CAVA: Contextual Argumentation for Value-based Assessment

Jack McKinlay[1][0000−0001−9822−8166], Janina A. Hoffmann[2][0000−0002−6246−2724], Andreas Theodorou[0000−0001−9499−1535], and Marina De Vos[1][0000−0003−3583−7671]

[1] Department of Computing Science, University of Bath, UK
[2] Department of Psychology, University of Bath, UK

Abstract. When implementing autonomous agents, it is critical that they make decisions aligned with human social values. In addition to modelling these values, value-based reasoning in autonomous agents requires context-aware decision-making that accounts for multiple valid strategies and potentially conflicting values. In this paper, we address these challenges by introducing CAVA: Contextual Argumentation for Value-based Assessment. CAVA is a methodology based on formal argumentation that models agents' values and value priorities in dynamic contexts, as well as their arguments for decisions. The semantics of the framework provides the set of acceptable arguments that can be used to identify credulously and sceptically acceptable options based on the agent's values, along with explanations of how these options support the agent's values. CAVA enables the expression of an agent's values and how they are used in decision-making in a form that is interpretable by human and autonomous agents alike.

Keywords: Value-based Reasoning · Contextual Reasoning · Argumentation

1 Introduction

Values serve as drivers in individual and shared decision-making [27]. However, modelling values for use in autonomous decision-making is challenging due to their abstract, cross-contextual nature [16, 19, 25]. A given value like "fairness" can be understood and applied in different scenarios, but what it means to act fairly depends on both the interpreter and the circumstances. While there are significant challenges in modelling values, they are a necessary component in designing automated systems to reflect and promote these values for the benefit of the system's stakeholders.

Previous attempts have been made to have AI systems learn our values through sub-symbolic training methods [9, 11]. However, these methods are criticised for lacking interpretability. Because of the contextual and

subjective nature of values, it is important that we can assess the particular values modelled and how they have been used in reasoning if we want to be able to assess AI alignment, as well as make statements about system behaviour and safety [1, 17, 29].

Given the opacity of sub-symbolic systems, we propose a symbolic framework for value-based reasoning as an alternative that avoids this opacity. In this paper, we explore argumentation frameworks for this task. Argumentation frameworks are a formal system for representing abstract information and resolving conflicts within this information [10]. Through defining arguments and attacks between arguments, argumentation frameworks can assess the acceptability of arguments based on whether they are able to repel attacks against them. Argumentation frameworks' ability to represent abstract information, and the conflicts between elements of that information, makes it well-suited to reasoning over abstract values that may not always be mutually achievable.

In addition, argumentation can be a powerful method for achieving explainability in AI [28, 30], which is vital in auditing the value-based decisions of an autonomous agent. Its applicability to non-monotonic reasoning [8, 13] is well-suited to value-based reasoning across dynamic contexts, as new contexts can change the relevant values. Argumentation can also facilitate collaboration and negotiation between agents with competing interests [5, 12], and support human-AI interaction even in the presence of uncertainty [20].

In this paper, we introduce our argumentation framework: **Contextual Argumentation for Value-based Assessment** (CAVA) as a mechanism for value-based reasoning. CAVA is an argumentation framework that models abstract values across contexts through features, which are observable, measurable aspects of the agent's operating environment that are considered relevant in decision-making. CAVA consists of both formalisms for values and the concepts needed to contextualise them, and a system of rules for supporting inference in value-based decision-making. We illustrate CAVA using a running example based on a smart sprinkler system for watering a garden, and how it resolves priorities in different weather conditions.

The rest of the paper proceeds as follows: In Section 2 we outline the background material. Then, we introduces the CAVA model in Section 3, followed by a discussion on how we model context in Section 4. In Section 5 we explain how to construct a CAVA argument and define attacks between them. Section 6 discusses related research and how CAVA builds on it. Finally, Section 7 consists of a discussion of the work and future directions.

2 Background

2.1 Argumentation

Argumentation frameworks were originally conceived by [10] as a system for modelling argumentation in human reasoning. His underlying principle was "The one who has the last word laughs best.", meaning that the value that repels all counterarguments is victorious.

Dung's framework does not include a fixed definition for what an argument is, beyond 'an abstract entity whose role is solely determined by its relations to other arguments.'. He also does not constrain what it means for one argument to attack another. This means that we can define these concepts to suit our own needs, provided that an attack still represents a binary relationship between arguments. We define CAVA arguments and CAVA attacks in Section 5.

CAVA is based on the two particular argumentation frameworks, presented in Kakas and Moraitis [14] and van der Weide et al. [31]. Kakas and Moraitis [14] and the *Gorgias* framework developed from it [15] allow for modelling priorities between arguments based on the reasoning scenario. However, in *Gorgias* the priorities between arguments must be defined manually and cannot be inferred. The framework in van der Weide et al. [31], in comparison, models hierarchical value-based argumentation. However, van der Weide et al.'s framework has no built-in mechanism to handle changes in values or conflicts between contexts. By taking elements from both of these frameworks and extending them, we have developed CAVA to achieve contextual argumentation with dynamic option evaluation.

2.2 Values & Value Structures

In this paper, we focus on values as defined by Schwartz [26], a widely cited framework backed by significant empirical research. Schwartz defines values as abstract, cross-contextual ideas, modelling 10 continuous value types based on empirical studies across 20 countries. The continuity arises because these types can be divided into subgroups, and contextual interpretations can span multiple types. E.g. family loyalty could fall under either 'tradition' (respect for tradition) or 'security' (family security).

Values must be applied to decision-making concretely; moving from abstract concept to instantiated behaviour [18]. Many authors model this relation hierarchically [2, 18, 21, 22]. E.g. Van de Poel shows how abstract animal welfare values translate into specific concerns like litter attention, which are grounded further into actionable requirements such as nests per hen.

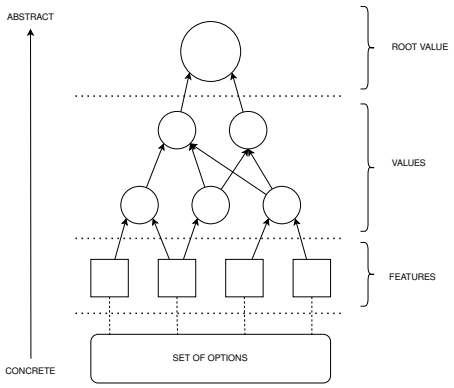

Fig. 1. Structure of the CAVA model for value-based reasoning using argumentation. Features (squares) are linked to values (circles) by influences (arrows). Options chosen by an agent can potentially impact any feature, which are associated with perspectives (not visually represented) over the outcomes of these options. These features influence values, which in turn influence more abstract values. Values also have perspectives between influencing features and values. At the highest level of abstraction we reach the root value, where values and their relationships to each other can not be/do not need to be explicitly modelled further.

Hierarchical models capture both the relation between abstract values and concrete interpretations, and the influences values have over each other [18, 24, 27]. van der Weide et al. [32] uses these influences to connect decision outcomes to grounded elements at the hierarchy's base, then evaluates outcomes by propagating upward through the hierarchy. CAVA extends this approach to argumentation.

3 CAVA Model for Value-based Reasoning

The CAVA model for value-based reasoning, $\mathcal{M} := \{V, F, O, I, P, \mathcal{F}, \mathcal{O}\}$ is a 7-tuple consisting of 5 sets of objects and 2 functions. Intuitively: values (V) are abstract, desirable concepts; features (F) are observable elements that are relevant to values; options (O) are choices available to the agent; influences (I) define connections between features and values; and perspectives (P) indicate priorities over feature states or values. The feature evaluation function ($\mathcal{F}$) and option outcome function ($\mathcal{O}$) map features and feature-option pairs, respectively, to states of the features. We will expand on each of these concepts in their relevant definitions.

The interaction between these components is illustrated in Fig. 1. Values, features, and influences are used to construct a directed acyclic graph. By using influences to connect features to values, and values to more ab-

stract values, we build a value hierarchy. At the highest level of abstraction, we capture the point where distinct values cannot be, or no longer need to be, modelled. This allows us to avoid stating whether or not there is a definitive single value that influences decision-making.

The option outcome function connects options and features to results of choosing options on features. Perspectives are used to define priorities in the constructed value hierarchy, reflecting preferences for different feature states (feature-level perspectives), or preferences between competing influencing features or values (value-level perspectives). Taken together, our CAVA model allows us to represent complex value hierarchies in argumentation that we can use to evaluate the different options available to an agent.

Definition 1 (Values). *In CAVA, we define* ***values*** $v \in V$ *as abstract decision-making factors, that cannot be modelled as observable elements that would be impacted by the decision.*

Definition 2 (Root Value). *We use a special type of value to represent the root node in the directed acyclic graph formed from the CAVA model, which we label the* ***root value*** $rv \in V$*.*

The root value serves as the most abstracted level of desires modelled by values in a CAVA model, and is the destination of all influences that do not terminate at another non-root value. The root value indicates where value abstraction terminates in a CAVA model, due to further value abstraction being either not needed or not possible. The root value can act as a proxy for a single ultimate value, like utility, or represent multiple incommensurable values as a single object in the model.

Because values in CAVA are abstract and immeasurable, we need to link them to contextual proxies that we can observe and affect directly. This is done through features.

Definition 3 (Features). ***Features*** $f \in F$ *are the contextual proxies that allow us to link the outcomes of options to values through perspectives. Features correspond to aspects of the environment or internal to the agent that can be measured and compared against alternative states of that feature.*

Definition 4 (Feature States). *A* ***feature state*** *is used to define different comparable states of a given feature. The* ***feature evaluation function*** $\mathcal{F} : F \to 2^{FS}$*, where* FS *is the set of all possible feature states, is a mapping such that* $\mathcal{F}(f)$ *for* $f \in F$ *(equivalently denoted* FS_f*) corresponds*

to the set of feature states that can be used for comparing possible states of feature f. For example, if feature f is 'amount of fuel in the tank' then $\mathcal{F}(f)$ could be the set of non-negative real numbers, up to the capacity of the fuel tank. We define the set of all feature states associated with a given CAVA *model as* $FS_{\mathcal{M}} = \bigcup_{f \in F} FS_f$.

To reason about possible alternative feature states and hence the satisfaction of our values, we need a mechanism to reflect how one decision may affect multiple features. When planning cooking, for example, selecting different ingredients can lead to both different costs and different nutritional profiles for the final meal plan. This is where we use options.

Definition 5 (Options). *Options $o \in O$ are choices available to the decision-maker that affects the features.*

Definition 6 (Option Outcome Function). *We map features to feature states using the* ***option outcome function****, $\mathcal{O} : O \times F \to FS$.*

We require that the option outcome function be consistent with the feature evaluation function, that is $\mathcal{O}(o, f) \in \mathcal{F}(f) \, \forall (o, f) \in O \times F$. Thus, for each feature f, $\mathcal{O}$ selects a permissible feature state from the set specified by $\mathcal{F}$.

A given option-feature pair only maps to one feature state. Otherwise, this would allow a given option to cause multiple outcomes on the same feature simultaneously. If this type of mapping seems essential to the model, then the feature should be split into more features.

Definition 7 (Influences). *We model the relationship between features and values, and between values and values, using influences. We define influences as a binary relation $I \subseteq (F \cup V) \times V$, where $(x, y) \in I$ denotes that x influences y such that:*

- ***No Irrelevant Features****: All features must influence at least one value.* $\forall \, f \in F \;\; \exists \, v \in V \cdot (f, v) \in I$
- ***No Contextless Values****: All values must be influenced by at least one feature or value.* $\forall \, v \in V \;\; \exists \, x \in (F \cup V) \cdot (x, v) \in I$.
- ***Acyclicity****: No value influences itself (directly or indirectly).* $\nexists v \in V \cdot (v, v) \in I^+$ *where I^+ is the transitive closure of I.*
- ***Root Value as Sink****: The root value rv can only be a destination, never a source.* $\nexists y \in V \cdot (rv, y) \in I$.
- ***Homogeneous Influences****: A value is influenced either by features or by other values, never both.* $\forall v \in V$ *if* $\exists \, f \in F \cdot (f, v) \in I$ *then* $\nexists v' \in V \cdot (v', v) \in I$.

– ***Termination at Root****: Any value that does not influence another non-root value must influence the root value.*
$\forall\, v \in V \setminus \{rv\} \cdot \nexists\, \{v' \in V \setminus \{rv\} \cdot (v, v') \in I\}\ \exists (v, rv) \in I$*. Similarly, no value influences both another non-root value and the root value.* $\nexists v, v' \in V \setminus \{rv\} \cdot (v, v') \in I \wedge (v, rv) \in I$.

These constraints ensure that the CAVA model forms a directed acyclic graph (DAG) with the root value as the unique sink, and with clear stratification between feature-influenced values and value-influenced values. The DAG structure prevents cycles in reasoning.

Influences indicate which features are relevant to a value, and how values affect other values. However, we need another mechanism to model how a value rates possible feature states, or how it might prioritise between different features or values influencing it. We define these priorities using perspectives.

Definition 8 (Perspectives). *Perspectives define preference orderings within the CAVA model and serve two distinct roles:*

Feature-level perspectives *specify strict partial orderings over feature states for features that influence a given value, and are used when comparing a given feature $f \in F$ in different states. For each value $v \in V$ and feature $f \in F$ where $(f, v) \in I$, we can define:*

$$p_v^f \subseteq FS_f \times FS_f$$

where FS_f is the set of possible states for feature f, and $(s_1, s_2) \in p_v^f$ denotes that state s_1 is preferred to state s_2 from the perspective of value v. We write $s_1 \succ_v^f s_2$ when $(s_1, s_2) \in p_v^f$.

Value-level perspectives *specify strict partial orderings over the features and values that influence a given value, and are used when comparing different features or different values. For each value $v \in V$, we can define:*

$$p_v \subseteq X_v \times X_v$$

where $X_v = \{x : x \in (F \cup V) \wedge (x, v) \in I\} \cup \{(f, s) : f \in F, (f, v) \in I, s \in FS_f\}$ is the set of elements that may be compared from the perspective of v, including features, values, and feature-state pairs. For $x_1, x_2 \in X_v$, we write $x_1 \succ_v x_2$ when $(x_1, x_2) \in p_v$.

The collection of all perspectives in a CAVA model is denoted $P = \{p_v^f, p_v \mid v \in V, f \in F\}$.

Perspectives are uniquely defined: each value-feature pair has at most one feature-level perspective, and each value has at most one value-level perspective [3].

The ordering p_v determines preferences between option outcomes: $(f_1, f_2) \in p_v$ implies $\mathcal{O}(o, f_1) \succ_v \mathcal{O}(o, f_2)$ for all $o \in O$; $((f_1, s_1), (f_2, s_2)) \in p_v$ where $s_i \in FS_{f_i}$ implies $\mathcal{O}(o, f_1) = s_1 \succ_v \mathcal{O}(o^*, f_2) = s_2$ when options produce those states; $((f_1, s_1), f_2) \in p_v$ implies $\mathcal{O}(o, f_1) = s_1 \succ_v \mathcal{O}(o^*, f_2)$ for any o^* producing s_1; $(f_1, (f_2, s_2)) \in p_v$ implies $\mathcal{O}(o, f_1) \succ_v \mathcal{O}(o^*, f_2) = s_2$ for any o^* producing s_2; and $(v_1, v_2) \in p_v$ where $v_1, v_2 \in V$ directly expresses priority of v_1 over v_2. These comparisons may be combined whilst maintaining transitivity.

Partial orders permit incommensurability, reflecting value incommensurability [6, 7, 23]. The root value's perspective p_{rv} may be empty or express limited preferences to avoid trivially defeating arguments.

Running Example Consider an automated sprinkler system for a garden. We design the system to promote keeping the garden watered (value of *functioning*) but also avoid wasting water (value of *conservation*). We can represent these values through the observable features of whether the garden is watered or not, and the amount of water used, respectively. The former feature influences functioning, while the latter feature influences conservation.

Our options are a simple binary choice between activating or not activating the sprinkler. If the sprinkler activates then the garden will be watered and some water will be used. If it does not activate then the garden is not watered and no water is used. From the perspective of the value of functioning, the garden being watered is preferable to it not being watered. From the perspective of conservation, using no water is preferable to using some water.

Let us assume that the fundamental behaviour of the sprinkler should prioritise functioning to keep the garden hydrated and healthy. We can model this using a value-level perspective on the roof value that prioritises the value of functioning over the value of conservation.

From this information, we can define our CAVA model for the system[4]:

[3] A lack of a perspective for a given value-feature pair or value is equivalent to no preferences.

[4] For conciseness and readability, we used labels for the values, features and options

$$\mathcal{M} := \{V, F, O, I, P, \mathcal{F}, \mathcal{O}\} :$$
$$V = \{v_1(\text{Functioning}), v_2(\text{Conservation})\}$$
$$F = \{f_1(\text{Watered State}), f_2(\text{Water Use})\}$$
$$O = \{o_1(\text{Activate}), o_2(\text{No Activate})\}$$
$$I = \{(f_1, v_1), (f_2, v_2), (v_1, rv), (v_2, rv)\}$$
$$P = \{p_{v_1}^{f_1}, p_{v_2}^{f_2}, p_{rv}\}$$

Such that:

$$\mathcal{F}(f_1) = \{watered, not\_watered\}, \mathcal{F}(f_2) = \{some, none\},$$
$$p_{v_1}^{f_1} = \{(watered, not\_watered)\}, p_{v_2}^{f_2} = \{(none, some)\}, p_{rv} = \{(v_1, v_2)\}$$
$$\mathcal{O}(o_1, f_1) = watered, \mathcal{O}(o_1, f_2) = not\_watered,$$
$$\mathcal{O}(o_1, f_2) = some, \mathcal{O}(o_2, f_2) = none,$$

4 Contexts

We understand contexts as the relevant set of circumstances considered during decision-making. The components defined in the previous section allow us to structure value-based decision-making in static contexts. However, values are not static in nature, but are instead sensitive to the context in which they are instantiated. This necessitates the formalisation of contextualised value-reasoning. To that effect, we define contexts as a pair containing the context-specific handling of value reasoning and the activation conditions for the context. The former are represented as two (partial) CAVA models that indicate on the one hand what needs to become part of any CAVA model observing this context and what becomes irrelevant. For an agent, the activation conditions can be related to the environment the agent is operating in.

Definition 9 (Context). *A **context** is a pair $c := (\Delta, R) \in C$. Δ is defined as a pair (Δ^+, Δ^-) containing modifications to CAVA models such that:*

- *Δ^+ is a (partial) CAVA model that indicates additional elements (if not already in set or function mapping) to be added to the CAVA model if the context becomes active.*
- *Δ^- is a (partial) CAVA model that indicates the removal of elements (if in set or function mapping) from the CAVA model if the context becomes active.*

*R is a set of propositions that evaluate to true or false. A context c is **active** if $\forall r \in R, r \equiv T$.*

In practice, there is rarely a single context active in decision-making, and different contexts may imply multiple competing considerations. This can lead to conflicts between contexts.

Definition 10 (Conflicts). *A **conflict**, denoted $c_1 <> c_2$, between contexts $c_1 = (\Delta_1, R_1)$ and $c_2 = (\Delta_2, R_2)$ exists if:*

- ***Direct negation**: $\exists X \in \Delta_1^+ \wedge \neg X \in \Delta_2^-$ for some component X*
- ***Feature-level perspective conflict**: $\exists\, p_v^f \in \Delta_1^+ \wedge p_v^{f'} \in \Delta_2^+ \cdot f = f' \wedge p_v^f \neq p_v^{f'}$.*
- ***Value-level perspective conflict**: $\exists\, p_v \in \Delta_1^+ \wedge p'_v \in \Delta_2^+ \cdot p_v \neq p'_v$*[5]
- ***Mapping conflict**: $\exists\, \mathcal{F} \in \Delta_1^+ \wedge \mathcal{F}' \in \Delta_2^+ \cdot \mathcal{F}(f) \neq \mathcal{F}'(f)$. Or similarly, $\exists\, \mathcal{O} \in \Delta_1^+ \wedge \mathcal{O}' \in \Delta_2^+ \cdot \mathcal{O}(o, f) \neq \mathcal{O}'(o, f)$*

A well-defined context must be internally consistent: the modifications in Δ cannot conflict with themselves. Conflicts between contexts are resolved using the refinement relationship, where refinement is used to identify the priority between contexts based on the idea that more specific contexts refine more general contexts [15].

Definition 11 (Context Refinement). *A context $c_2 \in C$ is a **refinement** of context $c_1 \in C$, written $c_1 \sqsubset c_2$, if $R_1 \in c_1 \subset R_2 \in c_2$. We say c_1 is the **refined** context and c_2 is the **refining** context.*

To model the case where no specific contexts are relevant to decision-making, while still defining a CAVA model for generic decision-making, we use default contexts.

Definition 12 (Default Context). *A **default context** is defined as $c_\emptyset = \{\Delta_{c_\emptyset}, \emptyset\} \in C \cdot \Delta_{c_\emptyset} = (\Delta^+_{c_\emptyset}, \emptyset)$. The **default CAVA model** $\mathcal{M}_{c_\emptyset}$ is defined by the default context $c_\emptyset$, such that $\Delta^+_{c_\emptyset} = \mathcal{M}_{c_\emptyset}$.*

*The **empty CAVA model**, $\mathcal{M}_\emptyset := \{\emptyset, \emptyset, \emptyset, \emptyset, \emptyset, \emptyset, \emptyset\}$, is initialised by a specific instance of a default context called the **empty context**, $\{(\emptyset, \emptyset), \emptyset\}$.*

Since $c_\emptyset$ has no requirements, it is always active and refined by all other contexts (as $\emptyset \subset R$ for any non-empty R). Default contexts contain only addition modifications, i.e. $\Delta^-_{c_\emptyset} = \emptyset$: since default contexts are designed to build on an empty CAVA model, removal modifications would be redundant.

[5] It may be the case that two perspectives on the same value differ without being incompatible: consider $(a, b) \in p_v$ and $(c, a) \in p'_v$. These could be combined to produce a perspective $p^*_v := ((c, a), (a, b))$. We do not address this in this paper, but note it for future work.

We can combine contexts using **context addition**. This would normally occur in response to requirements becoming active. To implement context addition reliably, we must restrict the set of contexts that context addition can be applied to. [6]

Definition 13 (Context Addition). *Define context addition,* $\oplus : C \times C \to C$*, as a partial operator defined only when* c_i *and* c_j *are either non-conflicting or* $c_i \sqsubset c_j$*.*

Context addition $c_i \oplus c_j := \{(\Delta^+_{c_{ij}}, \Delta^-_{c_{ij}}), R_i \cup R_j\}$*, where* $\Delta^+_{c_{ij}}$ *and* $\Delta^-_{c_{ij}}$ *are constructed as follows.*

Additions $\Delta^+_{c_{ij}}$ *is constructed by:*

- *For non-conflicting elements:* $\Delta^+_{c_i} \cup \Delta^+_{c_j}$
- *For conflicting elements, where* $x \in S \subseteq \Delta^+_{c_i}$ *and* $x' \in S' \subseteq \Delta^+_{c_j}$ *for some* $S, S' \in \{V, F, O, I, P\}$*: retain the element from the refining context.*
- *For conflicting mappings* $\phi \in \Delta^+_{c_i}$ *and* $\phi' \in \Delta^+_{c_j}$ *where* $\phi, \phi' \in \{\mathcal{F}, \mathcal{O}\}$*: retain the mapping from the refining context.*
- *For cross-modification conflicts, where* $x \in S \subseteq \Delta^+_{c_i}$ *and* $x \in S \subseteq \Delta^-_{c_j}$*, or* $\phi \in \Delta^+_{c_i}$ *and* $\phi \in \Delta^-_{c_j}$*: if* $c_j \sqsubset c_i$ *insert* x/ϕ *into* $\Delta^+_{c_{ij}}$*; if* $c_i \sqsubset c_j$ *insert* x/ϕ *into* $\Delta^-_{c_{ij}}$*.*

Removals $\Delta^-_{c_{ij}}$ *is constructed by:*

- $\Delta^-_{c_{ij}} = \Delta^-_{c_i} \cup \Delta^-_{c_j}$*, additionally including any elements or mappings moved from* $\Delta^+_{c_{ij}}$ *by cross-modification conflict resolution.*

We can combine sets of contexts together, and hence construct a CAVA model, through context addition. When combining sets of contexts this way, we need to ensure that we can add together the modifications across all contexts in the set to construct a complete CAVA model. To do this, we need to define a valid set of contexts.

Definition 14 (Valid Set of Contexts). *A set of contexts* $\mathcal{C}$ *is* ***valid*** *if* $\forall c_1, c_2 \in \mathcal{C} \cdot c_1 <> c_2$*, either* $c_1 \sqsubset c_2$ *or* $c_2 \sqsubset c_1$*, and the refinement relation* $\sqsubset$ *restricted to* $\mathcal{C}$ *is acyclic.*

By using context addition over a valid set of contexts, we can define a joint context and, in turn, a joint CAVA model.

Definition 15 (Joint Context & Joint CAVA Model). *A* ***joint context*** *is a valid set of contexts,* $\mathcal{C} = \{c_1, c_2, \ldots, c_n\}$*, for which all requirements in each context* $c_i \in \mathcal{C}$ *resolve as true. The* ***joint CAVA model*** $\mathcal{M}_\mathcal{C}$

[6] We defer defining **context subtraction** for future work, as it is not necessary to construct CAVA models.

is constructed by context addition, starting from an empty CAVA *model* $\mathcal{M}_\emptyset$ *and adding each individual context in the joint context such that context addition between the current and additional context is defined, until all contexts have been added. The joint* CAVA *model* $\mathcal{M}_\mathcal{C}$ *is then equal to* $\Delta^+_\mathcal{C}$*, the additional modifications in the combined context.*

Proposition 1. *For any joint context, an order of application of context addition between pairs of contexts always exists, such that a well-defined joint* CAVA *model can always be constructed* [7]*.*

Running Example We model the standard behaviour of our sprinkler using a default context:

$$c_\emptyset = \{\Delta_\emptyset, \emptyset\}$$

Where $\Delta^+_\emptyset \in \Delta_\emptyset = \mathcal{M}$ defined in the example in Section 3, and $\Delta^-_\emptyset \in \Delta_\emptyset = \emptyset$.

As a smart system, our sprinkler system should be able to adapt its behaviour based on its surroundings. In this example, we model the case when it is raining. This can be modelled by introducing a new specific context. When raining, we note that the garden will be watered regardless of whether the sprinkler activates, which means we should not assume functioning is necessary compared to conservation.

This context can be represented as:

$$c_{rain} = \{(\{\mathcal{O}(o_2, f_1) = watered\}, \{p_{rv}\}), Weather == rain\}$$

This context refines the default context $c_\emptyset$, as its requirements are a superset of the empty set. Hence, we apply the modifications from the refining context. In the joint context $\mathcal{C}_{rain} = \{c_\emptyset, c_{rain}\}$, modifications $\mathcal{O}(o_2, f_1) = watered$ and $p_{rv} \in \Delta^-_{rain}$ from c_{rain} (the refining context) conflict with $\mathcal{O}(o_2, f_1) = not\_watered$ and $p_{rv} \in \Delta^+_{c_\emptyset}$ from $c_\emptyset$ (the refined context) respectively. We remove the conflicting elements from the refined context and insert the elements in the refining context into our joint context CAVA model $\mathcal{M}_{\mathcal{C}_{rain}}$.

We have now defined all the components needed to demonstrate how we construct CAVA arguments, and how these arguments interact. Through this interaction, we can conclude what options are credulously or sceptically acceptable for an agent based on their modelled values.

[7] A proof is provided in the online appendix: https://tinyurl.com/nf69kvc5

5 Arguments using CAVA

The idea behind a CAVA argument is that we justify an option based on its expected effects on a given feature, and that we see that effect as desirable from the perspective of influenced values. CAVA arguments construct complete justification chains that trace from concrete option outcomes through hierarchical value structures. We evaluate which argument is stronger based on the perspectives we have defined in the CAVA model across competing feature states, features, and values.

Definition 16 (Argument). *Given a joint context $\mathcal{C}$ leading to the CAVA model $\mathcal{M}_\mathcal{C} := \{V, F, O, I, P, \mathcal{F}, \mathcal{O}\}$. An argument is a pair $\mathcal{A} = (A, \mathcal{C})$, where A is a 3-tuple containing:*

1. *an option, $o_i \in O$*
2. *a feature, $f_i \in F$*
3. *an ordered set of features and values that connect feature f_i to the root value rv called an **explanation**, $E := \langle x_0, x_1, \ldots, x_n \rangle : x_0 = f_i, x_n = rv, x_j \in V$ for $1 \leq j \leq n$ and $(x_j, x_{j+1}) \in I$ for $0 \leq j < n$.*

Definition 17 (Attack). *Given a joint context $\mathcal{C}$ leading to the CAVA model $\mathcal{M}_\mathcal{C} := \{V, F, O, I, P, \mathcal{F}, \mathcal{O}\}$. An argument $\mathcal{A}_1 = (A_1, \mathcal{C}) \cdot A_1 = \{o_1, f_1, E_1\}$ **attacks** another argument $\mathcal{A}_2 = (A_2, \mathcal{C})) \cdot A_2 = \{o_2, f_2, E_2\}$ if the following holds:*

- $\exists\, v^* \in V \cdot v^* \in E_1 \wedge v^* \in E_2$*: a shared value exists in the explanations in both arguments, which may be the root value.*
- $\exists p^f_{v^*}$ *for* $f \in \{E_1 \cap E_2\} \vee \exists p_{v^*}$*: there exists either a value-level or feature-level perspective defined on the shared value in $\mathcal{C}$.*
- $\exists\, x_1 \in E_1 \cdot (x_1, v^*) \in I \wedge \exists\, x_2 \in E_2 \cdot (x_2, v^*) \in I$*: there exists a feature/value in the explanations of both arguments that influences the shared value.*
- *The perspective $p^f_{v^*}$ or p_{v^*} does not imply priority for x_2 over x_1:*
  - *If the perspective is feature-level ($x_1 = x_2 = f \in F$): $(s_2, s_1) \notin p^f_{v^*}$ where $s_1 = \mathcal{O}(o_1, f)$, $s_2 = \mathcal{O}(o_2, f)$, and $o_1 \in A_1$ and $o_2 \in A_2$.*
  - *If the perspective is value-level, none of the following hold:*
    Direct comparison $(x_2, x_1) \in p_{v^*}$
    Specific states $((x_2, s_2), (x_1, s_1)) \in p_{v^*}$ *where* $s_1 = \mathcal{O}(o_1, x_1)$, $s_2 = \mathcal{O}(o_2, x_2)$
    Specific vs any state $((x_2, s_2), x_1) \in p_{v^*}$*, where* $s_2 = \mathcal{O}(o_2, x_2)$
    Any vs specific state $(x_2, (x_1, s_1)) \in p_{v^*}$*, where* $s_1 = \mathcal{O}(o_1, x_1)$

In the event of multiple value intersections, potential attacks are checked from least abstract to most abstract. For each intersecting value $v^* \in E_1 \cap E_2$, we check intersections in ascending order of the minimum index of v^* in either explanation. The first instance of a preference for the influencing option outcome/feature/value in $\mathcal{A}_2$ over the influencing option outcome/feature/value in $\mathcal{A}_1$ terminates the attack by $\mathcal{A}_1$ and no further intersections are considered. This holds even if a later intersection would imply a preference for $\mathcal{A}_1$ over $\mathcal{A}_2$.

Definition 18 (Argument Acceptability). *An argument is considered acceptable if it attacks any argument attacking it.* [8]

We say that, given a set of arguments and attack relations $\langle \mathcal{AR}, AT \rangle$ derived from an active CAVA model, an argument $\mathcal{A} \in \mathcal{AR}$ is acceptable given itself if and only if for each argument $\mathcal{B} \in \mathcal{AR}$, if $(\mathcal{B}, \mathcal{A}) \in AT$, then $(\mathcal{A}, \mathcal{B}) \in AT$.

Finally, we reach option acceptability as the conclusions derived using CAVA. Through constructing a CAVA model from active contexts and evaluating arguments from said model, we can assess options that are suitable or ideal for an agent, while identifying the values they support.

Definition 19 (Option Acceptability). *An option o^* is **credulously acceptable** if $\exists\, \mathcal{A} \in \mathcal{AR} \cdot \mathcal{A}$ is acceptable $\wedge\, o^* \in A \in \mathcal{A}$. An option o^* is **sceptically acceptable** if $\forall \mathcal{A} \in \mathcal{AR} \cdot \mathcal{A}$ is acceptable, if $\exists\, o \in A \in \mathcal{A} \cdot o \neq o^*$, then $\exists\, A^* \in \mathcal{A}^* \in \mathcal{AR} \cdot \mathcal{A}^*$ is acceptable $\wedge\, A^* = (A \setminus \{o\}) \cup \{o^*\}$*

A credulously acceptable option is worth considering as it belongs to at least one acceptable argument, but a sceptically acceptable option is ideal as it satisfies all values for which an acceptable argument exists for the given CAVA model.

Running Example We can construct the following arguments for our sprinkler scenario in the default context:

$$\begin{aligned}
\mathcal{A}_1 = (A_1, \mathcal{C}_\emptyset) = (A_1, \mathcal{C}_{rain}) : &\quad A_1 = \{o_1, f_1, E_1\} \\
\mathcal{A}_2 = (A_2, \mathcal{C}_\emptyset) = (A_2, \mathcal{C}_{rain}) : &\quad A_2 = \{o_2, f_1, E_1\} \\
\mathcal{A}_3 = (A_3, \mathcal{C}_\emptyset) = (A_3, \mathcal{C}_{rain}) : &\quad A_3 = \{o_1, f_2, E_2\} \\
\mathcal{A}_4 = (A_4, \mathcal{C}_\emptyset) = (A_4, \mathcal{C}_{rain}) : &\quad A_4 = \{o_2, f_2, E_2\} \\
\mathcal{C}_\emptyset = \{c_\emptyset\} &\quad E_1 = \langle f_1, v_1, rv \rangle \\
\mathcal{C}_{rain} = \{c_{rain}\} &\quad E_2 = \langle f_2, v_2, rv \rangle
\end{aligned}$$

[8] Unlike Dung's definition of acceptability in Section 2, in CAVA we do not define an acceptable argument with respect to a given set S. In CAVA we instead equate the set S with the argument under consideration.

Based on feature-level perspectives, $\mathcal{A}_1$ attacks $\mathcal{A}_2$ without an attack back, as both arguments use the same explanation and hence intersect on v_1. According to $p_{v_1}^{f_1}$, a watered garden ($\mathcal{O}(o_1, f_1) = watered$) is preferable to it not being watered ($\mathcal{O}(o_2, f_1) = not\_watered$). Similarly, $\mathcal{A}_4$ attacks $\mathcal{A}_3$ without an attack back according to $p_{v_2}^{f_2}$, as using no water ($\mathcal{O}(o_2, f_2) = none$) is preferable to using some ($\mathcal{O}(o_1, f_2) = some$). From a value-level perspective, $\mathcal{A}_1$ attacks $\mathcal{A}_4$ because according to p_{rv}, v_1 (functioning) is preferred to v_2 (conservation). In the default context the only acceptable argument is $\mathcal{A}_1$, and so by default o_1 is sceptically acceptable.

When it starts raining, the rain context $\mathcal{C}_{rain}$ activates as $weather == rain$ is true. This joint context leads to a different option outcome function $\mathcal{O}$ than in the default context, replacing $\mathcal{O}(o_2, f_1) = not\_watered$ with $\mathcal{O}(o_2, f_1) = watered$. In this example, the arguments are not changing as none of the elements in $\{V, F, O, I\}$ are changing.

In this joint context $\mathcal{C}_{rain}$, $\mathcal{A}_1$ does not attack $\mathcal{A}_2$. This is because $\mathcal{O}(o_2, f_1) = \mathcal{O}(o_1, f_1) = watered$ in $\mathcal{M}_{\mathcal{C}_{rain}}$, compared to $\mathcal{O}(o_2, f_1) = not\_watered$ in $\mathcal{M}_{\mathcal{C}_\emptyset}$. Perspective $p_{v_1}^{f_1}$ does not imply a preference for $\mathcal{A}_1$ over $\mathcal{A}_2$ in the rain. $\mathcal{A}_3$ is defeated by $\mathcal{A}_4$ as before.

Given that we have removed the root value perspective p_{rv} in the rain context, we no longer prioritise v_1 (functioning) over v_2 (conservation). Hence $\mathcal{A}_1$, $\mathcal{A}_2$ and $\mathcal{A}_4$ are all acceptable. This concludes with o_1 (sprinkler activating) being credulously acceptable, and o_2 (sprinkler not activating) being sceptically acceptable, since $A_2 = (A_1 \setminus \{o_1\}) \cup \{o_2\}$. Because not activating satisfies every value, it should be the action taken.

6 Related Work

Value-based argumentation has been previously studied in [4]. Here, values are modelled as being promoted or demoted by different arguments, and acceptable arguments are determined by an audience-specific value preference. Atkinson and Bench-Capon [3] framed value-based reasoning in a hierarchical manner, where decisions were made in pursuit of goals to realise values. CAVA presents an alternate framing of this problem, where options are taken to support increasingly abstract values with no mention of goals. The link between these two frameworks warrants further investigation. CAVA arguments also provide explainability for options via the values supported, which is not emphasised in Atkinson and Bench-Capon's work.

Hierarchical value-based reasoning was also modelled in Osman and d'Inverno [21]. Like CAVA, this framework goes from abstract values to

grounded features (labelled properties in their work). The work also includes a means of modelling context change. However, Osman and d'Inverno's approach to context modelling is to change value and property importance, but CAVA goes further and allows changing the values and features considered. CAVA complements the model in Osman and d'Inverno [21] by developing a comparable model of values for argumentation.

7 Conclusions & Future Work

We have introduced Contextual Argumentation for Value-based Assessment, or CAVA, as a new argumentation framework for modelling contextual value-based reasoning using a hierarchy of values.

We can extend CAVA from modelling a single agent's values for use in multi-agent systems. As an argumentation framework CAVA enables agents to express values to each other, and the justifications behind choices. With further development, negotiation between values and aggregating values could be explored. Combining this with other value-related mechanisms such as norms for aggregating values offers interesting potential in modelling multi-agent value alignment. A software implementation of CAVA as an agent reasoning mechanism would support further research of CAVA applications in multi-agent systems.

An area for further investigation is how CAVA defines argument acceptability. Currently our definition prefers arguments that maximally support given values from the available options. In practice, value-based decision-making does not always need to maximise values, and can work within satisficing objectives. Finally, we are interested in the applications of CAVA to the value alignment problem. Identifying and communicating values is an important dimension of value alignment [19], to which CAVA can contribute as a framework for modelling values.

Acknowledgments. This work is supported by UK Research and Innovation (Grant No.: EP/S023437/1).

Disclosure of Interests. The authors have no competing interests concerning the work in this paper.

Bibliography

[1] Abel, D., MacGlashan, J., Littman, M.L.: Reinforcement learning as a framework for ethical decision making. In: AAAI workshop: AI, ethics, and society, vol. 16, Phoenix, AZ (2016)

[2] Aler Tubella, A., Theodorou, A., Dignum, V., Dignum, F.: Governance by glass-box: Implementing transparent moral bounds for ai behaviour. arXiv preprint arXiv:1905.04994 (2019)

[3] Atkinson, K., Bench-Capon, T.: Legal case-based reasoning as practical reasoning. Artificial Intelligence and Law **13**(1), 93–131 (2005)

[4] Atkinson, K., Bench-Capon, T.J.: Value-based argumentation. FLAP **8**(6), 1543–1588 (2021)

[5] Bistarelli, S., Taticchi, C.: Introducing a tool for concurrent argumentation. In: European Conference on Logics in Artificial Intelligence, pp. 18–24, Springer (2021)

[6] Boot, M.: Problems of incommensurability. Social Theory and Practice pp. 313–342 (2017)

[7] Broome, J.: Incommensurable values (2000)

[8] Caminada, M.: Rationality postulates: Applying argumentation theory for non-monotonic reasoning. Journal of Applied Logics **4**(8), 2707–2734 (2017)

[9] Christiano, P.F., Leike, J., Brown, T., Martic, M., Legg, S., Amodei, D.: Deep reinforcement learning from human preferences. Advances in neural information processing systems **30** (2017)

[10] Dung, P.M.: On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. Artificial intelligence **77**(2), 321–357 (1995)

[11] Hadfield-Menell, D., Russell, S.J., Abbeel, P., Dragan, A.: Cooperative inverse reinforcement learning. Advances in neural information processing systems **29** (2016)

[12] Hunsberger, L.: Whatever you say. In: European Workshop on Logics in Artificial Intelligence, pp. 229–241, Springer (2006)

[13] Hunter, A.: Non-monotonic reasoning in deductive argumentation. arXiv preprint arXiv:1809.00858 (2018)

[14] Kakas, A., Moraitis, P.: Argumentation based decision making for autonomous agents. In: Proceedings of the second international joint conference on Autonomous agents and multiagent systems, pp. 883–890 (2003)

[15] Kakas, A.C., Moraitis, P., Spanoudakis, N.I.: Gorgias: Applying argumentation. Argument & Computation **10**(1), 55–81 (2018)
[16] Liscio, E., Lera-Leri, R., Bistaffa, F., Dobbe, R.I., Jonker, C.M., Lopez-Sanchez, M., Rodriguez-Aguilar, J.A., Murukannaiah, P.K.: Value inference in sociotechnical systems. In: Proceedings of the 2023 International Conference on Autonomous Agents and Multiagent Systems, pp. 1774–1780 (2023)
[17] Liscio, E., Van Der Meer, M., Siebert, L.C., Jonker, C.M., Murukannaiah, P.K.: What values should an agent align with? an empirical comparison of general and context-specific values. Autonomous Agents and Multi-Agent Systems **36**(1), 23 (2022)
[18] Maio, G.R.: Mental representations of social values. In: Advances in experimental social psychology, vol. 42, pp. 1–43, Elsevier (2010)
[19] McKinlay, J., De Vos, M., Hoffmann, J.A., Theodorou, A.: Understanding the process of human-ai value alignment. arXiv preprint arXiv:2509.13854 (2025)
[20] Modgil, S., Toni, F., Bex, F., Bratko, I., Chesnevar, C.I., Dvořák, W., Falappa, M.A., Fan, X., Gaggl, S.A., García, A.J., et al.: The added value of argumentation. Agreement technologies pp. 357–403 (2013)
[21] Osman, N., d'Inverno, M.: A computational framework of human values (2024)
[22] Van de Poel, I.: Translating values into design requirements. Philosophy and engineering: Reflections on practice, principles and process pp. 253–266 (2013)
[23] Raz, J.: Value incommensurability: some preliminaries. In: Proceedings of the Aristotelian Society, vol. 86, pp. 117–134, JSTOR (1985)
[24] Rokeach, M.: The nature of human values. Free peess (1973)
[25] Sanneman, L., Shah, J.: Transparent value alignment. In: Companion of the 2023 ACM/IEEE International Conference on Human-Robot Interaction, pp. 557–560 (2023)
[26] Schwartz, S.H.: Universals in the content and structure of values: Theoretical advances and empirical tests in 20 countries. In: Advances in experimental social psychology, vol. 25, pp. 1–65, Elsevier (1992)
[27] Schwartz, S.H., Bilsky, W.: Toward a universal psychological structure of human values. Journal of personality and social psychology **53**(3), 550 (1987)
[28] Sklar, E.I., Azhar, M.Q.: Explanation through argumentation. In: Proceedings of the 6th International Conference on Human-Agent Interaction, pp. 277–285 (2018)
[29] Svegliato, J., Nashed, S.B., Zilberstein, S.: Ethically compliant planning in moral autonomous systems. In: AISafety@ IJCAI (2020)

[30] Vassiliades, A., Bassiliades, N., Patkos, T.: Argumentation and explainable artificial intelligence: a survey. The Knowledge Engineering Review **36**, e5 (2021)

[31] van der Weide, T.L., Dignum, F., Meyer, J.J.C., Prakken, H., Vreeswijk, G.: Arguing about preferences and decisions. In: Argumentation in Multi-Agent Systems: 7th International Workshop, ArgMAS 2010 Toronto, ON, Canada, May 10, 2010 Revised, Selected and Invited Papers 7, pp. 68–85, Springer (2011)

[32] van der Weide, T.L., Dignum, F., Meyer, J.J.C., Prakken, H., Vreeswijk, G.A.: Practical reasoning using values: Giving meaning to values. In: Argumentation in Multi-Agent Systems: 6th International Workshop, ArgMAS 2009, Budapest, Hungary, May 12, 2009. Revised Selected and Invited Papers 6, pp. 79–93, Springer (2010)