abduced at the specified times. This approach ensures that temporal variables remain shared, can participate in constraint propagation, and preserves the relational temporal structure, enabling declarative expression of orderings, separations, and causal dependencies while respecting abductive semantics. The $abresolve/5$ predicate below illustrates how our system handles event abduction while preserving the relational structure of time.

```
% 1) If event already exists in residue, reuse it
abresolve(happens(A,T),[RH,RB],[],[RH,RB],false) :-
  member(happens(A,T),RH), !.
% 2) If time is ground, abduce the event at that exact time
abresolve(happens(A,T),[RH,RB],[],[[happens(A,T)|RH],RB],true) :-
  executable(A),
  ground(T),
  \+ member(happens(A,T),RH).
% 3) If time is variable, create a new time variable
abresolve(happens(A,T),[RH,RB],[],[[happens(A,Tnew)|RH],RB],true) :-
  executable(A),
  var(T),
  fresh_time(Tnew),
  T = Tnew.
```

Given these extensions, and per the additions to the problem shown in Figure 2, querying $abdemo([holdsAt(plant\_safe, t)], R)$ continues to produce the expected two plans, each corresponding with the two possible branches of achieving the fluent $pressure\_normal$; turning off the boiler or opening the valve. If the axiom $axiom(precondition(drain\_tank, pressure\_normal))$. is added to the domain, the model would now also be able to explain that event. Querying

$abdemo([happens(celebrate, t)], R)$ returns three complete plans, two correspond to the $plant\_safe$ branches with their respective $before/2$ ordering constraints, and a third explanation being the standalone behavior$happens(celebrate, t)$. Querying $abdemo([happens(sleep, t)], R)$ which sits at the end of our extended chemical plant scenario similarly has two explanation branches grounded in making the plant safe, another as a standalone behavior, but given an additional explanation where $happens(celebrate, T_p)$ is recognized as a precondition. Each example demonstrates the explanation of action goals with explicit preconditions, and that the extended framework correctly integrates action-level preconditions into abductive plan generation, preserving existing causal branches while introducing the necessary ordering constraints to justify action occurrences.

```
R = [[happens(sleep,t), happens(celebrate,c6), happens(cool_tank,c4),
happens(open_valve,c3), happens(drain_tank,c1)],
[before(c4,c5), before(c3,c5), before(c1,c5), before(c2,c5),
before(c4,c2), before(c3,c2), before(c3,c1), before(c1,c2)]]
R = [[happens(sleep,t), happens(celebrate,c10), happens(cool_tank,c8),
happens(turn_off_boiler,c7), happens(drain_tank,c1)],
[before(c8,c9), before(c7,c9), before(c1,c9), before(c2,c9),
before(c8,c2), before(c7,c2), before(c7,c1), before(c1,c2)]]
R = [[happens(sleep,t), happens(celebrate,c11)], [before(c11,t)]]
R = [[happens(sleep,t)], []]
```

The additional explanation(s), corresponding to the standalone behaviour(s), demonstrate that the framework continues to permit the abduction of event occurrences even in the absence of an explicitly established supporting causal structure. That is, an action may be hypothesised as occurring independently, without being triggered by prior events or justified by currently derived fluent conditions. This capacity to represent behaviour as potentially unprovoked, spontaneous, or not yet causally grounded is particularly desirable in normative contexts, where agents may act without an immediately observable reason, or where the justification for an action may emerge only retrospectively through further explanation. Multiple explanations, in this sense, provide an opportunity for evaluation and further inquiry.

The additional time-point ($c2$) arises because fluent preconditions are supported by introducing an existentially quantified time T_p such that $holdsAt(F, T_p)$ and $before(T_p, T)$. This time need not correspond to any actual event occurrence, rather, it serves as an intermediary time variable that appears only in ordering constraints and ensures that the required temporal support for the fluent is satisfied. These auxiliary time-points do not affect the actions in the plan or its causal structure. They can be filtered out after plan generation, or avoided by constraining precondition times to coincide with event times, but their presence is semantically harmless within the partial-order framework.

3.2 Conceptual Contribution

Conceptually, these extensions shift the framework from an implicitly ordered abductive planner toward a constraint-driven, justification-oriented, partially

ordered planning calculus. Temporal ordering is treated explicitly at the meta-level. As a result, concurrency emerges only when permitted by the domain, causal support is enforced prior to event introduction, and the framework can represent dependencies between actions and fluents in a way that respects the declarative semantics of the Event Calculus. This refinement incurs a controlled narrowing of the exploration space. By enforcing justification-first abduction and committing early to temporal constraints, the procedure reduces number of possible plans that need to be considered compared to exploring all ways of inserting and ordering events without guidance. Nevertheless, the procedure generates plans and explanations that are consistent with the domain axioms. The pruned branches correspond primarily to causally redundant or temporally degenerate reorderings of independent events. The framework therefore trades exhaustive plan enumeration for tractable generation of causally justified and temporally coherent plans.

4 A Running Normative Scenario with Conditional Norm Activation and Sanctioning

We apply these changes to a normative domain using Bicchieri's formalization of norms [6]. In this framework, we require representation for conditional social expectations. These are: empirical expectations, beliefs about common or prevalent behavior, and normative expectations, second-order beliefs about what others think ought to be done, or what is considered permitted, forbidden, or required. As an expectation-drive account of norms, we utilize a simple version of the Expectation Event Calculus [10], which is well suited for representing both types of expectations [21]. Bicchieri's definition of a social norms requires both an empirical expectation and a normative expectation that may be accompanied by a sanction to enforce the norm, though, in practice, not every normative expectation has an accompanying sanction.

We propose a straightforward normative scenario. A norm dictates that residents should retrieve their empty trash cans from the street after collection. Empirically, most residents follow this practice, and there is also a normative expectation that one ought to comply. For now, we omit deontic operators, representing normative expectations simply as what an agent should or should not do. This simplification allows us to sketch the basic structure of normative reasoning without committing to a more elaborate formal apparatus. However, richer forms of representation, like those described by Malle [27], quickly become necessary once we aim to capture the complexity of real-world normative systems. In this context, leaving a trash can out could obstruct neighbors or pedestrians, or simply be seen as unsightly. Residents who fail to collect their cans may face sanctions ranging from disapproving looks, a scoff, or to official complaints and fines... or worse (receiving a strongly worded letter from your neighbor). For now, we refer only to a general, unspecified and nondescript sanction. Like the primary norm, this sanction may expected, both empirically and normatively. Thus, we treat the sanction itself as a norm, but one that is conditional on the

primary social norm and the associated normative expectation. This is distinct from, but consistent with Bicchieri's formalisation [6, pp.11] allowing us to avoid introducing additional semantics or separate mechanisms for sanctions. It is ontologically first-order: it functions to maintain the integrity of the primary norm rather than to regulate enforcement behavior itself. As such, it does not constitute a metanorm [2], which has been modeled in the Expectation Event Calculus by Sengupta et al. [45]. We illustrate our example in Figure 3.

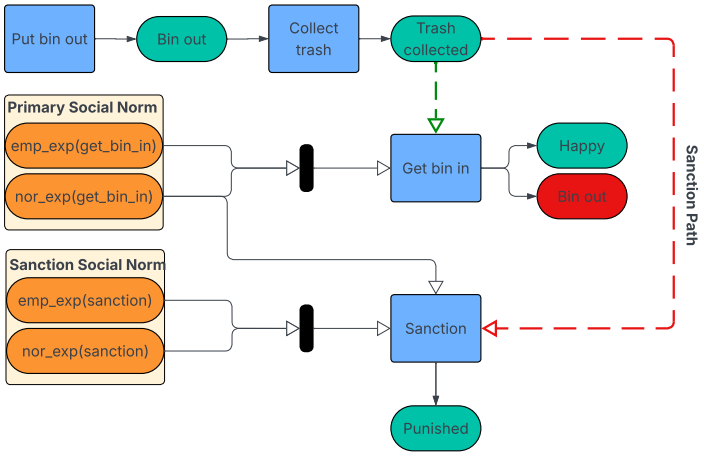

Fig. 3. A normative scenario illustrating how empirical and normative expectations over environmental states act as preconditions for norm compliant behavior.

The scenario begins with the environmental actions *put_bin_out* and *collect_trash*, which establish the state conditions *bin_out* and *trash_collected*. The latter is a precondition for the norm compliant action *get_bin_in*, which itself is dependent upon the presence of social expectations in the "primary norm". For clarity, note that norms are not modeled here as first-class objects; instead they are capture in terms of their constitutive expectations and the preconditions that support them. If the "primary norm" is fulfilled by completing the complementary behavior, the system progresses to the positive outcome state which we trivially refer to as the *happy* state. If the expectations or preconditions do not hold, the primary norm is not activated and therefore does not impose an obligation.

When the primary norm's normative expectation is active but the required action is not performed, a transgression can be identified that motivates the corresponding sanction. A sanctioning norm is thereby conditional both on on the normative expectation of the primary norm [6, pp.11], as well as conditional upon its own expectations; whether people actually carry out the sanction and

whether people believe it ought to be carried. The initial states and preconditions for this scenario are given below, the domain axioms follow Figure 3.

```
% Primary Social Norm
axiom(initially(emp_exp(get_bin_in)),[]).
axiom(initially(nor_exp(get_bin_in)),[]).
% Sanctioning Social Norm
axiom(initially(emp_exp(sanction)),[]).
axiom(initially(nor_exp(sanction)),[]).
% Behavior is conditional on expectations
axiom(precondition(E, emp_exp(E))).
axiom(precondition(E, nor_exp(E))).
% Behavior is conditional on world states (primary norm)
axiom(precondition(get_bin_in, trash_collected)).
% Behavior is conditional on world states (sanctioning norm)
axiom(precondition(sanction, trash_collected)).
% Behavior is conditional on primary nor_exp (sanctioning norm)
axiom(precondition(sanction, nor_exp(get_bin_in))).
```

Querying $abdemo([happens(sanction, t)], R)$. or $abdemo([happens(get\_bin\_in, t)], R)$. returns explanations that appropriately reflect the presence or absence of expectations, as well as any modifications to preconditions[3]. For example, when both the primary norm and sanctioning norm are active, explanations for a sanction may be that one did not take their bin back in, or, that the sanction occurred without (normative) 'justification'. Without justification (at least from the perspective of the agent) is particularly interesting, as it can inform subsequent agent behaviour, potentially prompting the agent to infer or construct additional rules that would render the sanction normatively justified, or to revise the existing normative model to account for the gap.

```
R = [[happens(sanction,t), happens(put_bin_out,p26), happens(collect_trash,p24)],
[before(p26,t), before(p24,t), before(p25,t),  before(p26,p25),  before(p26,p24),
before(p24,p25), before(p23,t)]]
R = [[happens(sanction,t)], []]
```

Importantly, conditionality is a central to Bicchieri's formalization; by incorporating precondition constraints into the abduced narrative, the generated explanations align with the conditional structure of norms and sanctions, and thereby Bicchieri's formalization. Thus, explanations about events and states may be generated to unpack why a sanction occurred. Applying our revised Abductive Event Calculus within this normative domain, each hypothesized event, such as a sanction, is introduced only after its causal and temporal preconditions are satisfied, maintaining relational temporal structure and allowing a clear, constraint-driven account of normative reasoning in the domain.

In this example, however, social expectations align, remain static, and there are no competing or subjective expectations; in our discussion and future work, we explore how the framework can accommodate these more complex normative scenarios.

[3] Code Available at: https://github.com/Nathanlloyd7/Normative-Explanations

5 Discussion

The ability to generate causally grounded explanations is central to reflective reasoning and norm-aware decision-making. Normative reasoning requires more than detecting whether obligations hold or violations occur; it requires explaining how such states arise from temporally ordered and causally supported actions. The extension to the Abductive Event Calculus introduced in this work facilitates the generation of explanations that consider both preconditions and temporal ordering as constraints on the abduced residue. Such capacities would allow agents not only to comply with prevailing norms, but also to critically examine and evaluate them [20], and also to generate plans for norm change as per the original use-case of Shanahan's AEC as a partial-order planner. Crucially, this explanatory structure is not specific to normative contexts; as demonstrated through the Chemical Plant example. Any domain in which behaviour must be interpreted through its enabling conditions; whether institutional, social, or individual, can be represented within the same justification-first framework. The same justification-first abductive mechanism that traces norm compliance can also be used to infer the latent beliefs, intentions, or goals underlying observed behaviour, thereby providing a formal basis for Theory of Mind reasoning if such a domain were to be specified.

The present work demonstrates the generation of causally grounded explanations, however, the examples considered assume a single-agent narrative under conditions of complete information. In our prior work, we explored extensions to the Event Calculus to support representation for self-other distinctions and perspective-taking [24]. Such perspective-taking becomes crucial in multi-agent environments characterised by incomplete or asymmetric information, where agents act on partial beliefs and normative evaluations may therefore be ambiguous. For instance, one may need to distinguish between the questions: *Did the agent violate the norm?* and *Did the agent believe they were compliant?* Normative judgment in such contexts depends not only on objective states of the world but also on the agent's subjective epistemic position. Incorporating perspective-taking enables epistemically sensitive reasoning, supporting more nuanced responses such as contextualised or graduated sanctions that account for excusable transgressions rather than indiscriminate retaliation. Moreover, justification structures may differ across agents, suggesting that permissibility may arise either from the formal normative rules themselves or from the subjective information available to particular agents. Extending the present justification-first framework to incorporate such epistemic distinctions would therefore enable richer multi-agent explanation and evaluation.

Furthermore, following our proposed extension for perspective-taking [24], recent work addressed grounding abductive explanations in perspective and an evolving context [23]. Shanahan's AEC, as well as the extensions presented in this work are detached from an agent's experience, an explanation or plan is generated objectively. This means that abduced explanations did not consider the dynamic context that arises from declared observations ($happens(E, T)$), and the subsequently deduced fluents ($holdsAt(F, T)$) from said observations. This work

contextualized explanations by expanding the top-level call to $abdemo(Goals, Residue)$ to instead explain events within a window; given as $explain(Start_t, Residue, Goal)$. The wrapper supports querying for an abduced narrative that is conditioned not only on the goal time but also on the agent's initial epistemic state and the set of events observed in the intervening interval. The resulting residue thus represents a hybrid explanation, combining inferred (abduced) events with explicitly observed ones ($happens(E, T)$). This prior work did not incorporate causal and temporal preconditions directly into the abductive reasoning process, a limitation that the present work sought to address.

6 Future Work

Although the specified domains help to constrain the generated explanations, this is only half of the task. Given multiple explanations, the complementary tasks of selection, filtering, and evaluation are not yet addressed. Hypothesis selection is typically guided by criteria such as likelihood, simplicity, coherence, or explanatory strength[43,36,30]. A notable contribution in this regard comes from Lipton [19], who, building upon Harman's work, argued for the need of a clear model or framework to identify "good" explanations; whether they be likely or lovely, or both. Indeed, "there will be several hypotheses which might explain the evidence, so one must be able to reject all such alternative hypotheses before one is warranted in making the inference" [14].

Nor once an explanation selected do we include methods for integrating accepted abduced states and events back into the narrative. This situates our work firmly within explanation generation, without yet, full belief revision, indeed, this is is a problem of knowledge assimilation [16]. The work by Bracciali and Kakas [8], as well as Knowledge Goals and Plan model of agency [15] describes how abduced states and events may be assumed into the knowledge base via $assume\_happens(E, T)$ and $assume\_holds(F, T)$. These distinctions, away from the generic axioms $happens(E, T)$ and $holdsAt(F, T)$ enables hypotheses to be provisionally incorporated into an agent's knowledge base and later revised if new information becomes available. This approach is especially appealing in social environments, where knowledge about others' actions and intentions is often incomplete and assumptions made are subject to change. In future work, we anticipate adopting similar semantics to distinguish between observed events and fluents, which are assumed to be true and immutable, and assumed events and fluents, which may be revised as new information becomes available. We may further introduce a layer of $assumed\_at$, so that even when a prior assumption has been revised, it can persist as an explanatory artifact, helping to account for behaviors it may have influenced.

Finally, if no explanation can be derived, the framework offers no mechanism for inspecting or revising the domain theory itself; the underlying rules cannot be directly modified, evaluated, or systematically compared. Incorporating such capabilities would enable a richer form of inquiry, supporting reflective reasoning about the adequacy of the knowledge base. For instance, if the knowledge base K

fails to account for some observation E, the failure to generate an explanation may signal that the domain theory is incomplete. In such a case, it may be necessary to introduce a new rule R_{new} (extending the inferential structure) and/or a new causal factor C_{new} (extending the ontology), yielding an expanded theory K' such that:

$$K' = K \cup \triangle \quad \text{with} \quad K' \models E$$

Here, $\triangle$ comprises additional causal laws and/or axioms involving newly introduced domain entities. This formalizes the intuition that extending the domain with a new causal dependency or rule can render the observation explainable, but also that explanatory success may require enlarging the representational vocabulary of the domain. However, enabling such theory revision raises substantial conceptual and computational challenges [40], in particular, the possibility of unknown unknowns: factors that are not merely unobserved but not even represented within the model. Unknown unknowns undermine the assumption that explanatory failure can always be remedied by incremental rule addition. As discussed by Barnes et al. [3], systems may inherit structural blind spots or cross-generational design assumptions that constrain what can be recognized as a potential cause in the first place, i.e., being unaware of an action to be represented as an $executable(A)$.

7 Conclusion

In this work, we have argued for the necessity of reflective capabilities within normative systems in order to enable agents to meaningfully interpret and "make sense" of social behaviour. To this end, we extended the Abductive Event Calculus to support the generation of causally grounded, justification-driven explanations. We first illustrated these modifications through an adaptation of Shanahan's canonical chemical plant example, thereby demonstrating that the extended framework preserves standard explanatory reasoning on $holdsAt(F, T)$ but now is also capable of explaining events themselves $happens(E, T)$. We then demonstrated this extension within a normative scenario, drawing on Bicchieri's formalisation of norms and the domain extension of the Event Calculus, the Expectation Event Calculus. This extension enables the generation of explanations that are not only causal, but also normative in character, accounting for norm compliance, violation, and sanctioning, with feasible extensions demonstrating norm activation or construction. In this way, empirical and normative expectations can be operationalised to represent conditional obligations and associated enforcement mechanisms. Several avenues for future work remain. These include scaling the framework to multi-agent settings, refining the interaction between empirical and normative expectations and explanation selection and assimilation.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Alberti, M., Chesani, F., Gavanelli, M., Lamma, E., Mello, P., Torroni, P.: Verifiable agent interaction in abductive logic programming: the SCIFF framework. ACM Transactions on Computational Logic (TOCL) **9**(4), 1–43 (2008)
2. Axelrod, R.: An evolutionary approach to norms. American political science review **80**(4), 1095–1111 (1986)
3. Barnes, C.M., Esterle, L., Brown, J.N.: "When you Believe in Things that you don't Understand": the Effect of Cross-Generational Habits on Self-Improving System Integration. In: 2019 IEEE 4th International Workshops on Foundations and Applications of Self* Systems (FAS* W). pp. 28–31. IEEE (2019)
4. Bellucci, F., Pietarinen, A.V.: Peirce's Abduction. In: Handbook of abductive cognition, pp. 7–20. Springer (2023)
5. Bendor, J., Swistak, P.: The Evolution of Norms. American Journal of Sociology **106**(6), 1493–1545 (2001)
6. Bicchieri, C.: The grammar of society: The nature and dynamics of social norms. Cambridge University Press (2005)
7. Bochman, A.: A causal theory of abduction. Journal of Logic and Computation **17**(5), 851–869 (2007)
8. Bracciali, A., Kakas, A.C.: Frame consistency: computing with causal explanations. In: NMR. pp. 79–87 (2004)
9. Bylander, T., Allemang, D., Tanner, M.C., Josephson, J.R.: The computational complexity of abduction. Artificial intelligence **49**(1-3), 25–60 (1991)
10. Cranefield, S.: Agents and expectations. In: International Workshop on Coordination, Organizations, Institutions, and Norms in Agent Systems. pp. 234–255. Springer (2013)
11. Douven, I.: Abduction. In: Zalta, E.N. (ed.) The Stanford Encyclopedia of Philosophy. Metaphysics Research Lab, Stanford University, Summer 2021 edn. (2021)
12. Duede, E., Evans, J.: The social abduction of science. arXiv preprint arXiv:2111.13251 (2021)
13. Eshghi, K.: Abductive Planning with Event Calculus. In: ICLP/SLP. pp. 562–579 (1988)
14. Harman, G.H.: The inference to the best explanation. The philosophical review **74**(1), 88–95 (1965)
15. Kakas, A., Mancarella, P., Sadri, F., Stathis, K., Toni, F.: The KGP model of agency. In: Proceedings of the 16th European Conference on Artificial Intelligence. p. 28–32. ECAI'04, IOS Press, NLD (2004)
16. Kakas, A.C., Kowalski, R.A., Toni, F.: Abductive logic programming. Journal of logic and computation **2**(6), 719–770 (1992)
17. Kate, R.J., Mooney, R.J.: Probabilistic abduction using markov logic networks. In: IJCAI-09 Workshop on Plan, Activity, and Intent Recognition (PAIR'09). pp. 22–28 (2009)
18. Kowalski, R., Sergot, M.: A logic-based calculus of events. New generation computing **4**, 67–95 (1986)
19. Lipton, P.: Inference to the Best Explanation. Routledge, London, 2nd edn. (2004), first edition published 1991
20. Lloyd, N., Lewis, P.R.: Towards reflective normative agents. In: Conference of the European Social Simulation Association. pp. 587–599. Springer (2023)
21. Lloyd, N., Lewis, P.R.: Incorporating social expectations into the expectation event calculus. In: ALIFE 2024: Proceedings of the 2024 Artificial Life Conference. MIT Press (2024)

22. Lloyd, N., Lewis, P.R.: Empirical expectations and coordination games. In: 2025 IEEE International Conference on Autonomic Computing and Self-Organizing Systems Companion (ACSOS-C). pp. 38–45. IEEE (2025)
23. Lloyd, N., Lewis, P.R.: Multi-Perspective Explanations for Multi-Agent Systems. In: 2025 IEEE International Conference on Autonomic Computing and Self-Organizing Systems Companion (ACSOS-C). pp. 106–111. IEEE (2025)
24. Lloyd, N., Lewis, P.R.: Why Was I Sanctioned? In: Proceedings of 1st Workshop on Advancing Artificial Intelligence through Theory of Mind. pp. 154–158 (2025), https://arxiv.org/abs/2505.03770
25. Magnani, L.: Abduction, Reason, and Science: Processes of Discovery and Explanation. Kluwer Academic/Plenum Publishers, New York (2001)
26. Magnani, L.: Handbook of abductive cognition. Springer (2023)
27. Malle, B.F., Bello, P., Scheutz, M.: Requirements for an artificial agent with norm competence. In: Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society. pp. 21–27 (2019)
28. Mazur, L.B., Sticksel, I.K.: An empirical study of psychology and logic. Abduction and belief as normalizing habits of positive expectation. New Ideas in Psychology **63**, 100874 (2021)
29. Nikiforakis, N.: Punishment and counter-punishment in public good games: Can we really govern ourselves? Journal of Public Economics **92**(1-2), 91–112 (2008)
30. Niño, D.: Peircean pragmatism and inference to the best explanation. In: XIth International Conference on Pragmatism (2009)
31. Ostrom, E., Walker, J., Gardner, R.: Covenants with and without a sword: Self-governance is possible. American political science Review **86**(2), 404–417 (1992)
32. Paavola, S.: Abduction through grammar, critic, and methodeutic. Transactions of the Charles S. Peirce Society **40**(2), 245–270 (2004)
33. Paavola, S.: Hansonian and Harmanian abduction as models of discovery. International Studies in the Philosophy of Science **20**(01), 93–108 (2006)
34. Paavola, S.: Abduction as a Logic of Discovery. In: Handbook of Abductive Cognition, pp. 43–60. Springer (2023)
35. Pareschi, R.: Abductive reasoning with the GPT-4 language model: Case studies from criminal investigation, medical practice, scientific research. Sistemi intelligenti **35**(2), 435–444 (2023)
36. Paul, G.: Approaches to abductive reasoning: an overview. Artificial intelligence review **7**(2), 109–152 (1993)
37. Peirce, C.S.: Collected papers of charles sanders peirce, vol. 5. Harvard University Press (1934)
38. Peirce, C.S.: The essential peirce, volume 2: Selected philosophical writings (1893-1913), vol. 2. Indiana University Press (1992)
39. Peirce Charles, S.: The essential peirce 2, peirce edition project (1998)
40. Reyd, S., Diaconescu, A., Dessalles, J.L., Esterle, L.: A roadmap for causality research in complex adaptive systems. In: 2024 IEEE International Conference on Autonomic Computing and Self-Organizing Systems Companion (ACSOS-C). pp. 35–40. IEEE (2024)
41. Sadri, F., Stathis, K., Toni, F.: Normative KGP agents. Computational & Mathematical Organization Theory **12**(2), 101–126 (2006)
42. Sadri, F., Toni, F., Stathis, K.: Normative KGP Agents: A preliminary report. In: NORMAS. pp. 85–96 (2005)
43. Schurz, G.: Patterns of abduction. Synthese **164**, 201–234 (2008)
44. Schurz, G.: Theory-generating abduction and its justification. In: Handbook of Abductive Cognition, pp. 181–208. Springer (2023)

45. Sengupta, A., Cranefield, S., Pitt, J.: Generalising Axelrod's Metanorms Game Through the Use of Explicit Domain-Specific Norms. In: International Workshop on Coordination, Organizations, Institutions, Norms, and Ethics for Governance of Multi-Agent Systems. pp. 21–36. Springer (2023)
46. Shanahan, M.: Solving the frame problem: a mathematical investigation of the common sense law of inertia. MIT press (1997)
47. Shanahan, M.: An abductive event calculus planner. The Journal of Logic Programming **44**(1-3), 207–240 (2000)
48. Shanahan, M.: The Event Calculus Explained. In: Artificial intelligence today: Recent trends and developments, pp. 409–430. Springer (2001)
49. Tan, K., Qi, Z., Zhong, J., Xu, Y., Zhang, W.: KN-VLM: KNowledge-guided Vision-and-Language Model for visual abductive reasoning. Multimedia Systems **31**(2), 146 (2025)
50. Thagard, P.: Abductive inference: From philosophical analysis to neural mechanisms. Inductive reasoning: Experimental, developmental, and computational approaches pp. 226–247 (2007)
51. Thagard, P.: Can ChatGPT make explanatory inferences? Benchmarks for abductive reasoning. In: Abductive Minds: Essays in Honor of Lorenzo Magnani-Volume 1, pp. 189–218. Springer (2025)