

Supplementary material for the article:

Trends and Challenges in Machine Ethics: An Updated Survey of Artificial Moral Agency Implementations

Journal:	AI and Ethics
Corresponding author:	Michael Krah
Co-author:	Martin Tröschel
Corr. author e-mail:	michael.krah@uol.de
Corr. author affiliation:	Carl von Ossietzky Universität Oldenburg Department of Computing Science Ammerländer Heerstraße 114-118 Oldenburg, 26129, Germany

1 Cardoso *et al.* (2022)

Key Points 1

- Written in the Jason agent programming language (p. 26)
- Agents in the ethical governor system: (1) execution agent, (2) set of evidential reasoners, (3) arbiter (p. 26)
- Ethical governor can be part of a larger multi-agent system (p. 26)
- Execution agents asks evidential reasoners to suggest actions (actions that require ethical reasoning identified and defined by developer) (p. 26)
- Each evidential reasoner will choose an action based on its instantiation, using epistemic reasoning and its set of beliefs, plans, and Prolog-like rules (p. 26)
- Arbiter collects all suggestions, picks one of them based on an ethical reasoning strategy, and sends the selected choice to the execution agent (p. 26)
- Abstraction and instantiation level (pp. 26-7)
 - Abstraction level: communication protocol among agents
 - Instantiation level: instantiated agents include their parent abstractions (e.g., execution agent implements the suggested actions)

Table 1: First dimension

	Ethical theory	Categorization
Ethical theory type	Deontological ethics	
	Consequentialism	
	Virtue ethics	
	Particularism	
	Hybrid	Hierarchically specific
		Hierarchically nonspecific
	Configurable ethics	✓(C + D)
Conflict sensitivity	Ambiguous	
	B/w AMAs	✓
	Within AMA	
	None	
Source	Developer	✓
	User	
	Expert	
	Lay people	
	(Legal) code/guidelines	
	Stakeholders	
	None	
Moral patient sensitivity	Yes	
	No	✓
	Unclear	

Table 2: Second dimension

Non-technical aspect		Categorization
Approach	Top-down	✓
	Bottom-up	
	Hybrid	
Contribution type	Model representation	
	Model selection	
	Judgment provision	✓
	Action selection/execution	✓
Evaluation	Test	Non-expert
		Expert
	Prove	Laws
		Model checker
		Logical proof
	Informal	Example scenario ✓
		Face validity
	None	
Domain-specific	yes/no	✗

Table 3: Third dimension

Technical aspect		Categorization
Tech type	Logical reasoning	Deductive logic ✓
		Non-monotonic logic
		Abductive logic
		Deontic logic
		Rule-based system
		Event calculus
		Knowledge representation and ontologies
		Inductive logic
	Probabilistic reasoning	Bayesian approach
		Markov models
		Statistical inference
	Learning	Inductive logic
		Decision tree
		Reinforcement learning
		Neural networks
		Evolutionary computing
	Optimization	
	Case-based reasoning	
Input	Case	Logical representation ✓
		Numerical representation
		(Structured) language representation
Implementation availability	Sensor data	
	Specification details ✓	
	Implementation details ✓	
	Code (link) provided https://github.com/autonomy-and-verification/ethicalgov	
Other	Hardware ✗	
	Feedback ✗	
	Explanation ✗	
	User interface ✗	
	Automated processing ✗	

2 Dennis *et al.* (2021)

Key Points 2

- The HERA system (Lindner, Bentzen, and Nebel 2017) is used as an embedded ethical reasoner (p. 11471)
- An ethical reasoner uses an ethical encoding to recommend or allow some action or set of actions given some situation (p. 11472)
- Utilitarian reasoner calculates the sum of the utilities of all the consequences of an interpretation and compares it to all the other interpretations (p. 11471)
- For the Kantian reasoner, an action is only permissible as long as no moral patient is used as a means to an end (p. 11471-2)
- As part of the agent's reasoning cycle, an external environment is polled for new perceptions which become beliefs which are then used to select actions (p. 11471)
- The agent's environment may be purely software or it may include actions and perceptions from the real world (p. 11471)
- A context is identified, through perception and logical reasoning, as a logical formula ϕ that holds true in that context, and the truth of ϕ can be used to control the ethical encoding used by the system (p. 11471)
- A context specification describes how a context is detected and how an ethical encoding can be changed (which update function on ethical encodings is used depends on the context detected) (p. 11472)
- A context sensitive ethical reasoner polls an external environment to update its belief base, examines context specifications, and forms a set of new candidate ethical encodings based on the update function on ethical encodings (p. 11472)
- Each time an ethical reasoner is required, a HERA model can be constructed from a JUNO model, which is a key element of the operational semantics described by Dennis *et al.* (2021, p. 11473)
- JUNO's reasoning cycle (pp. 11474-5)
 1. JUNO agent updates beliefs with new percepts while removing any perception beliefs that are no longer observed
 2. Additional beliefs may be added to the belief based by means of logical inference from the belief base
 3. JUNO determines which context(s) it is currently in and updates the model by replacing the current model with the default model, then repeatedly applying all applicable update functions
 4. A HERA model is constructed from the JUNO model, and passed to the HERA reasoner, which produces a set of ethically acceptable actions, from which JUNO selects an action
 5. JUNO executes the selected action
 6. Additional reasoning may be applied to remove beliefs

Table 4: First dimension

	Ethical theory	Categorization
Ethical theory type	Deontological ethics	
	Consequentialism	
	Virtue ethics	
	Particularism	
	Hybrid	Hierarchically specific
		Hierarchically nonspecific
	Configurable ethics	✓(D + C + Hyb: D > C)
Conflict sensitivity	Ambiguous	
	B/w AMAs	
	Within AMA	
	None	✓
Source	Developer	
	User	✓
	Expert	
	Lay people	
	(Legal) code/guidelines	
	Stakeholders	
	None	
Moral patient sensitivity	Yes	✓
	No	
	Unclear	

Table 5: Second dimension

Non-technical aspect		Categorization
Approach	Top-down	✓
	Bottom-up	
	Hybrid	
Contribution type	Model representation	
	Model selection	
	Judgment provision	✓
	Action selection/execution	✓
Evaluation	Test	Non-expert
		Expert
	Prove	Laws
		Model checker
		Logical proof
	Informal	Example scenario ✓
		Face validity
	None	
Domain-specific	yes/no	✗

Table 6: Third dimension

Technical aspect		Categorization	
Tech type	Logical reasoning	Deductive logic	✓
		Non-monotonic logic	
		Abductive logic	
		Deontic logic	
		Rule-based system	
		Event calculus	
		Knowledge representation and ontologies	
		Inductive logic	
	Probabilistic reasoning	Bayesian approach	
		Markov models	
		Statistical inference	
	Learning	Inductive logic	
		Decision tree	
		Reinforcement learning	
Neural networks			
Evolutionary computing			
Optimization			
Case-based reasoning			
Input	Case	Logical representation	✓
		Numerical representation	
		(Structured) language representation	
	Sensor data		
Implementation availability	Specification details		
	Implementation details		
	Code (link) provided	https://autonomy-and-verification.github.io/tools/mcapl	
Other	Hardware	✗	
	Feedback	✗	
	Explanation	✗	
	User interface	✗	
	Automated processing	✗	

3 Chaput *et al.* (2021)

Key Points 3

- Paper presents a hybrid approach to learn behaviors under the guidance of judgments originating from symbolic reasoning, using judging and learning agents (p. 14)
- Symbolic judging agents (top-down) and neural learning agents (bottom-up) (p. 15)
- Abstract architecture (pp. 15-6)
 - Multi-agent system comprising humans, learning agents, and judging agents
 - Each judging agent is associated with a single moral value and has a set of rules to determine if a given action supports or defeats this moral value
 - Rules can target consequences of actions (consequentialism) or actions themselves (deontological ethics)
 - Using these rules, judging agents compute a judgment that serves as feedback for learning agents
 - Using the feedback, learning agents select and perform the action which maximizes their reward over time steps
 - Human designers may rectify the learners' decisions by adjusting the rules in the judging agents that generate feedback for the learning agents
- Formal model (p. 16)
 - Set J of judging agents: each agent $j \in J$ is associated with a single moral value and has a set of rules to determine if a given action supports or defeats this value
 - Set L of learning agents: each agent $l \in L$ learns a behavior and performs actions in the environment based on feedback sent by all judging agents
 - The learning model is conceptualized as a Stochastic (or Markovian) Game, which can be solved with Reinforcement Learning [29]
- Learning agents (pp. 16-7)
 - Are tasked with learning how to select an action in a given state so as to maximize the received reward over time steps
 - The Q-Table [32] is responsible for learning the Q-value of a state-action pair so that the agent is able to pick the best action for each given state (*figure available in original paper*)
 - Two Dynamic Self-Organization Maps (DSOM), which are 2D lattices of interconnected neurons each of which is associated with a vector: (1) State-DSOM, (2) Action-DSOM
 - Each neuron corresponds to a discrete state or action
 - Decision process: (1) agent receives a state s (a vector); (2) s is compared to State-DSOM U , and the agent assumes that the closest vector prototype u is the discrete state identifier of s ; (3) since u corresponds to a neuron in the State-DSOM that is mapped to a row in the Q-Table, it is used to perform a lookup on the Q-Table and pick an action, yielding a discrete action identifier w (actions with the best interest are given a higher probability); (4) agent selects the neuron w from the Action-DSOM W and uses its associated vector prototype as the proposed action; (5) proposed action is noised to obtain the perturbed action a ; (5) perturbed action a is performed
 - Learning process: (1) agent receives a new state s' and a reward r (a scalar value); (2) if r is better than the Q-value, the action was a good one, and the Action-DSOM prototypes are updated toward the perturbed action w' ; (3) the memorized interests (Q-values) of state-action pairs are updated; (4) the State-DSOM is updated, with prototypes being moved toward the previous state s
- Judging agents (pp. 17-18)
 - Judging agents generate beliefs from their perceptions of the environment and from the

- perceived actions undertaken by learning agents
- Learning agents handle actions as vectors of real values, whereas judging agents treat each of the vector’s components as a separate action
- A fixed set of values and associated moral rules are used to generate a moral evaluation, which uses the generated beliefs to produce a moral valuation of whether a perceived action is moral, immoral, or neutral with regard to a value
- The feedback F_l received by a learning agent l is the set of judgments computed by all judging agents $j \in J$ with regard to each action component
- Judgment for learning (p. 18)
 - Learning agents expect a numeric reward, whereas judging agents manipulate symbols; learning agents use vectors of numbers as input, whereas judging agents reason over beliefs
 - Symbolic valuations are transformed into a numeric reward as follows: (1) the judgment of a single judging agent, i.e., a list of valuations, is transformed into a single number; (2) the reward r_l is the average of the rewards from each judging agent
- Experiments (pp. 18-20)
 - Simulation: (a) neighborhood of households, offices, and schools; (b) each building is represented by a learning agent in the Smart Grid; (c) the learning agent has to learn how to consume and exchange energy in order to satisfy the comfort needs of the occupants, while considering the interests of other participants in the grid
 - Four moral values and associated rules for all agents: (1) security of supply; (2) affordability; (3) inclusiveness; (4) environmental sustainability
 - Simulation is an ethical dilemma, since no action can satisfy all moral rules, meaning that every action implies a transgression of at least one moral value
 - Learning agents represent prosumers participating in the micro-grid, with each agent being associated with a specific consumption profile (Household, Office, School)
 - At each step, every learning agent performs an action, which is represented as a vector of parameters (quantity of energy consumed from the micro-grid, quantity consumed from its personal storage, quantity given to the micro-grid, quantity of energy taken from the micro-grid for its personal storage, quantity bought from the national grid for its personal storage, quantity sold to the national grid)
 - Every learning agent’s action is based on its perception of the environment (e.g., available energy, energy waste, over-consumption, personal storage, payoff)
 - Judging agents are implemented in Jason [10] and use the JaCaMo platform [8], which is implemented in Java; REST messages are used to solve the communication problem between the learning agents, which are implemented in Python
 - One judging agent is implemented for each moral value, and each judging agent judges each component of the learning agents’ actions, while having access to the perceptions of each agent
 - Judging agents contain rules in a pseudo-Prolog language to determine if an action supports or defeats their associated value
- Results (pp. 20-1)
 - Multiple sets of simulations based on the following parameters: (a) number of learning agents (small vs. medium); (b) consumption profile (annually vs. daily); (c) judges’ configuration (mono-values with one judging agent vs. incremental, where judging agents are activated one by one, vs. decremental vs. default, where all agents are activated the whole time)
 - No difference between small and medium experiments
 - Focus on default and incremental scenarios, and the annually consumption profile
 - Reward increased over time steps, but a drop in rewards at the end of the simulation,

- mostly due to the inclusiveness reward, which seems harder to learn
- Adding judges one by one (incremental) seems to mitigate the negative impact on inclusiveness when compared to the default scenario

Table 7: First dimension

	Ethical theory	Categorization
Ethical theory type	Deontological ethics	
	Consequentialism	
	Virtue ethics	
	Particularism	
	Hybrid	Hierarchically specific
		Hierarchically nonspecific
	Configurable ethics	✓(D + C + Hyb: HNS)
Conflict sensitivity	Ambiguous	
	B/w AMAs	✓
	Within AMA	
	None	
Source	Developer	✓
	User	
	Expert	
	Lay people	
	(Legal) code/guidelines	
	Stakeholders	
	None	
Moral patient sensitivity	Yes	
	No	✓
	Unclear	

Table 8: Second dimension

Non-technical aspect		Categorization
Approach	Top-down	
	Bottom-up	
	Hybrid	✓
Contribution type	Model representation	
	Model selection	
	Judgment provision	✓
	Action selection/execution	✓
Evaluation	Test	Non-expert
		Expert
	Prove	Laws
		Model checker
		Logical proof
		Example scenario ✓
	Informal	Face validity
		None
Domain-specific	yes/no	✗

Table 9: Third dimension

Technical aspect		Categorization
Tech type	Logical reasoning	Deductive logic
		Non-monotonic logic
		Abductive logic
		Deontic logic
		Rule-based system
		Event calculus
		Knowledge representation and ontologies
		Inductive logic
	Probabilistic reasoning	Bayesian approach
		Markov models
		Statistical inference
	Learning	Inductive logic
		Decision tree
		Reinforcement learning
		Neural networks
		Evolutionary computing
	Optimization	
	Case-based reasoning	
Input	Case	Logical representation
		Numerical representation
		(Structured) language representation
Implementation availability	Sensor data	
	Specification details	
	Implementation details	
	Code (link) provided	
Other	Hardware	
	Feedback	
	Explanation	
	User interface	
	Automated processing	

4 Pagnucco *et al.* (2021)

Key Points 4

- Paper develops a formal approach to ethical reasoning about action founded on an account of the epistemic situation calculus with knowledge (815)
- Background (815-7)
 - Ethical frameworks
 - Situation calculus
- Consequentialist ethics proposal 1 (817-9)
 - Only the nature of the final outcome achieving a specified goal is considered, regardless of the sequence of actions achieving the goal
 - Agent's ethical preferences are captured by a function *goodness* that indicates the moral desirability of a state
 - The known goodness of a situation s is obtained by aggregating the goodness of all alternative situations s' at s
 - Situation s is a preferred known ethical outcome iff s satisfies the set goal ϕ , is executable, and all other situations s' satisfying ϕ and being executable imply them being less desirable, meaning that the known goodness of all s' is lesser or equal to that of s
- Consequentialist ethics proposal 2 (819)
 - All sequences of actions leading to a goal are considered by function *goodness**, which uses a domain specific aggregation function *agg** that aggregates the goodness of all states up to the state s that achieves the goal
 - The known consequential goodness of s is obtained by aggregating *goodness**(s') of all alternative situations s' at situation s
 - Situation s is a preferred known consequential outcome if s achieves the specified goal, is executable, and all other situations s' achieving the specified goal that are executable imply that their known consequential goodness is lesser or equal to that of s
- Deontological ethics proposal (819-20)
 - Deontological ethics is interpreted as meaning that if one is duty bound to perform an action, then one should do so
 - When an action's duty precondition axiom is true and the action is possible, one is obliged to perform it
 - A situation s is dutifully executable if all situations that are a subsequence of s , which result from performing a at s' imply that there is a duty to perform a at s' or there is no other action a' for which there is a duty to perform it
- Implementation: Encoding of ethical scenarios using an Answer Set Programming planning approach, which tries to mirror the epistemic situation calculus encoding (820)

Table 10: First dimension

	Ethical theory	Categorization
Ethical theory type	Deontological ethics	
	Consequentialism	
	Virtue ethics	
	Particularism	
	Hybrid	Hierarchically specific
		Hierarchically nonspecific
	Configurable ethics	✓(D + C)
Conflict sensitivity	Ambiguous	
	B/w AMAs	
	Within AMA	
	None	✓
Source	Developer	
	User	
	Expert	
	Lay people	
	(Legal) code/guidelines	
	Stakeholders	
	None	✓
Moral patient sensitivity	Yes	
	No	
	Unclear	✓

Table 11: Second dimension

Non-technical aspect		Categorization
Approach	Top-down	✓
	Bottom-up	
	Hybrid	
Contribution type	Model representation	✓
	Model selection	
	Judgment provision	
	Action selection/execution	
Evaluation	Test	Non-expert
		Expert
	Prove	Laws
		Model checker
		Logical proof
		Example scenario ✓
	Informal	Face validity
		None
Domain-specific	yes/no	✗

Table 12: Third dimension

Technical aspect		Categorization
Tech type	Logical reasoning	Deductive logic ✓
		Non-monotonic logic
		Abductive logic
		Deontic logic
		Rule-based system
		Event calculus
		Knowledge representation and ontologies
		Inductive logic
	Probabilistic reasoning	Bayesian approach
		Markov models
		Statistical inference
	Learning	Inductive logic
		Decision tree
		Reinforcement learning
		Neural networks
		Evolutionary computing
	Optimization	
	Case-based reasoning	
Input	Case	Logical representation ✓
		Numerical representation
		(Structured) language representation
Implementation availability	Sensor data	
	Specification details	
	Implementation details ✓	
Other	Code (link) provided	
	Hardware ✗	
	Feedback ✗	
	Explanation ✗	
	User interface ✗	
	Automated processing ✗	

5 Baumann and Heitz (2022)

Key Points 5

- Paper presents a procedure for the moral assessment and implementation of group fairness in prediction-based decision making systems (20)
- A step-by-step procedure for fair prediction-based decision making (20-1)
 - Prediction-based decision making systems taking decisions $D = \{0, 1\}$ for individuals of a population
 - Decision is based on a prediction of a single binary attribute $Y = \{0, 1\}$
 - Assumption that a prediction model exists that outputs the probability p denoting an individual's probability of $Y = 1$ and that there is a decision rule that maps this probability into a binary decision D
 - The principle of Fair Equality of Chances (FEC)
 - * Three moral categories: (1) benefit U_{DS} , (2) justifier J (the claim to U_{DS}), (3) group membership G
 - * An imperfect prediction-based decision rule satisfies FEC iff individuals equal in their values for J have the same expectation of U_{DS} , irrespective of their G values
 - Proposed procedure: (a) moral assessment of fairness, (b) choice of statistical fairness criterion, (c) implementation of fairness
- Morally assessing fairness (21)
 - Define benefit: utility from the perspective of the affected individuals U_{DS}
 - Define groups: it is assumed that there is no moral justification for receiving unequal benefits because of group membership
 - Define justice: Is there a justifier J for unequal benefits among decision subjects?
- Choose statistical fairness criterion (21-2)
 - Separation: requires that, among all those individuals equal in Y , an equal share of individuals is assigned a positive decision across groups
 - Sufficiency: requires that, among all those individuals assigned to the same decision D , an equal share of individuals belongs to the positive class $Y = 1$ across groups
 - Independence: requires an equal share of individuals to be assigned to a positive decision ($D = 1$) across groups G
 - Conditional statistical parity: requires parity of expected decisions among those subgroups of G who are equal in their values for legitimate factors L
 - The Extended Fair Equality of Chances (E-FEC) principle can be used to define further group fairness criteria: an imperfect prediction-based decision rule satisfies E-FEC iff individuals equal in their morally relevant values for $j^* \in J$ have the same expectations of U_{DS} irrespective of their G values
 - *Table of group fairness criteria available in original paper*
- Implementing fairness (22-3)
 - Post-processing solution maximizing the decision maker utility while satisfying a fairness constraint: $\arg \max_d E(U_{DM})$ subject to FC
 - The optimal decision rule d^* depends on the chosen definition of fairness: e.g., for (conditional) statistical parity and separation (and its relaxations): $d^* = \begin{cases} 1 & p \geq \tau_G \\ 0 & \text{otherwise} \end{cases}$, where $\tau_G \in [0, 1]$ denote constants that differ depending on group membership $g \in G$
 - The tradeoff between the performance and the fairness of a prediction-based decision system can be balanced by considering a partial fulfillment of the chosen FC

- Application (23-4)
 - Prediction-based decision making system making jail-or-release decisions using ProPublica data
 - Post-processing solution to derive optimal fair decision rules based on a training set (2/3) and evaluation based on the test set (1/3)
 - Define benefit: individuals who are released ($D = 0$) are better off than those who are detained ($D = 1$)
 - Define groups: G is race, where $G = c$ denotes Caucasian subjects, and $G = a$ denotes African American subjects
 - Define justice: it is morally justifiable that $J = Y$, because detainees should only be released if they do not recidivate
 - Choose group fairness criterion
 - * Separation as FC
 - * Imperfect decision rule, given that the dominant moral fairness issue is to make sure that the harm of being kept in prison ($D = 1$) if not being a danger to public safety ($Y = 0$) is distributed fairly among the groups (FPR parity)
 - * FPR parity: defendants of type $Y = 0$ deserve the same sentencing outcome D , on average, regardless of group membership G
 - Implementing fairness
 - * U_{DM} equivalent to the accuracy of the decisions
 - * Reformulation of FC to allow for partial fulfillment of the constraint, where $\gamma = 1$ amounts to fully enforcing fairness, and $\gamma < 1$ amounts to partially enforcing fairness
 - Results: decision rule with group-specific thresholds results in a decision maker utility of $U_{DM} = 0.662$ and equal $FPR_{G=a} = FPR_{G=c} = 0.3$

Table 13: First dimension

	Ethical theory	Categorization
Ethical theory type	Deontological ethics	
	Consequentialism	
	Virtue ethics	
	Particularism	
	Hybrid	Hierarchically specific
		Hierarchically nonspecific
	Configurable ethics	✓(C)
Conflict sensitivity	Ambiguous	
	B/w AMAs	
	Within AMA	
	None	✓
Source	Developer	
	User	
	Expert	
	Lay people	
	(Legal) code/guidelines	
	Stakeholders	✓
	None	
Moral patient sensitivity	Yes	✓
	No	
	Unclear	

Table 14: Second dimension

Non-technical aspect		Categorization
Approach	Top-down	✓
	Bottom-up	
	Hybrid	
Contribution type	Model representation	✓(formalization of various fairness principles)
	Model selection	
	Judgment provision	
	Action selection/execution	✓
Evaluation	Test	Non-expert
		Expert
	Prove	Laws
		Model checker
		Logical proof
	Informal	Example scenario ✓
		Face validity
	None	
Domain-specific	yes/no	✗

Table 15: Third dimension

Technical aspect		Categorization
Tech type	Logical reasoning	Deductive logic
		Non-monotonic logic
		Abductive logic
		Deontic logic
		Rule-based system
		Event calculus
		Knowledge representation and ontologies
		Inductive logic
	Probabilistic reasoning	Bayesian approach
		Markov models
		Statistical inference
	Learning	Inductive logic
		Decision tree
		Reinforcement learning
		Neural networks
		Evolutionary computing
	Optimization	✓
	Case-based reasoning	
Input	Case	Logical representation
		Numerical representation
		(Structured) language representation
	Sensor data	
Implementation availability	Specification details	
	Implementation details	
	Code (link) provided	https://github.com/joebaumann/fair-prediction-based-decision-making
Other	Hardware	✗
	Feedback	✗
	Explanation	✗
	User interface	✗
	Automated processing	✗

6 Erdoğan (2022)

Key Points 6

- Introduction (1-2)
 - Paper applies abstract moral theories (Utilitarianism, Maximin/Leximin, Egalitarianism, Prioritarianism) to shortlisting using Linear Programming (LP)
- Motivation (2)
 - Necessary conditions an ethical scoring method should satisfy: (1) If two applicants have the exact same qualifications, they should have the same score; and (2) if an applicant has qualifications that are greater than or equal to those of another applicant for each type of qualification, then the score of the former applicant must be greater than or equal to that of the latter
 - A linear function with nonnegative coefficients satisfies both conditions, which is why applicants' qualifications are multiplied by an appropriate weight
 - Qualification values are normalized to the interval $[0, 1]$
 - The recruiting department may provide lower and upper bounds on the weight assigned to each qualification type
- Models (2-4)
 - Preliminaries
 - * I is the set of applicants, J is the set of qualification types
 - * q_{ij} is the qualification level of applicant $i \in I$ for qualification $j \in J$; normalized to the range $[0, 1]$, this is given by $\hat{q}_{ij}$
 - * Lower bound for each type of qualification $0 \leq l_j \leq 1$ with condition $\sum_{j \in J} l_j \leq 1$; upper bound $l_j \leq u_j \leq 1$ with condition $\sum_{j \in J} u_j \geq 1$
 - * x_j is the decision variable that corresponds to the weight of qualification type j , y and z are the minimum and maximum scores attained by the applicants
 - Utilitarianism (U)
 - * Maximization of the sum total of happiness (Tännsjö, 2019)
 - * Maximization of the sum of all scores assigned to the applicants: maximization of $\sum_{i \in I} \sum_{j \in J} \hat{q}_{ij} x_j$, subject to constraint (2) $\sum_{j \in J} x_j = 1$, and (3) $l_j \leq x_j \leq u_j \forall j \in J$
 - * Optimal solution can be found in $O(|J| \log(|J|))$
 - * Solution algorithm (*pseudo-code available in original paper*)
 - Maximin model (M)
 - * Gives priority to the person who is worst off (Tännsjö, 2019)
 - * Maximizes the lowest score among all participants: maximization of y subject to constraints (2), (3), (5) $y \leq \sum_{j \in J} \hat{q}_{ij} x_j \forall i \in I$, and (6) $y \geq 0$
 - * No closed-form optimal solution
 - Leximin model (L)
 - * Caters to the needs of those who are next in line after the need of those who are worst off have been catered to (Tännsjö, 2019)
 - * Maximizes y first (optimal solution denoted by y^*), followed by the sum of all applicant scores: maximization of $\sum_{i \in I} \sum_{j \in J} \hat{q}_{ij} x_j$ subject to constraints (2), (3), (8) $\sum_{j \in J} \hat{q}_{ij} x_j \geq y^* \forall i \in I$
 - * Same complexity as Maximin
 - Egalitarian model
 - * Defines inequalities as negative (Tännsjö, 2019)
 - * Finds weights that minimize the differences between the maximum and the minimum scores of applicants: minimizes $z - y$ subject to constraints (2), (3), (5), (6), (10)

- $z \geq \sum_{j \in J} \hat{q}_{ij} x_j \forall i \in I$, and (11) $z \geq 0$
- * Polynomial level of complexity
- Prioritarian models
 - * Aim to prioritize the subset of disadvantaged applicants at an aggregate level, where $D \in I$ is the subset of disadvantaged applicants, y' is the minimum score among the disadvantaged applicants, z' is the maximum score among the disadvantaged applicants
 - * Prioritarian/Utilitarian (P/U): maximize $\sum_{i \in D} \sum_{j \in J} \hat{q}_{ij} x_j$ subject to constraints (2), (3)
 - * Prioritarian/Maximin (P/M): maximize y' subject to constraints (2), (3), (14) $y' \leq \sum_{j \in J} \hat{q}_{ij} x_j \forall i \in D$, and (15) $y' \geq 0$
 - * Prioritarian/Leximin (P/L): maximize $\sum_{i \in I} \sum_{j \in J} \hat{q}_{ij} x_j$ subject to constraints (2), (3), (17) $\sum_{j \in J} \hat{q}_{ij} x_j \geq y'^* \forall i \in D$
 - * Prioritarian/Egalitarian (P/E): minimize $z' - y'$ subject to constraints (2), (3), (14), (15), (19) $z' \geq \sum_{j \in J} \hat{q}_{ij} x_j \forall i \in I \setminus D$, (20) $z' \geq 0$
- Computational experiments (4-5)
 - Recommended formulas for lower bound $l_j = 1/(\sqrt{\rho} \times |J|) \forall j \in J$, and for upper bound $u_j = \sqrt{\rho}/|J| \forall j \in J$
 - Experiments use $\rho = 4$
 - All models are solved, i.e., optimal weights are found, using a Simplex algorithm
 - Without disadvantaged applicants
 - * Prioritarian models run the risk of providing too much advantage to the disadvantaged applicants when their qualification profiles match, and too little when their qualification profiles do not match
 - Larger number of applicants
 - * A larger number of disadvantaged applicants is more likely to cause a mismatch
 - * Suggested remedy: (1) establish quota for disadvantaged applicants, (2) use non-Prioritarian model separately on disadvantaged and non-disadvantaged applicants, (3) select top ranked applicants in both rankings to fill the quota
- Eliminating the applicant with the lowest score
 - Since (M) and (L) are prone to return low quality rankings in the presence of dominated applicants, it is suggested that (M) and (L) are run repeatedly, eliminating the applicant with the minimum score every time

Table 16: First dimension

	Ethical theory	Categorization
Ethical theory type	Deontological ethics	
	Consequentialism	
	Virtue ethics	
	Particularism	
	Hybrid	Hierarchically specific
		Hierarchically nonspecific
	Configurable ethics	✓(C)
Conflict sensitivity	Ambiguous	
	B/w AMAs	
	Within AMA	
	None	✓
Source	Developer	
	User	✓
	Expert	
	Lay people	
	(Legal) code/guidelines	
	Stakeholders	
	None	
Moral patient sensitivity	Yes	
	No	✓
	Unclear	

Table 17: Second dimension

Non-technical aspect		Categorization
Approach	Top-down	✓
	Bottom-up	
	Hybrid	
Contribution type	Model representation	✓
	Model selection	
	Judgment provision	
	Action selection/execution	✓
Evaluation	Test	Non-expert
		Expert
	Prove	Laws
		Model checker
		Logical proof
	Informal	Example scenario ✓
		Face validity
	None	
Domain-specific	yes/no	✓

Table 18: Third dimension

Technical aspect		Categorization
Tech type	Logical reasoning	Deductive logic
		Non-monotonic logic
		Abductive logic
		Deontic logic
		Rule-based system
		Event calculus
		Knowledge representation and ontologies
		Inductive logic
	Probabilistic reasoning	Bayesian approach
		Markov models
		Statistical inference
	Learning	Inductive logic
		Decision tree
		Reinforcement learning
		Neural networks
		Evolutionary computing
	Optimization	✓
	Case-based reasoning	
Input	Case	Logical representation
		Numerical representation
		(Structured) language representation
Implementation availability	Sensor data	
	Specification details	
	Implementation details	
	Code (link) provided	
Other	Hardware	
	Feedback	
	Explanation	
	User interface	
	Automated processing	

7 Kim *et al.* (2021)

Key Points 7

- Introduction (871-2)
 - Proposal of deontological value alignment (between AI and human values) within a hybrid approach that combines ML-based and logic-based approaches
- Two different value alignment systems (872-6)
 - Two categories of value alignment (VA): ML-based and logic based
 - ML-based approaches rely heavily on inputs, thus, risking the naturalistic fallacy: just because people behave in, for instance, racist ways (input data), it cannot follow that people ought to behave in such ways
 - Logic-based VA, on the other hand, works with a set of systematic, abstract, symbol-and-rule mechanisms
 - Psychology: System 1 thinking is associated with biased and unethical thinking, which can often be avoided when shifting to System 2 thinking
 - ML systems draw upon System 1 behavior, making them inherently vulnerable to unethical decision-making
 - The reasoning portion of moral thinking relies upon systemic operations instead of associations
- Related work (876-7)
 - Approach tries not to opt for a particular version of moral theory
 - In the hybrid approach, bottom-up learning does none of the normative work, but is used only to evaluate the truth of test propositions derived from ethical principles
- Integrating ethical principles and empirical VA (877-85)
 - Ethical principles imply certain logical propositions that must be true in order for a given action plan to be ethical (test propositions)
 - Empirical observation determines whether these propositions are true
 - Actions and reasons
 - * Ethical principles rest on the universality of reason: an agent who takes a set of reasons as justifying an action must take the reasons as justifying the same action for any agent to whom those reasons apply
 - * Three principles looked at in the paper: (1) generalization, (2) utility maximization, (3) respect for autonomy
 - Generalization principle
 - * Principle: a rational agent must believe that their reasons for acting are consistent with the assumption that all rational agents to whom the reasons apply could engage in the same action
 - * Using quantified modal logic, an action plan can be expressed as $(C_1(a) \wedge C_2(a)) \implies_a A_1(a)$, where $C_1(a)$ = Agent a would like to possess an item on display in a shop, $C_2(a)$ = a can get away with stealing the item, $A_1(a)$ = a will steal the item, $\implies_a$ indicates that a regards $C_1(a)$ and $C_2(a)$ as justifying $A_1(a)$
 - * Using quantified modal logic, an action plan is generalizable iff (generalization principle): $\diamond_a P(\forall x(C(x) \implies_x A(x)) \wedge C(a) \wedge A(a))$, where $C(a) \implies_a A(a)$ is an action plan for agent a , $C(a)$ is a conjunction of reasons for taking action $A(a)$, $P(S)$ means that it is physically possible for proposition S to be true, and $\diamond_a S$ means that a can rationally believe S
 - * For the theft example, the test proposition then is: $\diamond_a P(\forall x(C_1(x) \wedge C_2(x) \implies_x A_1(x)) \wedge C_1(a) \wedge C_2(a) \wedge A_1(a))$

- * Since the test proposition is false, the action plan is unethical
- Maximizing utility
 - * Utilitarian principle construed in a deontological fashion: the agent is required to select actions that the agent can rationally believe will maximize utility
 - * Action plan $C(a) \implies_a A(s)$ satisfies the utilitarian principle only if agent a can rationally believe that action $A(a)$ creates at least as much utility as any ethical action available under the same circumstances: $\diamond_a \forall A'(E(C(a), A'(a)) \rightarrow u(C(a), A(a)) \geq u(C(a), A'(a)))$, where $u(C(a), A(a))$ is a utility function that measures the total net expected utility of $A(a)$ under conditions $C(a)$, A' ranges over action, and where $E(C(a), A'(a))$ means that $A'(a)$ is available for a under conditions $C(a)$ and that the action plan $C(a) \implies_a A'(a)$ is generalizable and respects autonomy
 - * A particular action plan is ethical only if the test proposition is true, and empirical VA tells us whether it is true
- Respect for autonomy
 - * An agent should not adopt an action plan that the agent is rationally constrained to believe is inconsistent with an ethical action plan of another agent, without informed consent
 - * Agent a 's action plan $C(a) \implies_a A(a)$ is consistent with b 's action plan $C'(b) \implies_b A'(b)$ when $\diamond_a P(A(a) \wedge A'(b)) \vee \neg \Box_a P(C(a) \wedge C'(b))$, where $P(S)$ means that it is physically possible for proposition S to be true, $\diamond_a S$ means that a can rationally believe S , and $\diamond_a S \equiv \neg \Box_a \neg S$, where $\Box_a \neg S$ means that rationality requires a to deny S
 - * Ethical reasoning tells us that the test proposition must be true if autonomy is to be respected, whereas empirical reasoning tells us whether it is true
- Implementation issues
 - * Main implication of the proposal is that the portion of an AI system that makes ethically relevant decisions must be rule-based, because it must consist of action plans
 - * Machine learning and other forms of empirical VA can then be used to evaluate the truth of the test propositions

Table 19: First dimension

	Ethical theory	Categorization
Ethical theory type	Deontological ethics	
	Consequentialism	
	Virtue ethics	
	Particularism	
	Hybrid	Hierarchically specific
		Hierarchically nonspecific
	Configurable ethics	✓(D + C)
Conflict sensitivity	Ambiguous	
	B/w AMAs	
	Within AMA	
	None	✓
Source	Developer	
	User	
	Expert	
	Lay people	
	(Legal) code/guidelines	
	Stakeholders	
	None	✓
Moral patient sensitivity	Yes	
	No	
	Unclear	✓

Table 20: Second dimension

Non-technical aspect		Categorization
Approach	Top-down	✓
	Bottom-up	
	Hybrid	
Contribution type	Model representation	✓
	Model selection	
	Judgment provision	
	Action selection/execution	
Evaluation	Test	Non-expert
		Expert
	Prove	Laws
		Model checker
		Logical proof
	Informal	Example scenario
		Face validity
	None	✓
Domain-specific	yes/no	✗

Table 21: Third dimension

Technical aspect		Categorization
Tech type	Logical reasoning	Deductive logic ✓
		Non-monotonic logic
		Abductive logic
		Deontic logic
		Rule-based system
		Event calculus
		Knowledge representation and ontologies
		Inductive logic
	Probabilistic reasoning	Bayesian approach
		Markov models
		Statistical inference
	Learning	Inductive logic
		Decision tree
		Reinforcement learning
		Neural networks
		Evolutionary computing
	Optimization	
	Case-based reasoning	
Input	Case	Logical representation ✓
		Numerical representation
		(Structured) language representation
Implementation availability	Sensor data	
	Specification details ✓	
	Implementation details	
Other	Code (link) provided	
	Hardware ✗	
	Feedback ✗	
	Explanation ✗	
	User interface ✗	
	Automated processing ✗	

Key Points 8

- Introduction (1038-9)
 - Ultimate goal to build methods that allow for learning and trading off different conflicting values
 - Jointly optimizing for different rewards can be cast into multi-objective RL (MORL)
 - To avoid manually specifying multiple reward functions, inverse reinforcement learning (IRL) and preference-based RL are employed
 - Paper proposes Multi-Objective Reinforced Active Learning (MORAL), which combines active preference learning and inverse reinforcement learning (IRL) to interactively learn a policy of social norms from expert demonstrations
- Preliminaries (1039)
 - Multi-objective RL
 - * Multi-objective Markov decision process (MOMDP) for framing the problem of aligning an agent with multiple experts
 - * MOMDP given by $\langle \mathcal{S}, \mathcal{A}, p, \mathbf{r}, \mu_0, \gamma \rangle$, with state space $\mathcal{S}$, set of actions $\mathcal{A}$, transition distribution $p(s'|s, a)$, vector-valued reward function $\mathbf{r}(s, a) \in \mathbb{R}^m$, and discount factor $\gamma \in [0, 1)$
 - * Optimal solutions are given by a Pareto frontier $\mathcal{F} = \{\pi \mid \nexists \pi' \neq \pi : J_{\mathbf{r}}(\pi') \geq J_{\mathbf{r}}(\pi)\}$, where $J_{\mathbf{r}}(\pi) = \mathbb{E}_{\pi}[\sum_{t=0}^T \gamma^t \mathbf{r}(s_t, a_t)]$ is the vector-valued return of a policy $\pi : \mathcal{S} \rightarrow \Delta_{\mathcal{A}}$ that maps states to a distribution over actions
 - * Convex convergence set (CCS) $\mathcal{F}^* = \{\pi \in \mathcal{F} \mid \exists \mathbf{w} \in \mathbb{R}^m : \mathbf{w}^T J_{\mathbf{r}}(\pi) \geq \mathbf{w}^T J_{\mathbf{r}}(\pi'), \forall \pi' \in \mathcal{F}\}$ is the subset of Pareto-optimal solutions that can be obtained through optimizing for linear combinations of the different reward components
 - Proximal policy optimization (PPO)
 - * Given a weight $\mathbf{w} \in \mathbb{R}^m$, one can optimize for policies on the CCS using PPO on the scalarized rewards $r(s, a) = \mathbf{w}^T \mathbf{r}(s, a)$
 - * PPO maximizes the return of a parametrized policy π_{ϕ} by performing gradient descent on the clipped objective $\mathcal{L}^{CLIP}(\phi) = \mathbb{E}_t[\min(r_t(\phi)\hat{A}_t, \text{clip}(r_t(\phi), 1 - \epsilon, 1 + \epsilon)\hat{A}_t)]$, where $\mathbb{E}_t$ is the expectation at time t , r_t is a ratio of the new versus current policy, $\hat{A}_t$ is an estimated advantage at time t and $\text{clip}(x, a, b)$ limits the value of x to the interval $[a, b]$
 - Adversarial IRL (AIRL)
 - * The maximum entropy IRL goal is to derive a reward function r_{θ} from a demonstration dataset $\mathcal{D} = \{\tau_i\}_{i=1}^N$ of expert trajectories $\tau = \{s_t, a_t\}_{t=0}^T$ by solving a maximum likelihood problem $\max_{\phi} \mathbb{E}_{\tau \sim \mathcal{D}}[\log p_{\theta}(\tau)]$, where

$$p_{\theta}(\tau) \propto \mu_0(s_0) \prod_{t=0}^T p(s_{t+1}|s_t, a_t) \exp(r_{\theta}(s_t, a_t)) = \bar{p}_{\theta}(\tau)$$
 - * AIRL jointly trains a policy (generator) π_{ϕ} alongside a discriminator of the form $D_{\theta}(s, a) = \frac{\exp(f_{\theta}(s, a))}{\exp(f_{\theta}(s, a)) + \pi_{\phi}(a|s)}$
 - * D_{θ} is trained using a binary cross-entropy loss to distinguish trajectories in $\mathcal{D}$ from π_{ϕ}
 - * The agent maximizes its returns using the reward $r(s, a) = \log(D_{\theta}(s, a)) - \log(1 - D_{\theta}(s, a))$
- Multi-objective reinforced active learning (1039-40)
 - Two step procedure which separates reward and policy training to learn from multiple

experts (*figure available in original paper*)

- * First step: use a set of trajectories $\mathcal{D}_E = \cup_{i=1}^k \mathcal{D}_{E_i}$ demonstrated by k distinct experts and run AIRL to obtain a vector of reward functions $\mathbf{r} = (f_{\theta_1}, \dots, f_{\theta_k})$ and imitation policies $(\pi_{E_1}^*, \dots, \pi_{E_k}^*)$ for each subset of trajectories $\mathcal{D}_{E_i}$
- * Second step: run an interactive MORL algorithm for learning a distribution over weights that determine a linear combination of the different components in $\mathbf{r}$
 - Take a prior $p(\mathbf{w})$ over scalarization weights and initialize a reward function $r(s, a) = \mathbb{E}_{\mathbf{w}}[\mathbf{w}^T \mathbf{r}(s, a)]$
 - Repeatedly (i) optimize for the scalar reward r by running PPO for a given number of steps, (ii) query an expert for a pairwise comparison q_n of two trajectories, and (iii) update the posterior $p(\mathbf{w}|q_1, \dots, q_n)$
 - Reinitialize reward function to the posterior mean to be used by PPO in the next iteration
- Query and update: probabilistic model over expert preferences
 - * Bradley-Terry model

$$p(\tau_i \succ \tau_j | \mathbf{w}) = \frac{\exp(\mathbf{w}^T \mathbf{r}(\tau_i))}{\exp(\mathbf{w}^T \mathbf{r}(\tau_j)) + \exp(\mathbf{w}^T \mathbf{r}(\tau_i))}$$

with $\mathbf{r}(\tau) = \sum_{(s,a,s') \in \tau} \mathbf{r}(s, a, s')$ being the reward obtained from a trajectory τ and $\tau_i \succ \tau_j$ denoting the preference of τ_i over τ_j

- * Assuming that a number of pairwise comparisons $\{q_1, \dots, q_n\}$ have been obtained, the posterior is updated in a Bayesian manner: $p(\mathbf{w}|q_1, \dots, q_n) \propto p(\mathbf{w}) \prod_{t=1}^n p(q_t | \mathbf{w})$
- * Bayesian model allows to maintain a posterior that determines which reward components should be prioritized at each iteration, and an expert is then able to fine-tune the agent to any specific behavior that can result from the linear combination of reward components by providing pairwise preferences
- * Active learning procedure forms queries based on the amount of removed posterior volume: $\max_{\tau_i, \tau_j} \min(\mathbb{E}_{\mathbf{w}}[1 - p(\tau_i \succ \tau_j | \mathbf{w})], \mathbb{E}_{\mathbf{w}}[1 - p(\tau_j \succ \tau_i | \mathbf{w})])$, where the expectation over $\mathbf{w}$ is approximated using Markov Chain Monte Carlo
 - Since maximizing this expression over all pairs of feasible trajectories is computationally intractable (for sufficiently complex MOMDPs), a discrete search over randomly sampled pairs of trajectories that arise during on-policy RL experience is performed (*pseudo-code available in original paper*)
 - Since forming queries based on on-policy experience can only lead to locally optimal solutions, assuming fixed weights $\mathbf{w}$, MORAL optimizes for an entropy regularized objective $\pi^* = \arg \max_{\pi} \mathbb{E}_{\pi} [\sum_{t=0}^T \mathbf{w}^T \mathbf{r}(s_t, a_t) - \log \pi(s_t, a_t)]$
 - MORAL can be interpreted as finding an average of Kullback-Leibler (KL) divergences between the policy distribution over trajectories

$$\pi(\tau) = \mu_0(s_0) \prod_{t=0}^{T-1} p(s_{t+1} | s_t, a_t) \pi(a_t | s_t)$$

- Given $\mathbf{w} \in \mathbb{R}^k$ with $\mathbf{w} \geq 0$, $\sum w_i = 1$, $\pi^* = \arg \min_{\pi} \sum_{i=1}^k w_i D_{KL}(\pi(\tau) || p_{\theta_i}(\tau))$
- Reward normalization
 - * Although $\mathbf{w}$ is bounded by the prior, $\mathbf{w}^T \mathbf{r}(\tau)$ can become arbitrarily large, which introduces a risk of removing significant parts of the posterior support based on a single query

- * Therefore, the policies obtained from AIRL are used to normalize each reward component by setting $\overline{f_{\theta_i}}(s, a) = \frac{\overline{f_{\theta_i}}(s, a)}{|J(\pi_{E_i}^*)|}$, where $J(\pi_{E_i}^*) = \mathbb{E}_{\pi_{E_i}^*} [\sum_{t=0}^T \gamma^t f_{\theta_i}(s, a)]$ is the scalar return of $\pi_{E_i}^*$
- Experiments (1040-44)
 - Setup
 - * Gridworld environments, formed by a binary array $I \in \{0, 1\}^{C \times W \times H}$ of width W , height H , and channels that encode grid occupancy for all C object types
 - * Environments with a primary reward function r_p , which encodes a generic task that can easily be solved through deep RL, and which is added as an additional reward component to the AIRL reward functions for the active learning step
 - * Convolutional neural network architecture for PPO, consisting of two base convolutional layers with kernel size 2, and 64, as well as 256 output channels respectively; activations are fed into two separate convolutional layers with kernel size 2 and 32 output channels each, followed by a linear layer for the critic and actor heads
 - Emergency
 - * Emergency gridworld
 - Agent and 6 humans positioned in a 6×6 grid
 - Humans need to be escorted before a time limit of $T = 75$
 - Bottom right corner contains a goal state, which the agent uses when standing on its cell (fire extinguisher cell)
 - At each step, the agent can either move in one of the four directions, interact with an adjacent cell, or do nothing
 - * Reward vector, MORL, and preferences
 - Primary goal r_p to give a reward of +0.1 for each time step spent in the fire extinguisher cell
 - To learn the social norm of helping people, a reward f_{θ} is found by running AIRL on 50 synthetic demonstrations coming from a PPO agent that maximizes the number of people saved
 - Forming of reward vector $\mathbf{r} = (r_p, f_{\theta})$ and running interactive MORL using 25 queries
 - Preferences: Given two trajectories (τ_i, τ_j) , $i \succ j$ if the number of people saved in τ_i exceeds that of τ_j (if the same number of people are saved, the trajectory where more time is spent in the fire extinguisher cell is preferred)
 - Queries are spread out evenly over $6 \cdot 10^6$ environment steps
 - * Results
 - MORAL approximates a Pareto-optimal point that opts for saving all people present in the world, while maximizing the agent’s performance with respect to the primary goal
 - Delivery
 - * Delivery gridworld
 - Randomly initialized 16×16 gridworld
 - Primary goal of delivering packages to 12 locations, with reward r_p of +1 when the agent interacts with a delivery cell
 - Agent can also *help* people, and it can *clean* dirty tiles
 - Random placement of vases throughout the grid, which break whenever the agent steps on their position
 - Episodes of length $T = 50$, 12 *help*, *clean* objectives, and 8 vases
 - * Preferences
 - Multi-objective problem including three norms: *help* (active behavior), *clean* (active behavior), *avoid* (passive, i.e., inaction)

- Preferences are assumed to be automatically provided by a subjective distribution $m \in \Delta_{\{1,2,3\}}$ encoding desired priorities for *deliver*, *help*, *clean*
- Given a pair of trajectories (τ_1, τ_2) , two vectors $s_i = (o_1^i, \dots, o_n^i)$ are calculated, where o_k^i denotes the obtained returns in terms of the k -th objective in trajectory i
- Normalizing the observed returns into a discrete distribution $\bar{s}_i = \frac{s_i}{\|s_i\|_1}$, preferences can be provided according to a KL divergence metric $i^* = \arg \min_{i \in \{1,2\}} D_{KL}(\bar{s}_i \| m)$
- * Experimental setup
 - Two conflicting demonstration data sets generated by a PPO agent optimizing for (i) helping people and (ii) cleaning tiles, while both try to avoid stepping on vases
 - Use of moral as a regularizer and vector $\mathbf{r} = (r_p, f_{\theta_1}, f_{\theta_2})$, where θ_1 and θ_2 denote the trained AIRL parameters
 - Preference vector m is varied to match all possible ratios in $\{1, 2, 3\}^3$ during the active learning stage
 - 25 queries spread evenly throughout the $8 \cdot 10^6$ environment steps
- Comparison to Deep Reinforcement Learning from Human Preferences
- Ablation
- Related work (1044-5)
 - Machine Ethics
 - Inverse Reinforcement Learning
 - Learning from Expert Feedback
 - Multi-Objective-Decision-Making

Table 22: First dimension

	Ethical theory	Categorization
Ethical theory type	Deontological ethics	
	Consequentialism	✓
	Virtue ethics	
	Particularism	
	Hybrid	Hierarchically specific
		Hierarchically nonspecific
	Configurable ethics	
	Ambiguous	
Conflict sensitivity	B/w AMAs	
	Within AMA	✓
	None	
Source	Developer	
	User	
	Expert	✓
	Lay people	
	(Legal) code/guidelines	
	Stakeholders	
	None	
Moral patient sensitivity	Yes	
	No	✓
	Unclear	

Table 23: Second dimension

Non-technical aspect		Categorization
Approach	Top-down	
	Bottom-up	✓
	Hybrid	
Contribution type	Model representation	
	Model selection	
	Judgment provision	
	Action selection/execution	✓
Evaluation	Test	Non-expert
		Expert
	Prove	Laws
		Model checker
		Logical proof
	Informal	Example scenario ✓
		Face validity
	None	
Domain-specific	yes/no	✗

Table 24: Third dimension

Technical aspect		Categorization
Tech type	Logical reasoning	Deductive logic
		Non-monotonic logic
		Abductive logic
		Deontic logic
		Rule-based system
		Event calculus
		Knowledge representation and ontologies
		Inductive logic
	Probabilistic reasoning	Bayesian approach
		Markov models
		Statistical inference
	Learning	Inductive logic
		Decision tree
		Reinforcement learning ✓
		Neural networks
		Evolutionary computing
	Optimization	
	Case-based reasoning	
Input	Case	Logical representation
		Numerical representation ✓
		(Structured) language representation
Implementation availability	Sensor data	
	Specification details	
	Implementation details	
	Code (link) provided	
Other		
	Hardware	
	Feedback	
	Explanation	
	User interface	
	Automated processing	

9 Baumann *et al.* (2022)

Key Points 9

- Introduction: decision rules which maximize some utility while satisfying group fairness metrics PPV parity, FOR parity, and sufficiency (2315-6)
- Optimal decisions under fairness constraints (2317-21)
 - Problem statement and notation
 - * A decision maker has to make a binary decision D for each individual i , based on a feature vector $c_i \in \mathbb{R}^m$ which includes a protected attribute $a_i \in A$ denoting group membership
 - * $n_{A=a}$ is the number of individuals that are part of group a
 - * Attribute A is binary
 - * Decisive feature for the decision is a binary target variable Y
 - * Given that Y is unknown, probability $p_i = P[Y = 1]$ is used, which is given as a function of x_i
 - * The decision rule is a function d that maps p_i to a binary decision
 - * The goal of a rational decision maker is to maximize expected utility while also satisfying some definition of group fairness
 - * *Table of group fairness metrics available in original paper*
 - Within-group fairness
 - * A decision rule $d(p, a)$ satisfies within-group fairness with respect to protected attribute A is $\forall i \in S_{a|D=1} \forall j \in S_{a|D=0} (p_i > p_j)$, where S_a is the set of all individuals of group $a \in A$
 - Optimal decision rules with or without fairness constraints
 - * Expected individual utility:

$$u_i = \begin{cases} u_{11}p_i + u_{12}(1 - p_i) & \text{for } D = 1 \\ u_{21}p_i + u_{22}(1 - p_i) & \text{for } D = 0 \end{cases}$$

- * Expected utility gain for switching from $D = 0$:

$$\tilde{u} = \begin{cases} 0 & \text{for } D = 0 \\ \alpha p_i + \beta(1 - p_i) & \text{for } D = 1 \end{cases}$$

where $\alpha = u_{11} - u_{21}$ and $\beta = u_{12} - u_{22}$

- * Optimization for many decisions d_i over a population of individuals: $\arg \max_d \tilde{U} = \sum_{i \in S} \tilde{u}_i d_i = \sum_{i \in S} (p_i(\alpha - \beta) + \beta) d_i$
- * Optimum unconstrained decision rule d^* :

$$d_i^* = \begin{cases} 1 & \text{for } p_i > \frac{-\beta}{\alpha - \beta} \\ 0 & \text{otherwise} \end{cases}$$

- * Constrained optimization problem: $\arg \max_d \tilde{U}$ subject to some fairness constraint
- * Constrained optimization problem solved for two cases: (1) number of individuals to be selected $n_{D=1}$ predefined; (2) not predefined
- Optimal decision rules under positive predictive value (PPV) parity
 - * PPV: average probability of individuals with $D = 1$ to have $Y = 1$
 - * PPV parity: PPV should be the same across groups

- * Constrained optimization problem:

$$\begin{aligned} & \arg \max_d \tilde{U} \\ \text{subject to } & \frac{1}{n_{A=0|D=1}} \sum_{j \in S_0} p_j d_j = \frac{1}{n_{A=1|D=1}} \sum_{j \in S_1} p_j d_j = PPV, \text{ for } PPV \in [0, 1] \end{aligned}$$

- * Solution in two steps: (1) derive optimal decision rule d^* for a predefined value $PPV_t \in [0, 1]$; maximizing decision maker's utility over all possible values of PPV_t
- * For each PPV_t , there exist individuals in each group with $p = PPV_t$, and to fulfill the constraint, one of these individuals can be selected in each group, or multiple individuals can be selected such that the average of p equals PPV_t
- * For a given value of PPV_t and a predefined number of selected individuals $n_{D=1}$, any selection fulfilling the fairness constraint leads to a total utility of $\tilde{U} = (\alpha PPV_t + \beta(1 - PPV_t))n_{D=1}$
- * When the size of both groups is large but finite and $\alpha PPV_t + \beta(1 - PPV_t) > 0$, the decision maker's goal is to find the selection that satisfies the constraint $PPV = PPV_t$ with the maximum $n_{D=1}$
- * For any given PPV_t where $n_{D=1}$ is not predefined, the optimal fair decision rules d^* take the following form:

$$d_i^* = \begin{cases} 1 & \text{for } p_i \geq \tau_a \text{ for } PPV_t > BR_{A=a} \\ 0 & \text{otherwise for } PPV_t > BR_{A=a} \\ 1 & \text{for } p_i \leq \tau_a \text{ for } PPV_t < BR_{A=a} \\ 0 & \text{otherwise for } PPV_t < BR_{A=a}, \end{cases}$$

where τ_a denote different group-specific constants and $BR_{A=a}$ denotes group a 's base rate (BR) which is defined as the ratio of individuals belonging to the positive class ($Y = 1$) in a group: $BR_{A=a} = P[Y = 1|A = a] = \frac{1}{n_{A=a}} \sum_{i \in S_a} p_i$

- * From a discretization of all PPV for which a solution exists, we choose the one and the corresponding $n_{D=1}$ that maximizes the total utility
- Optimal decision rules under sufficiency
 - * Sufficiency requires PPV parity and FOR (false omission rate) parity
 - * Constrained optimization problem:

$$\begin{aligned} & \arg \max_d \sum_{i \in S} \tilde{u}_i \\ \text{subject to } & \frac{1}{n_{A=0|D=1}} \sum_{j \in S_0} p_j d_j = \frac{1}{n_{A=1|D=1}} \sum_{j \in S_1} p_j d_j \\ & \text{and to } \frac{1}{n_{A=0|D=0}} \sum_{j \in S_0} p_j (1 - d_j) = \frac{1}{n_{A=1|D=0}} \sum_{j \in S_1} p_j (1 - d_j) \end{aligned}$$

- * Since a decision rule splits a group into those selected ($D = 1$) and those not selected ($D = 0$), $\sum_{i \in S_a} p_i = (\sum_{i \in S_a} p_i (1 - d_i)) + (\sum_{i \in S_a} p_i d_i)$
- * Specifying $PPV_{A=a} = PPV_t$, this gives:

$$n_{A=a} BR_{A=a} = n_{A=a|D=0} FOR_{A=a} + n_{A=a|D=1} PPV_t$$

- * With $n_{A=a|D=0} = n_{A=a} - n_{A=a|D=1}$, we get: $FOR_{A=a} = \frac{n_{A=a} BR_{A=a} - n_{A=a|D=1} PPV_t}{n_{A=a} - n_{A=a|D=1}}$

- * The solution space consisting of all combinations of *PPV* and *FOR* that can be achieved by all groups can be constructed for one or for more groups: ...
 - * Optimal decision rules d^* that satisfy sufficiency lead to within-group unfairness in all but one of the groups if a solution exists
 - * The constrained maximization problem can be solved by choosing the optimal PPV-FOR combination
- Illustrative examples (2321-2325)
- Synthetic example
 - COMPAS

Table 25: First dimension

	Ethical theory	Categorization
Ethical theory type	Deontological ethics	
	Consequentialism	
	Virtue ethics	
	Particularism	
	Hybrid	Hierarchically specific
		Hierarchically nonspecific
	Configurable ethics	✓(C)
Conflict sensitivity	Ambiguous	
	B/w AMAs	
	Within AMA	
	None	✓
Source	Developer	
	User	
	Expert	
	Lay people	
	(Legal) code/guidelines	
	Stakeholders	
	None	✓
Moral patient sensitivity	Yes	
	No	
	Unclear	✓

Table 26: Second dimension

Non-technical aspect		Categorization
Approach	Top-down	✓
	Bottom-up	
	Hybrid	
Contribution type	Model representation	✓
	Model selection	
	Judgment provision	
	Action selection/execution	✓
Evaluation	Test	Non-expert
		Expert
	Prove	Laws
		Model checker
		Logical proof
	Informal	Example scenario ✓
		Face validity
	None	
Domain-specific	yes/no	✗

Table 27: Third dimension

Technical aspect		Categorization
Tech type	Logical reasoning	Deductive logic
		Non-monotonic logic
		Abductive logic
		Deontic logic
		Rule-based system
		Event calculus
		Knowledge representation and ontologies
		Inductive logic
	Probabilistic reasoning	Bayesian approach
		Markov models
		Statistical inference
	Learning	Inductive logic
		Decision tree
		Reinforcement learning
		Neural networks
		Evolutionary computing
	Optimization	✓
	Case-based reasoning	
Input	Case	Logical representation
		Numerical representation
		(Structured) language representation
	Sensor data	
Implementation availability	Specification details	
	Implementation details	
	Code (link) provided	https://github.com/joebaumann/fair-prediction-based-decision-making
Other	Hardware	✗
	Feedback	✗
	Explanation	✗
	User interface	✗
	Automated processing	✗

10 Bourgne *et al.* (2022)

Key Points 10

- Introduction
 - ACE (Action-Causality-Ethics) modular framework, which divides reasoning into three layers: (i) Action model; (ii) Causality analysis layer; (iii) Ethical layer
- Action model
 - Action context
 - * Event specification
 - Type: action, expressed by domain predicate `action(E)`, or automatic event `auto(E)`
 - Preconditions `prec(F,E)` and effects `effect(E,F)`
 - Possible priorities between them: `prio(E1,E2)`
 - To account for multiple agents, action events are represented with function term `act(Ag,ActName)`, where `Ag` is an agent and `ActName` is an action type and its parameter
 - * Initial situation
 - Fluents that are initially true, using facts of the form `initially(F)`
 - Objects that are present and their type
 - Positive fluents that can be formed upon them: `pfluent(F)`
 - * Scenario
 - Represented as a set of ground instances of the `performs(A,T)` predicate, indicating that action `A` is performed at time `T`
 - A scenario is correct if all performed actions occur at the given time in the unfolding of events simulated by the event motor given the action context
 - Event motor
 - * Produces the event trace of each scenario from the initial situation and the performed actions
 - * Event trace: set of ground instances of predicate `holds(F,T)`, indicating that fluent `F` holds at time `T`, and of predicate `occurs(E,T)`, indicating that event `E` occurs between time `T` and `T+1`
 - * Event effect axioms: define the principles that govern fluents
 - A fluent holds at `T=0` if it was so initially:
$$\text{holds}(F,0) \text{ :- initially}(F).$$
 - A fluent holds at `T` if it was just initiated by an event occurrence at `T-1`:
$$\text{holds}(F,T) \text{ :- effect}(E,F), \text{ occurs}(E,T-1), \text{ pfluent}(F), \text{ time}(T).$$
 - A fluent which is true at `T` holds until the occurrence of an event which terminates it:
$$\text{holds}(F,T) \text{ :- holds}(F,T-1), \{ \text{occurs}(E,T-1) : \text{effect}(E,\text{neg}(F)) \} 0, \text{ time}(T).$$
 - It is false in all other cases (made explicit for the fluents that appear as negative preconditions):
$$\text{holds}(\text{neg}(F),T) \text{ :- prec}(\text{neg}(F),_, \text{ not holds}(F,T), \text{ time}(T).$$
 - * Event precondition axioms: characterize the behavior and the triggering of events, defining which predicates of type `occurs(E,T)` are verified at each moment

- E is possible at T when all preconditions hold:

$$\text{poss}(E,T) \text{ :- } \{\text{not holds}(F,T) : \text{prec}(F,E)\}0, \text{ event}(E), \text{ time}(T).$$

- The triggering of an automatic event means that it tries to execute:

$$\text{trigg}(U,T) \text{ :- } \text{poss}(U,T), \text{ auto}(U).$$

- The triggering of an action means that it tries to execute:

$$\text{trigg}(A,T) \text{ :- } \text{poss}(A,T), \text{ performs}(A,T), \text{ action}(A).$$

- Any triggered event occurs, unless it is **overtaken** by the occurrence of another event having priority (indicated by predicate $\text{prio}(E2,E1)$) over it:

$$\text{ovtkBy}(E1,E2,T) \text{ :- } \text{poss}(E1,T), \text{ occurs}(E2,T), \text{ prio}(E2,E1).$$

$$\text{overtaken}(E1,T) \text{ :- } \text{ovtkBy}(E1,_,T).$$

$$\text{occurs}(E,T) \text{ :- } \text{trigg}(E,T), \text{ not overtaken}(E,T).$$

- * Concurrency and event priorities: multiple events can occur at the same time, provided that they are compatible
- Detecting omission: an agent can be liable to praise or blame when it decides not to act, which is why the notion of omission is introduced
- * An omission occurs when an action is possible, not overtaken, and not performed:

$$\begin{aligned} \text{occurs}(\text{omit}(D,X,\text{no}),T) \text{ :- } & A=\text{act}(D,X), \text{ poss}(A,T), \\ & \text{not overtaken}(A,T), \\ & \text{not performs}(A,T). \end{aligned}$$

where $\text{act}(D,X)$ is an action and no a constant

- * A forced omission occurs when an action is overtaken by another action, but other choices could have been made to perform it:

$$\text{occurs}(\text{omit}(D,X,A2),T) \text{ :- } \text{ovtkBy}(\text{act}(D,X),A2,T), \text{ action}(A2).,$$

where $A2$ is an action that overturned $\text{act}(D,X)$

— Causality analysis

- Event-based causality (Berreby, Bourgne, and Ganasia, 2018): occurrence of an event resulting in the occurrence of another event
- Causal relation is represented by $r(R,A,B)$, where R is a keyword qualifying the kind of causal relation, and A and B are either of the form $o(E,T)$ (i.e., $\text{occurs}(E,T)$) or $h(F,T)$ (i.e., $\text{holds}(F,T)$)
- Causing and enabling
 - * An event ($o(F,T)$) causes its effect ($h(F,T+1)$), i.e., causes a fluent F to hold when it is its effect and remains true from T to $T+1$:

$$\begin{aligned} r(\text{causes},o(E,T1),h(F,T2)) \text{ :- } & \text{occurs}(E,T1), \text{ holds}(F,T2), \\ & \text{effect}(E,F), \{\text{not holds}(F,T) : \\ & T>T1, T<T2, \text{ time}(T)\}0, T2>T1. \end{aligned}$$

- * The occurrence of an event causes an automatic event if it causes at least one of its preconditions:

$$r(\text{causes}, h(F, T), o(U, T)) \text{ :- holds}(F, T), \text{ prec}(F, U), \\ \text{occurs}(U, T), \text{ auto}(U).$$

- * The occurrence of an event causes an action if it causes one of its preconditions, enabling an agent to perform it:

$$r(\text{enables}, h(F, T), o(A, T)) \text{ :- holds}(F, T), \text{ prec}(F, A), \\ \text{occurs}(A, T), \text{ action}(A).$$

- * Combining both steps, an event E1 causes/enables an event E2 if it causes a fluent F which causes/enables E2:

$$r(R, o(E1, T1), o(E2, T2)) \text{ :- } r(\text{causes}(E1, T1), h(F, T2)), \\ r(R, h(F, T2), o(E2, T2)), \\ T1 < T2, \text{ posRel}(R).$$

where $\text{posRel}(\text{causes}; \text{enables})$.

- * Causing is transitive:

$$r(R, o(E1, T1), o(E3, T3)) \text{ :- } r(\text{causes}, o(E1, T1), o(E2, T2)), \\ r(R, o(E2, T2), o(E3, T3)), \\ T1 \leq T3, \text{ posRel}(R).$$

– Causal effects of omissions

- * Predicate `compl` to get the complementary of a fluent literal:

$$\text{compl}(F, \text{neg}(F)) \text{ :- pfluent}(F). \\ \text{compl}(\text{neg}(F), F) \text{ :- pfluent}(F).$$

- * An omission causes the preservation of the fluents that would have been affected by an action if they are not otherwise changed by another concurrent event:

$$r(\text{causes}, o(\text{omit}(D, X, A), T1), h(F, T2)) \text{ :-} \\ \text{occurs}(\text{omit}(D, X, A), T1), \text{ holds}(F, T2), \text{ compl}(F, \text{neg}(F)), \\ \text{effect}(\text{act}(D, X), \text{neg}(F)), \\ \{\text{not holds}(F, T) : T > T1, T < T2, \text{ time}(T)\}0, T2 > T1.$$

- * When the omission is a consequence of the volition to do the other action, it is explicitly stated that the other action causes the omission:

$$r(\text{causes}, o(A, T), o(\text{omit}(D, X, A), T)) \text{ :- occurs}(\text{omit}(D, X, A), T), A \neq \text{no}.$$

- * When omitting to perform an action, one allows the events that would have been overtaken by that action to occur:

$$\begin{aligned} r(\text{allows}, o(\text{omit}(D, X, A), T), o(U, T)) &:- \\ &\text{occurs}(\text{omit}(D, X, A), T), \text{occurs}(U, T), \\ &\text{prio}(\text{act}(D, X), U), \text{auto}(U). \end{aligned}$$

- * Causal relation `allows` is transitive: `posRel(allows)`.
- Modelling causal effects of plans
- * A plan causes each of the actions to be performed:

$$r(\text{causes}, \text{plan}, o(A, T)) :- \text{occurs}(A, T), \text{performs}(A, T).$$

- * A plan causes each of the omissions that do not result from other choices:

$$r(\text{causes}, \text{plan}, o(\text{omit}(D, X, \text{no}), T)) :- \text{occurs}(\text{omit}(D, X, \text{no}), T).$$

— Ethical assessment

- Two elements for the ethical layer (cf. Berreby et al. 2017)
 - * Model of the Good: performs an evaluation of the Good produced by the actions
 - * Theories of the Right: relies on this evaluation to assess what is or is not permissible
- Model of the Good
 - * Must evaluate the good and bad produced by each event in order to qualify and quantify it
- Theories of the Right
 - * Each theory of the right `th(P)` represents a given ethical principle, and the model can infer impermissible actions according to each of its ethical principles using predicate `imp(P, A)`, indicating that `A` is impermissible according to theory `P`
 - * Here: assessment of the whole scenario using constant `plan`
 - * Maximizing principles require a comparison of different scenarios
 - To each deterministic unfolding of a scenario corresponds a unique simulation `S`, which is added to the occurrence of events
 - `o(E, T)` rewritten as `o(S, E, T)`, `plan` rewritten as `plan(S)`, `occurs(E, T)` rewritten as `occurs(S, E, T)`
 - A permissible scenario is one that is not impermissible:

$$\text{per}(P, \text{plan}(S)) :- \text{sim}(S), \text{not } \text{imp}(P, \text{plan}(S)), \text{th}(P).$$

- * Some theories of the Right: (i) Benefit Cost Analysis and Act-utilitarianism
 - Aggregating the individual good and bad of all consequences of one's actions, considering that each causal relation `R` can be given a weight `fR`:

$$w(A) = \sum_{R \in \text{posRel}} \left(\sum_{(k, x, m, e) \in G_R(A)} k \cdot f_R - \sum_{(k, x, m, e) \in B_R(A)} (f_R \times k) \right),$$

where $G_R(A) = \{(k, x, m, e) | \text{wGood}(k, e, m, k), \text{r}(R, A, e)\}$, B_R is similar to G_R using `wBad` instead, $f_{\text{causes}} = f_{\text{allows}} = 10$, and $f_{\text{enables}} = 0$

- Translated to ASP rules with sum aggregates deriving `weightAct(A, w(A))`

- Benefit cost analysis principle **benCost** (act in a way that brings about more good than bad) according to theory **th(benC)**:

imp(benC,plan(S)) :- **weightAct(plan(S),N)**, **number(N)**, **N<0**, **sim(S)**.

- Act-utilitarianism principle **actUti** (act in a way that maximizes the overall good) according to theory **th(actU)**:

imp(actU,plan(S)) :- **weightAct(plan(S),N1)**, **weightAct(plan(S2),N2)**,
N1<N2, **sim(S;S2)**.

* Some theories of the Right: (iii) Kant's 2nd categorical imperative

- Act in such a way that you treat humanity never as a means to an end, but always at the same time as an end
- Additional specification: (a) moral patient **mPatient(P)**; (b) which event affects them and how **affect(EvOcc,P,K)**, where **EvOcc** is an event occurrence, **P** a moral patient, $K \in \{-1, 1\}$, indicating whether they are positively or negatively affected; (c) what is the aim of the scenario **aim(plan(S),EvOcc)**
- Definition of end, means, and impermissible scenario according to **th(kant(1;2))**:

end(A,X) :- **affect(G,X,1)**, **aim(A,G)**,
{affect(G2,X,-1): aim(A,G2)}0.

means(A,X) :- **r(R1,A,E)**, **affect(E,X,_)**, **r(R2,E,G)**,
aim(A,G), **posRel(R1;R2)**.

imp(kant(N),plan(S)) :- **means(plan(S),X)**, **not end(plan(S),X)**,
mPatient(X).

Table 28: First dimension

	Ethical theory	Categorization
Ethical theory type	Deontological ethics	
	Consequentialism	
	Virtue ethics	
	Particularism	
	Hybrid	Hierarchically specific
		Hierarchically nonspecific
	Configurable ethics	✓(D + C)
Conflict sensitivity	Ambiguous	
	B/w AMAs	
	Within AMA	
	None	✓
Source	Developer	
	User	
	Expert	
	Lay people	
	(Legal) code/guidelines	
	Stakeholders	
	None	✓
Moral patient sensitivity	Yes	
	No	
	Unclear	✓

Table 29: Second dimension

Non-technical aspect		Categorization
Approach	Top-down	✓
	Bottom-up	
	Hybrid	
Contribution type	Model representation	✓
	Model selection	
	Judgment provision	
	Action selection/execution	
Evaluation	Test	Non-expert
		Expert
	Prove	Laws
		Model checker
		Logical proof
	Informal	Example scenario ✓
		Face validity
	None	
Domain-specific	yes/no	✗

Table 30: Third dimension

Technical aspect		Categorization
Tech type	Logical reasoning	Deductive logic ✓
		Non-monotonic logic
		Abductive logic
		Deontic logic
		Rule-based system
		Event calculus
		Knowledge representation and ontologies
		Inductive logic
	Probabilistic reasoning	Bayesian approach
		Markov models
		Statistical inference
	Learning	Inductive logic
		Decision tree
		Reinforcement learning
		Neural networks
		Evolutionary computing
	Optimization	
	Case-based reasoning	
Input	Case	Logical representation ✓
		Numerical representation
		(Structured) language representation
Implementation availability	Sensor data	
	Specification details	
	Implementation details	
Other	Code (link) provided	
	https://gitlab.lip6.fr/sarmiento/CME2021	
	Hardware	✗
	Feedback	✗
	Explanation	✗
	User interface	✗
	Automated processing	✗

11 Bringsjord *et al.* (2021)

Key Points 11

- Introduction (310-11)
 - Formalisms and techniques for an AI capable of adjudicating competing reports from subsidiary agents through time, and delivering to humans a rational, ethically correct recommendation based upon such adjudication, with concomitant justification for the recommendation
- Methods (311-18)
 - Our overall logicist approach
 - * Artificial agents/AI: agent computes a function from what is perceived (percepts) to behavior (actions), and is assumed to operate in a certain environment
 - * Logicist approach to AI/robotics: artificial agents specify and compute their functions via automated reasoning over a given formula ϕ in some formal language $\mathcal{L}$ for some logic $\mathfrak{L}$
 - * Ethics is rooted in logic
 - Our machine-ethics approach, brutally summarized
 - * First step: selection of an ethical theory (or theories) from a family of ethical theories, with the framework being general enough to be applicable to any ethical theory, provided it can be used to formalize deontic operators
 - Formalization made possible by a cognitive calculus $\mathcal{C} = \langle \mathcal{L}, \mathcal{I} \rangle$, where $\mathcal{L}$ is a formal language and $\mathcal{I}$ a collection of inference schemata
 - * Second step: automate the generation of proofs of (un-)ethical behavior
 - * Third step: integrate automated ethical reasoning into a cognitive robot's operating system
 - * Fourth step: implement the ethical OS into a physical robot and arrive at a moral machine, which can be formally verified to always act in accordance with an ethical theory
 - The specific need to handle dynamic dialectic
 - * Set of desiderata $\mathcal{D}$ detailing the capabilities an automated reasoner must have:
 - d_1 be defeasible in nature through time (non-monotonic)
 - d_2 be able to resolve inconsistencies of various sorts, and tolerate them when necessary in a manner that fully permits reasoning to continue
 - d_3 make use of values beyond standard bivalence and standard trivalence (e.g., true, false, unknown), specifically probabilities and strength factors
 - d_4 be argument-based, where the arguments have internal inference-to-inference structure both in terms of declarative formulae and inference schemata, so that detailed verification, justification/explanation is available
 - d_5 have specified inference-schemata
 - d_6 be able to allow automated reasoning over the sociocognitive elements of knowledge, belief, desire, perception, communication, emotion, etc.
 - d_7 be able to allow automated reasoning that can tackle Turing-unsolvable reasoning problems
- Results (318-23)
 - Cognitive calculus used herein
 - * Cognitive calculus employed in the present paper: $\mu\mathcal{C}$

- * Grammar of $\mu\mathcal{C}$:

$$\phi ::= \begin{cases} t : \text{Formula} | \forall t : \phi | \exists t : \phi \\ \neg\phi | \phi \wedge \psi | \phi \vee \psi | \phi \rightarrow \psi | \phi \leftrightarrow \psi \\ \mathbf{B}(a, t, \phi) | \mathbf{C}(t, \phi) \end{cases}$$

, where $t ::= x : S | c : S | f(t_1, \dots, t_n)$ denote the conventional terms (variables, constants, function symbols), $\mathbf{B}(a, t, \phi)$ denotes that agent a believes formula ϕ at time t , and $\mathbf{C}(t, \phi)$ denotes that it is common knowledge at time t that ϕ holds

- * Given a set of formulae $\Gamma = \{\phi_1, \dots, \phi_n\}$, the inference schema $\mathcal{I}$ determines whether Γ can be used to derive a given formula ψ : $\Gamma \vdash \psi$
- * $\mathcal{I}$ of $\mu\mathcal{C}$ includes the standard inference schemata for first-order logic, with quantification into modal formulae allowed, and modal inference schemata for inference with beliefs and common knowledge: (*available in original paper*)
- Generalized ethical problems
 - * Given some background knowledge Γ , ethical principle ρ tells whether action α is ethically permissible, obligatory or forbidden for agent a at time t in situation Σ : $\Gamma \cup \Sigma \vdash \rho(a, \alpha, t)$?
 - * Clausification
 - Formal principle ρ is a logicized version of informal principle $\tilde{\rho}$ stated in natural language
 - Ethical principle ρ can be decomposed into ethical relevant clauses $\rho_1, \dots, \rho_k$ such that the principle holds iff the clauses hold
 - Goal to break ρ down into clauses $\rho_1, \dots, \rho_k$ that match the informal version $\tilde{\rho}$: (*available in original paper*)
 - * Agents
 - As part of situation Σ , there is a set of agents $\{a_1, \dots, a_n\}$ each having (potentially inconsistent) beliefs about which clauses hold
 - Σ can be decomposed with the goal to ascertain what adjudicator agent a_{adj} ought to believe at t , e.g.,: (*available in original paper*)
 - Each agent a believes a subset of $\beta_a \subseteq \{\rho_1, \dots, \rho_k\} \cup \{\neg\rho_1, \dots, \neg\rho_k\}$
 - Goal: Given $\beta_{a_1}, \dots, \beta_{a_n}$, specify a procedure for computing $\mathbf{B}(a_{adj}, t', \rho)$, where $t' > now$.
- Solution to a generalized ethical problem
 - * Adjudication framework is based upon uncertainty system $\mathcal{S}$
 - * $\mathcal{S}$ assumes as a primitive the reasonableness operator $(\phi \succeq_t^\alpha \psi)$, meaning that agent a at time t finds ϕ to be at least as reasonable as ψ
 - * Withholding operator $\mathbf{W}(a, t, \phi) \equiv \neg\mathbf{B}(a, t, \phi) \wedge \neg\mathbf{B}(a, t, \neg\phi)$
 - * Where Θ and Ω are variables denoting one of the modal operators $\{\mathbf{B}, \mathbf{W}\}$, multi-agent reasonableness is given by: (*available in original paper*)
 - * Multi-agent reasonableness: For every agent a_i that has an Ω formula in ψ , there is an agent a_j that has a Θ formula in ϕ
 - * Four uncertainty levels
 - Level 1: Agent a believes at least one other agent a_i believes that ϕ : (*available in original paper*)
 - Level 2: Agent a believes that it is more reasonable to believe ϕ than withhold ϕ : (*available in original paper*)
 - Level 3: Agent a believes that it is more reasonable to believe ϕ than believe $\neg\phi$: (*available in original paper*)

- Level 4: Agent a believes that every agent believes ϕ : (*available in original paper*)
- Instantiation of the generalized problem: a scenario (drone attack)

Table 31: First dimension

	Ethical theory	Categorization
Ethical theory type	Deontological ethics	
	Consequentialism	
	Virtue ethics	
	Particularism	
	Hybrid	Hierarchically specific
		Hierarchically nonspecific
	Configurable ethics	✓(D)
Conflict sensitivity	Ambiguous	
	B/w AMAs	✓
	Within AMA	
	None	
Source	Developer	
	User	
	Expert	
	Lay people	
	(Legal) code/guidelines	
	Stakeholders	
	None	✓
Moral patient sensitivity	Yes	
	No	
	Unclear	✓

Table 32: Second dimension

Non-technical aspect		Categorization
Approach	Top-down	✓
	Bottom-up	
	Hybrid	
Contribution type	Model representation	
	Model selection	
	Judgment provision	✓
	Action selection/execution	✓
Evaluation	Test	Non-expert
		Expert
	Prove	Laws
		Model checker
		Logical proof
		Example scenario ✓
	Informal	Face validity
		None
Domain-specific	yes/no	✗

Table 33: Third dimension

Technical aspect		Categorization
Tech type	Logical reasoning	Deductive logic
		Non-monotonic logic ✓
		Abductive logic
		Deontic logic
		Rule-based system
		Event calculus
		Knowledge representation and ontologies
		Inductive logic
	Probabilistic reasoning	Bayesian approach
		Markov models
		Statistical inference
	Learning	Inductive logic
		Decision tree
		Reinforcement learning
		Neural networks
		Evolutionary computing
	Optimization	
	Case-based reasoning	
Input	Case	Logical representation ✓
		Numerical representation
		(Structured) language representation
Implementation availability	Sensor data	
	Specification details ✓	
	Implementation details	
Other	Code (link) provided	
	Hardware ✗	
	Feedback ✗	
	Explanation ✗	
	User interface ✗	
	Automated processing ✗	

12 Chen *et al.* (2022)

Key Points 12

- Introduction
 - Flexible robotic architecture with cases derived from different ethical frameworks, which potentially allows a robot to produce morally acceptable actions based on the selected ethical framework and the current situation
- Background
- Architecture design
 - *Figure available in original paper*
 - Goal to enable a robot to use various ethical frameworks for more robust ethical decision-making in HRI
 - Use of case-based reasoning, with multiple cases for each ethical framework that are drawn from surveys of laypersons or experts
 - Case selection module will find the most relevant case for a chosen ethical framework using similarity measures between the current situation and the case indices
 - The action selection module will then output an appropriate action for the robot to execute
- Architecture implementation
 - Example scenario: pill sorting with an older adult, with the goal to find out whether it is moral to deceive an adult in a pill sorting task to keep them engaged with the task
 - Case base derived from survey studies
 - Ethical architecture (*figure available in original paper*)
 - Case selection module selects the most relevant case for a chosen ethical framework based on the current situation (user task performance and user emotional state)
 - The robot will use behavior probabilities to generate different behaviors corresponding to an action
 - Using action probabilities and behavior probabilities, the robot outputs an action by performing a gesture

Table 34: First dimension

	Ethical theory	Categorization
Ethical theory type	Deontological ethics	
	Consequentialism	
	Virtue ethics	
	Particularism	
	Hybrid	Hierarchically specific
		Hierarchically nonspecific
	Configurable ethics	✓(Hyb: HNS (D + C + V + A (folk morality) + others))
Conflict sensitivity	Ambiguous	
	B/w AMAs	
	Within AMA	✓
	None	
Source	Developer	
	User	
	Expert	✓
	Lay people	✓
	(Legal) code/guidelines	
	Stakeholders	
	None	
Moral patient sensitivity	Yes	✓
	No	
	Unclear	

Table 35: Second dimension

Non-technical aspect		Categorization
Approach	Top-down	✓
	Bottom-up	
	Hybrid	
Contribution type	Model representation	
	Model selection	
	Judgment provision	
	Action selection/execution	✓
Evaluation	Test	Non-expert
		Expert
	Prove	Laws
		Model checker
		Logical proof
	Informal	Example scenario ✓
		Face validity
	None	
Domain-specific	yes/no	✗

Table 36: Third dimension

Technical aspect		Categorization
Tech type	Logical reasoning	Deductive logic
		Non-monotonic logic
		Abductive logic
		Deontic logic
		Rule-based system
		Event calculus
		Knowledge representation and ontologies
	Probabilistic reasoning	Inductive logic
		Bayesian approach
		Markov models
	Learning	Statistical inference
		Inductive logic
		Decision tree
		Reinforcement learning
		Neural networks
		Evolutionary computing
	Optimization	
	Case-based reasoning	✓
Input	Case	Logical representation
		Numerical representation
		(Structured) language representation
Implementation availability	Sensor data	
	Specification details	
	Implementation details	
Other	Code (link) provided	
	Hardware	
	Feedback	
	Explanation	
	User interface	
	Automated processing	

13 Giancola *et al.* (2022)

Key Points 13

- Introduction
 - Given an obligation and a set of potentially-inconsistent, ethically-charged beliefs, how can an artificially-intelligent agent ensure that its actions maximize the likelihood that the obligation is satisfied?
- Cognitive calculi
 - Two components of a cognitive calculus
 - * Multi-sorted n^{th} -order logic with modal operators for modeling cognitive attitudes
 - * Inference schemata that express the semantics of the modal operators
 - Deontic cognitive event calculus
 - * Signature of $\mathcal{DC}\mathcal{EC}$ (*available in original paper*)
 - * Inference schemata of $\mathcal{DC}\mathcal{EC}$ (*available in original paper*)
 - Inductive deontic cognitive event calculus
 - * Two components to go from a deductive system to an inductive system, i.e., one that allows to reason about uncertainty: (1) an uncertainty system, and (2) inference schemata that delineate methods by which inferences linking formulae and other information can be used to build formally valid arguments
 - * Additional syntax for $\mathcal{IDC}\mathcal{EC}$ (*available in original paper*)
 - * Additional inference schemata for $\mathcal{IDC}\mathcal{EC}$ (*available in original paper*)
 - First inference schema: agents are able to infer evident beliefs ($\sigma = 4$) from what they perceive
 - Second inference schema: agent is able to infer a belief that is provable from the beliefs they currently assert, so long as the belief set is not inconsistent, where atom ζ can only be proved if $\{\phi_1, \dots, \phi_m\}$ is inconsistent
 - Third inference schema: specifies how uncertainty for $\neg\phi$ is derived from uncertainty for ϕ
 - Cognitive likelihood
 - * 11 cognitive likelihood values used
 - * Likelihood values are obtained by independently justifying each likelihood value by appeal to rational human-level cognition
 - * Only non-negative likelihood values need to be defined due to the third additional inference schema
- Highly-expressive automated planning
 - Stanford Research Institute Problem Solver (STRIPS): (i) set of formulae describing the initial state of the world; (ii) set of actions which describe methods by which the planner can change the world state, where an action consists of (1) a set of preconditions, (2) a set of additions (formulas added to the world by taking the action), and (3) a set of deletions; (iii) goal set which denotes these formulae that the agent in question wishes to hold
 - SPECTRA is a STRIPS-style planner which provides a planning formalisms that can handle arbitrary formulae for describing the world, states of minds, and an unknown set of objects
- Selecting plans using cognitive likelihood
 - Agents are given the following: (1) an obligation, (2) knowledge regarding the conditions required to satisfy the obligation, and (3) a set of (potentially inconsistent) ethically-charged beliefs regarding actions the agent can take to affect the status of the obligation
 - Decision-making framework (*figure available in original paper*)

- * An agent a is obligated to perform action α , given that it believes some precondition ϕ holds, and it knows the conditions that will enable α to happen
- * a has a set of beliefs regarding formulae pertinent to its obligations, and various subsets of those beliefs are passed to Spectra, with the goal of generating plans which cause α to occur
- * Each of those plans is assigned a likelihood based on the likelihood of the weakest belief required to generate the plan
- * Select the plan with the highest likelihood as the one to enact

Table 37: First dimension

	Ethical theory	Categorization
Ethical theory type	Deontological ethics	✓
	Consequentialism	
	Virtue ethics	
	Particularism	
	Hybrid	Hierarchically specific
		Hierarchically nonspecific
	Configurable ethics	
	Ambiguous	
Conflict sensitivity	B/w AMAs	✓
	Within AMA	
	None	
Source	Developer	
	User	
	Expert	
	Lay people	
	(Legal) code/guidelines	
	Stakeholders	
	None	✓
Moral patient sensitivity	Yes	
	No	
	Unclear	✓

Table 38: Second dimension

Non-technical aspect		Categorization
Approach	Top-down	✓
	Bottom-up	
	Hybrid	
Contribution type	Model representation	
	Model selection	
	Judgment provision	
	Action selection/execution	✓
Evaluation	Test	Non-expert
		Expert
	Prove	Laws
		Model checker
		Logical proof
	Informal	Example scenario ✓
		Face validity
	None	
Domain-specific	yes/no	✗

Table 39: Third dimension

Technical aspect		Categorization
Tech type	Logical reasoning	Deductive logic
		Non-monotonic logic
		Abductive logic
		Deontic logic
		Rule-based system
		Event calculus
		Knowledge representation and ontologies
		Inductive logic ✓
	Probabilistic reasoning	Bayesian approach
		Markov models
		Statistical inference
	Learning	Inductive logic
		Decision tree
		Reinforcement learning
		Neural networks
		Evolutionary computing
	Optimization	
	Case-based reasoning	
Input	Case	Logical representation ✓
		Numerical representation
		(Structured) language representation
Implementation availability	Sensor data	
	Specification details ✓	
	Implementation details	
Other	Code (link) provided	
	Hardware ✗	
	Feedback ✗	
	Explanation ✗	
	User interface ✗	
	Automated processing ✗	

14 Kolker *et al.* (2023)

Key Points 14

- Introduction (1-3)
 - Operationalizes and implements the hypothetical retrospection procedure, i.e., a mechanism for handling uncertainty that can be adapted to different ethical theories
- Related work (4-5)
- Hypothetical retrospection (5-6)
 - Hansson: a potential choice is hypothetically retrospected upon from the endpoint of its major foreseeable outcomes, meaning that potential branches of future development are used to build resonate arguments about what to do in the present
 - Major search principle among the branches of future development: find the most probable future developments which are the most difficult to defend morally
 - Ideally, branches are described with the same type of information
 - Implementation assesses actions assuming their potential branches are provided
 - Using their potential branches, actions are compared to each others with a selection of ethical theories
 - Search for an action which never leads an agent to morally regret its choice in retrospect, meaning that it is not logically flawed under retrospection
- Implementation (7-12)
 - Formalism (7-9)
 - * Ethical decision problem: $\langle A, B, S, U, F, I, m \rangle$
 - * S : environment's ethically relevant properties of Boolean variables
 - * I : initial truth assignments to S , before actions are taken
 - * U : ordered set of utility classes in descending order of importance
 - Utility class: unordered set of individual utility assignments represent as tuples of $\langle s_k, \phi, v \rangle$, where s_k denotes a state variable in S and $v \in \mathbb{R}$ represents the variable's utility when assigned Boolean value ϕ
 - In U , where $i < j$, all the positive utilites in u_i are considered greater than any utility in u_j ; all the negative utilities in u_i are considered less than any utility in u_j
 - * F : set of states forbidden by a given deontological theory
 - Forbidden state is a tuple $\langle s, \phi \rangle$, where $s \in S$ is a state variable forbidden from being assigned the Boolean value ϕ
 - * A : set of available actions
 - * B : set of all potential branches of future development
 - * m : mapping $A \rightarrow \mathcal{P}(B)$ that associates each action with its potential branches of future development, where $\mathcal{P}$ is the powerset function
 - Each branch $b \in m(a)$ is an ordered sequence of events that could occur after action a
 - Event: tuple $\langle s, \phi, p \rangle$, where $s \in S$, ϕ is the new Boolean value of s , and p is the probability that the event occurs
 - Branch: sequence of events that can occur after the action is taken
 - * Permissible action: Given an ethical decision problem $\langle A, B, S, U, F, I, m \rangle$, a permissible action is an action $a \in A$ such that for all potential branches of future development $b \in m(a)$, there is an acceptability over their events in state space S . If no such actions exist, action a is permissible if it maximizes the cumulative probability of its acceptable branches.
 - Probability representation (9)

- * Relation of poetic words to mathematical probability ranges (*table available in original paper*)
- Argumentation model (9-10)
 - * $Argument(b)$, i.e., argument for an action $a \in A$, selected in initial state I , resulting in branch $b \in m(a)$ with probability p : *From the initial state I , it was acceptable to perform action a , resulting in consequences b with probability p .*
 - * For branches $b_1 \in m(a_1)$, $b_2 \in m(a_2)$ and any generic moral principle, the following critical questions are asked for $Argument(b_1)$ to attack $Argument(b_2)$:
 - CQ1** *Did b_2 violate a moral principle that b_1 did not?*
 - CQ2** *Did a_2 hold a greater probability of breaking the moral principle than a_1 ?*
 If both questions are answered positively, $Argument(b_1)$ attacks $Argument(b_2)$
 - * For utilitarianism, the questions are embedded as follows:
 - CQ1** *Did b_2 bring greater utility value than b_1 ?*
 - CQ2** *Did a_2 expect greater utility value than a_1 ?*
 - * For a generic deontological do-no-harm principle critical questions are embedded as follows:
 - CQ1** *Did b_2 cause harm where b_1 did not?*
 - CQ2** *Did a_2 expect greater probability of causing harm than a_1 ?*
 - * After searching for all attacks on all branches, an action should be selected with complete acceptability. If no such action exists, an action should be selected with maximal acceptability, i.e., summing the probability of each non-attacked argument and selecting an action with a maximal sum.
- Algorithm (10-12)
 - * Algorithm 1: action selection (*pseudo-code available in original paper*)
 - Lines 2-4: All actions are compared by their potential branches of future development.
 - Lines 5-6: Attacks are generated between arguments by asking critical questions (based on a Utilitarian and a Deontological theory).
 - Lines 7-13: Attacked branches are marked.
 - Lines 17-25: The action with maximum acceptability is selected.
 - * Algorithm 2: Utilitarianism (*pseudo-code available in original paper*)
 - Lines 1-9: Beginning with the lowest utility class, the first class is found where branch utilities are unequal (making critical question 1 positive), and the branch with the greater utility becomes the attacker, the other the defender.
 - Lines 10-12: If utilities are equal through all classes, there are no attacks.
 - Lines 13-17: The defender branch attempts to use the foresight argument to defend itself: For each lower indexed class, if the defender's action has greater expected utility, defence is successful and there is no attack.
 - Line 18: If the attacker action has greater or equal expected utility across all classes, defence fails, critical question 2 is positive, and the defender branch thus attacked.
 - * Algorithm 3: Deontology (*pseudo-code available in original paper*)
 - Lines 1-3: Iterate across the set of forbidden assignments and check the events in either for a violation.
 - Line 4 and 9: If one branch has a violation that the other has not, critical question 1 is positive.
 - Lines 4-13: If the violating branch's action does not have a greater probability of not making the assignment, critical question 2 is positive and the index of the violating branch is returned.
 - Line 15: If no branch is attacked, neither index is returned.

Table 40: First dimension

	Ethical theory	Categorization
Ethical theory type	Deontological ethics	
	Consequentialism	
	Virtue ethics	
	Particularism	
	Hybrid	Hierarchically specific
		Hierarchically nonspecific
	Configurable ethics	✓(D + C + Hyb: HNS)
Conflict sensitivity	Ambiguous	
	B/w AMAs	
	Within AMA	✓
	None	
Source	Developer	
	User	
	Expert	
	Lay people	
	(Legal) code/guidelines	
	Stakeholders	
	None	✓
Moral patient sensitivity	Yes	
	No	
	Unclear	✓

Table 41: Second dimension

Non-technical aspect		Categorization
Approach	Top-down	✓
	Bottom-up	
	Hybrid	
Contribution type	Model representation	
	Model selection	
	Judgment provision	
	Action selection/execution	✓
Evaluation	Test	Non-expert
		Expert
	Prove	Laws
		Model checker
		Logical proof
	Informal	Example scenario ✓
		Face validity
	None	
Domain-specific	yes/no	✗

Table 42: Third dimension

Technical aspect		Categorization
Tech type	Logical reasoning	Deductive logic
		Non-monotonic logic
		Abductive logic
		Deontic logic
		Rule-based system
		Event calculus
		Knowledge representation and ontologies
		Inductive logic
	Probabilistic reasoning	Bayesian approach
		Markov models
		Statistical inference
	Learning	Inductive logic
		Decision tree
		Reinforcement learning
		Neural networks
		Evolutionary computing
	Optimization	✓
	Case-based reasoning	
Input	Case	Logical representation
		Numerical representation
		(Structured) language representation
	Sensor data	
Implementation availability	Specification details	
	Implementation details	
	Code (link) provided	https://github.com/sameysimon/HypotheticalRetrospectionMachine
Other	Hardware	✗
	Feedback	✗
	Explanation	✗
	User interface	✗
	Automated processing	✗

15 Medda *et al.* (2024)

Key Points 15

- Introduction
 - Framework GNNUERS (GNN-based Unfairness Explainer in Recommender Systems) generates a set of recommendations that are fair for end users, guided by the demographic parity principle
- Related work
- Problem formulation
 - Recommendation task
 - * Goal to predict whether or to what extent an unseen item $i \in I$ would be of interest to a user $u \in U$ based on I_u , i.e., the set of items u has interacted with in the past
 - * User-item interactions as an undirected bipartite graph: $G = (U, I, E)$
 - $U \cup I$: set of vertices, with $n = |U| + |I|$
 - E : set of edges, with $E = \{(u, i) | u \in U, i \in I\}$
 - * GNNs can be used to predict potentially interesting links between users U and items I in G
 - * One-layer GNN: $f(A, W) \rightarrow \hat{R}$
 - A : $n \times n$ adjacency matrix representing G
 - W : learned weight matrix of f
 - $\hat{R}_{u,i}$: linking probability between user u and item i
 - * Function f is defined as $f(\text{softmax}[D^{-\frac{1}{2}}AD^{-\frac{1}{2}}], W)$
 - $D_{i,j} = \sum_j A_{i,j}$ are entries in the degree matrix D
 - W : weight matrix
 - g : combining function depending on the GNN implementation (here softmax)
 - * Q_u : list of the top- k items recommended to user u from $\hat{R}_u$, i.e., items I sorted in decreasing relevance given the user-item relevance matrix $\hat{R}$ and user u
 - * Q : set of all recommended lists
 - Unfairness explanation task
 - * Counter-factual explanation: set of interactions perturbed with respect to the original user-item interactions when the GNN produces at least one different recommendation to users and when these perturbed interactions are used for training
 - * Generate $\tilde{A}$, a perturbed version of adjacency matrix A , that leads to recommended lists $\tilde{Q}$, with $\tilde{Q} - Q \neq \emptyset$, when $\tilde{A}$ is used for training instead of A
 - * Produce a perturbed adjacency matrix $\tilde{A}$ that leads to the highest fairness across users by means of the lowest number of perturbations on the original adjacency matrix A
 - * Fairness is modeled according to the notion of demographic parity: $dp(Q) : Q \rightarrow \mathbb{R}$
 - * Generate $\tilde{A}$ that results in recommended lists $\tilde{Q}$, where $dp(\tilde{Q}) > dp(Q)$, constrained to the number of perturbed edges with respect to A ; that is, given GNN f , the goal is to minimize the following objective function: $\mathcal{L}(A, \tilde{A}) = \mathcal{L}_{fair}(A, f(\tilde{A}, W)) + \mathcal{L}_{dist}(A, \tilde{A})$
 - $\mathcal{L}_{fair}$ monitors fairness, operationalized as the complement of the demographic parity function h (previously called dp in the paper)
 - $\mathcal{L}_{dist}$ is the term controlling the distance between the perturbed adjacency matrix $\tilde{A}$ and the original one A , operationalized with a distance function d
- GNNUERS
 - Bi-partite graph perturbation
 - * Obtaining vector p for perturbation, which is of size B , i.e., the number of existing edges in the original graph

- First, a real-valued vector $\hat{p}$ is defined
- Sigmoid transformation is applied
- Binarization of entries such that values ≥ 0.5 become 1 and values < 0.5 become 0, which results in p
- The initialization of $\hat{p}$ should guarantee $\tilde{A} = A$, i.e., a real-valued α is selected to initialize $\hat{p}$ such that $p_i = 1, \forall i \in [0, B]$
- * Perturbed matrix $\tilde{A}$ is populated as follows: $\tilde{A}_{u,i} = \begin{cases} p_{h(u,i)} & \text{if } 0 \leq h(u,i) < B \\ A_{u,i} & \text{otherwise} \end{cases}$
 - Mapping of 2D indices (u, i) of A to 1D index for p , i.e., $h : \mathbb{N}^{|U|} \times \mathbb{N}^{|I|} \rightarrow \mathbb{N}_{<B}$
 - Optimization of p : deletion of non-zero entries $A_{u,i} \neq 0$, such that an edge $A_{u,i}$ is deleted if $p_{h(u,i)} = 0$
- Perturbed graph generation
 - * Counterfactual model $\tilde{f}(A, W; \hat{p}) \rightarrow \tilde{R}$ extends f using p as parameter and the weights W learnt by f as additional input
 - $\tilde{f}$ predicts the altered relevance matrix $\tilde{R}$ by combining the normalized version of the perturbed adjacency matrix $\tilde{L} = \tilde{D}^{-\frac{1}{2}} \tilde{A} \tilde{D}^{-\frac{1}{2}}$ and W according to the implementation of the original model $f(\tilde{D}_{u,u} = \sum_i \tilde{A}_{u,i})^a$
 - $\tilde{f}$ learns only , while the weights W remain constant
- Loss function optimization
 - * Demographic Parity (DP) is operationalized as the mean of the absolute pair-wise utility difference across all demographic groups: $\mathcal{L}_{fair}(\tilde{R}, A) = \frac{1}{\binom{|G|}{2}} \sum_{1 \leq i \leq j \leq |G|} \left\| S(\tilde{R}^{G_i}, A^{G_i}) - S(\tilde{R}^{G_j}, A^{G_j}) \right\|_2^2$
 - G : set of considered demographic groups
 - S : function that measures the recommendations utility level
 - * Group with the higher utility is denoted as unprotected, whereas the one with the lower utility is denoted as protected
 - * Approximate version of Normalized Discounted Cumulative Gain (NDCG) is used as utility metric S :
$$NDCG_{ApproxLoss}(r, a) = - \frac{1}{DCG(a, a)} \sum_i \frac{2^{a_i} - 1}{\log_2(1 + z_i)}$$

$$\text{s.t. } z_i = 1 + \sum_{j \neq i} \sigma\left(\frac{r_j - r_i}{\gamma}\right)$$
 - DCG: Discounted Cumulative Gain
 - r : item relevance score produced by the recommender system
 - γ : scaling constant
 - * Distance function d is based on the absolute element-wise difference between $\tilde{A}$ and A :
$$\mathcal{L}_{dist} = \beta \frac{1}{2} \sigma\left(\sum_{i,j} \left\| \tilde{A}_{i,j} - A_{i,j} \right\|_2^2\right)$$
 - Sigmoid function $\sigma(x) = \frac{|x|}{1+|x|}$ used to bound distance loss to the same range of $\mathcal{L}_{fair}$, i.e., $[0, 1]$
 - β : normalization parameter that balances the losses due to the trend of $\mathcal{L}_{fair}$ to report values $<< 0.5$
- Gradient deactivation
 - * Edges are perturbed for two objectives: (1) increase the utility for the protected group,

- and (2) decrease the utility for the unprotected group
- * Edges perturbed for one objective could negatively affect the other objective
- * Gradient deactivation is performed on the recommendations generated for the protected group: back-propagation updates the perturbation vector only from the unprotected group viewpoint
- * GNNUERS objective, thus, is to delete edges generating the gap in recommendation utility between unprotected and protected users
- * The term deactivated is used to characterize a group associated with inactive gradient, and the term activated is used for a group associated with active gradient
- Resource usage
- Evaluation
 - Graph topological properties used for evaluation: degree (DEG), sparsity (SP), reachability (RB)
 - Data preparation: MovieLens 1M (ML-1M) corpus, Last.FM 1K (LFM-1K) corpus, Insurance (INS) corpus, Ta Feng (FENG) corpus on grocery shopping, where for each user 20% of user-item interactions form the test set, 10% the validation set, and 70% the training set
 - GNN-based recommender systems on which GNNUERS was adopted: GCMC, NGCF, LightGCN
 - Explanation baseline methods: GNNUERS and GNNUERS+CN (which limits the perturbation to the edges connected to used nodes of the advantaged demographic group) are compared against RND-P which randomly perturbs edges
 - Unfairness explainability: except extreme cases, GNNUERS selects edges that systematically and significantly explain unfairness, regardless of the data, models and demographic groups on which it is applied
 - Impact on recommendation utility: GNNUERS and GNNUERS+CN reduce the utility for the unprotected group, detecting edges that generate a disparity in performance, while reporting a negligible loss for the protected group
 - Edge selection process: GNNUERS edges perturbation process can be described by the dependency from user nodes SP, RB, and item nodes DEG
- Discussions and limitations

^aThis looks wrong.

Table 43: First dimension

	Ethical theory	Categorization
Ethical theory type	Deontological ethics	
	Consequentialism	✓
	Virtue ethics	
	Particularism	
	Hybrid	Hierarchically specific
		Hierarchically nonspecific
	Configurable ethics	
	Ambiguous	
Conflict sensitivity	B/w AMAs	
	Within AMA	
	None	✓
Source	Developer	
	User	
	Expert	
	Lay people	
	(Legal) code/guidelines	
	Stakeholders	
	None	✓
Moral patient sensitivity	Yes	
	No	
	Unclear	✓

Table 44: Second dimension

Non-technical aspect		Categorization
Approach	Top-down	
	Bottom-up	✓
	Hybrid	
Contribution type	Model representation	
	Model selection	
	Judgment provision	
	Action selection/execution	✓
Evaluation	Test	Non-expert
		Expert
	Prove	Laws
		Model checker
		Logical proof
	Informal	Example scenario ✓
		Face validity
	None	
Domain-specific	yes/no	✓

Table 45: Third dimension

Technical aspect		Categorization
Tech type	Logical reasoning	Deductive logic
		Non-monotonic logic
		Abductive logic
		Deontic logic
		Rule-based system
		Event calculus
		Knowledge representation and ontologies
		Inductive logic
	Probabilistic reasoning	Bayesian approach
		Markov models
		Statistical inference
	Learning	Inductive logic
		Decision tree
		Reinforcement learning
		Neural networks ✓
		Evolutionary computing
	Optimization	
	Case-based reasoning	
Input	Case	Logical representation
		Numerical representation ✓
		(Structured) language representation
Implementation availability	Sensor data	
	Specification details	
	Implementation details	
Other	Code (link) provided	
	https://github.com/jackmedda/R-S-BGExplainer	
	Hardware	✗
	Feedback	✗
	Explanation	✗
	User interface	✗
	Automated processing	✗

16 Rosen *et al.* (2022)

Key Points 16

- Introduction (1634-5)
 - Paper develops algorithms that represent and learn norm-conforming reward models, which learn from human teaching
 - Two kinds of teaching: (1) the teacher instructs the learner, (2) the teacher lets the learner explore and discover, guided by varying amounts of feedback
 - Approach uses behavioral research insights into how humans teach robots norms and uses these to develop new learning algorithms
- Related work (1635)
- Preliminaries (1635-7)
 - Human teaching of robots: empirical study
 - * Teaching by evaluation was favored when a teacher had gathered immediate or accumulated evidence about the robot’s learning progress and also for reoccurring tasks
 - * Teaching by instruction was favored when people were just starting out in their teaching session and when they observed a surprising setback in learning progress
 - Technical background
 - * Markov decision processes (MDPs)
 - MDP represented by tuple $(\mathcal{S}, \mathcal{A}, T, \gamma, R)$, where $\mathcal{S}$ is the set of states, $\mathcal{A}$ is the set of actions, $T : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \rightarrow [0, 1]$ represents the transition dynamic of the environment, $\gamma \in [0, 1]$ is a scalar discount factor, and $R : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \rightarrow \mathbb{R}$ is a reward model
 - Value function $V^\pi(s) = \mathbb{E}[\sum_{k=0}^{\infty} \gamma^k r_{t+k+1} | s = s_t]$ represents the expected return when starting in state s and acting under policy π
 - State-action value function $Q^\pi(s, a) = \mathbb{E}[\sum_{k=0}^{\infty} \gamma^k r_{t+k+1} | s = s_t, a = a_t]$ is the expected return when starting in state s , taking action a , and acting forever according to policy π
 - Optimal policy π^* is the policy that maximizes $V^\pi(s)$ for all $s \in \mathcal{S}$
 - Advantage function $A^\pi(s, a) = Q^\pi(s, a) - V^\pi(s)$ measures the relative value of actions $a \in \mathcal{A}$ that can be taken from state s when acting under policy π (when Q^π and V^π are known)
 - If π is optimal, taking optimal action a^* from s has advantage 0
 - * Reward models
 - When a MDP lacks a reward model (MDP/ R), it may be learned from environment signals
 - When states are feature-based, the reward model can be expressed as $R(s, a, s') = R'(s) = \mathbf{w}^T \phi(s)$, where $\phi : \mathcal{S} \rightarrow \mathbb{R}^m$ maps from state space to feature space, and $\mathbf{w} \in \mathbb{R}^m$ are learned feature weights, where m is the number of dimensions of the feature space
 - * Markov chains
 - Discrete-time Markov Chain consists of a sequence of random variables $\{X_t\}_{t \in \mathbb{N}}$, an initial state distribution $\mathcal{D}$, from which initial state $X_0 \sim \mathcal{D}$ is sampled, a set of transition probabilities $P(X_{t+1} = x_j | X_t = x_i)$ that satisfy the Markov property, in which the probability of transitioning to the next state x_j depends only on the current state x_i
 - When simulating a MDP as a Markov Chain, the transition probabilities can be computed from the transition dynamics T and the current stochastic policy π
 - * Absorbing states

- State x_a is absorbing if $P(X_{t+1} = x_j | X_t = x_a) = \delta_{aj}$, where δ_{aj} is 1 if $a = j$ and 0 if $a \neq j$
- When the goal state of a discrete MDP is absorbing, a Markov Chain can be used to simulate the exact probability that an agent ends up in a goal state after a certain number of actions based on the initial state distribution and the agent's policy
- * Feedback
 - When the reward is explicitly available: the consequence of an agent's action a in state s resulting in a transition to state s' is a scalar-reward from R
 - Instructive feedback: sequence of state-action pairs $\{(s_i, a_i)\}_{i \leq N}$, where $N \in \mathbb{N}$ represents the length of an exemplary trajectory, and action a_i is the desired (or optimal) action that should be taken from state s_i
 - Evaluative feedback: like rewards, it is scalar, but, unlike rewards, the returned values for equivalent state-action pairs can change over time (e.g., when evaluative feedback is based on policy-dependent advantage $A^\pi(s, a)$)
- Methods and technical approach (1637-8)
 - Norm-learning from feedback
 - * General approach
 - The agent models the environment as a discrete MDP
 - The agent maintains a current estimate of a feature-based reward model $R(s) = \mathbf{w}^T \phi(s)$, where the weights $\mathbf{w}$ are learned based on feedback
 - Initially, $\mathbf{w} = \mathbf{0}$, because all policies are optimal under 0 reward
 - * Differentiable planning
 - Agent selects an action by solving the MDP under the agent's current reward model estimate
 - Agent performs dynamic programming with the softmax Bellman operator to a finite number of iterations to receive approximations V^π and Q^π , where π is the softmax of the Q -values (*pseudo-code available in original paper*)
 - All outputs are differentiable with respect to $\mathbf{w}$
 - Once π is received, the agent picks the optimal action $a^* = \operatorname{argmax}_a \pi(a|s) = \operatorname{argmax}_a Q^\pi(s, a)$ to take in current state s
 - * Learning reward from feedback
 - a. Instruction-based (policy-independent) feedback
 - Whenever instruction feedback is received by the agent, each (s, a) in the sequence is added to feedback buffer $\mathcal{B}$
 - The agent seeks to maximize the likelihood that its policy takes the preferred actions in $\mathcal{B}$ using gradient update $\mathbf{w} \leftarrow \mathbf{w} + \alpha \nabla_{\mathbf{w}} [\log L]$ with learning rate α , where $L(\pi; \mathbf{w}) = \prod \pi(a|s; \mathbf{w})$, and $\nabla_{\mathbf{w}} [\log L] = \sum \frac{\nabla_{\mathbf{w}} \pi(a|s; \mathbf{w})}{\pi(a|s; \mathbf{w})}$, with the product and sums taken over all $(s, a) \in \mathcal{B}$
 - b. Scalar-based (policy-dependent) evaluative feedback f
 - Feedback value of f is used to update the probability of the action a just taken by the agent from state s according to psuedoloss L_p whose gradient with respect to $\mathbf{w}$ is $\nabla_{\mathbf{w}} L_p = \frac{\nabla_{\mathbf{w}} \pi(a|s; \mathbf{w})}{\pi(a|s; \mathbf{w})} f$
 - Positive scalar values of f increase the likelihood of taking a in s , whereas negative values decrease the likelihood
 - Contexts and automated teaching
 - * Each teacher is provided with (a) access to a copy of the MDP/ R the agent is acting in, (b) the agent's current policy π , (c) the state-action pair (s, a) the agent has just acted on, (d) a violation set $\mathcal{V} \subset \mathcal{S}$, which indirectly encodes that actions that would transition into $\mathcal{V}$ are prohibited

- * Each automated teacher is assigned a feedback strategy to act under: provide demonstration and evaluative feedback according to probability distribution $\mathcal{P}$, which stays constant during training
- * Demonstration feedback
 - Automated teachers provide single-step-length demonstrations representing the preferred action a the teacher believes the agent should take from state s under context $\mathcal{V}$
 - In simulation, the automated teacher pre-compute their preferred actions in every state under $\mathcal{V}$ according to the least path cost for reaching the goal state in the MDP/ R
 - Passing through or occupying states $s \in \mathcal{V}$ incurs high cost compared to states $s \notin \mathcal{V}$
- * Evaluative feedback
 - Teachers provide policy-dependent advantage-based feedback by evaluating the agent's policy under the teacher's internal ground truth reward model R^* based on context $\mathcal{V}$ to arrive at $A^\pi(s, a)$, which is returned as f to the agent
- Evaluation metrics
 - * Stochastic goal success
 - A H -step walk of a Markov Chain is simulated starting from an initial uniform distribution over all non-goal states, and returning the total probability of being in an absorbing goal state after H steps
 - Transition probability matrix $\mathbf{P}^\pi$ with elements $P^\pi(s'|s) = \sum_a \pi(a|s)T(s'|s, a)$ for $s \notin \mathcal{S}_g$ is constructed from the agent's current stochastic policy π and the environmental transition matrix T
 - Set of goal states $\mathcal{S}_g$ is set as absorbing with $P^\pi(s's) = \delta_{ss'}$ if $s \in \mathcal{S}_g$
 - Starting from an initial uniform state distribution represented by vector $\mathbf{b}_0$ over all non-goal states allows to capture the probability $b_H(s)$ of ending up in a particular state $s \in \mathcal{S}$ after H -step roll-outs: $\mathbf{b}_H = \mathbf{b}_0(\mathbf{P}^\pi)^H$
 - Stochastic goal success: $\sum_{s \in \mathcal{S}} b_H(s)$
 - * Stochastic policy violations
 - For stochastic violations $\frac{1}{|\mathcal{S}|} \sum_s \sum_a \pi(a|s)T(s'|s, a)B(s, a, s')$, the probability mass that is associated with violating actions in the policy π is normalized over all states
 - Violation matrix elements $B(s, a, s') = \begin{cases} 0 & \text{if } s' \notin \mathcal{V} \\ 1 & \text{if } s' \in \mathcal{V} \end{cases}$
- Results (1638)
 - Automated teaching experiments using GridWorlds (a) of varying sizes, (b) in varying contexts $\mathcal{V}$, and (c) at varying levels of state abstraction, where (d) teacher feedback was modeled as one of seven mixtures of demonstration and scalar evaluation
 - An agent using the likelihood-based algorithm can learn context-dependent norms from varying combinations of demonstration and evaluative feedback
- Discussion and conclusion (1638-40)

Table 46: First dimension

	Ethical theory	Categorization
Ethical theory type	Deontological ethics	
	Consequentialism	
	Virtue ethics	
	Particularism	
	Hybrid	Hierarchically specific
		Hierarchically nonspecific
	Configurable ethics	
Conflict sensitivity	Ambiguous	✓(folk morality)
	B/w AMAs	
	Within AMA	
	None	✓
Source	Developer	
	User	
	Expert	
	Lay people	✓(future implementation plan)
	(Legal) code/guidelines	
	Stakeholders	
Moral patient sensitivity	None	
	Yes	✓(future implementation plan)
	No	
	Unclear	

Table 47: Second dimension

Non-technical aspect		Categorization
Approach	Top-down	
	Bottom-up	✓
	Hybrid	
Contribution type	Model representation	
	Model selection	
	Judgment provision	
	Action selection/execution	✓
Evaluation	Test	Non-expert
		Expert
	Prove	Laws
		Model checker
		Logical proof
	Informal	Example scenario ✓
		Face validity
	None	
Domain-specific	yes/no	✗

Table 48: Third dimension

Technical aspect		Categorization
Tech type	Logical reasoning	Deductive logic
		Non-monotonic logic
		Abductive logic
		Deontic logic
		Rule-based system
		Event calculus
		Knowledge representation and ontologies
		Inductive logic
	Probabilistic reasoning	Bayesian approach
		Markov models
		Statistical inference
	Learning	Inductive logic
		Decision tree
		Reinforcement learning ✓
		Neural networks
		Evolutionary computing
	Optimization	
	Case-based reasoning	
Input	Case	Logical representation
		Numerical representation ✓
		(Structured) language representation
Implementation availability	Sensor data	
	Specification details	
	Implementation details	
	Code (link) provided	
Other		
		https://github.com/sbwncpodoe/ijcai
Other	Hardware	
	Feedback	
	Explanation	
	User interface	
	Automated processing	

17 Shea-Blymyer and Abbas (2021a)

Key Points 17

- Introduction
 - Goal: formalizing ethical obligations in a useful and interpretable way, analyzing the properties of these obligations, and automatically verifying that a system has given obligations
- Dominance act utilitarianism (DAU)
 - Syntax: $A := \phi | \neg A | A \wedge A | [\alpha cstit : A] | [\alpha dstit : A] | \odot [\alpha cstit : A] | \odot ([\alpha cstit : A] / \phi) | XA$
 - * $\alpha \in Agents$
 - * ϕ is a formula in CTL*, in which temporal operator $\Box$ means Always, $\Diamond$ means Eventually, and $\mathcal{R}$ means Release ($\phi \mathcal{R} \psi$ means that if ϕ never becomes true, ψ always holds, or ψ remains true until and once ϕ becomes true); path quantifier $\forall$ means for all paths and $\exists$ means that there exists a path
 - * DAU-specific operators: (i) $[\alpha cstit : A]$: α sees to it that A is true; (ii) $[\alpha dstit : A]$: α deliberately sees to it that A is true; (iii) $\odot [\alpha cstit : A]$: α ought to ensure that A is true; (iv) $\odot ([\alpha cstit : A] / \phi)$: under condition ϕ , α ought to ensure that A is true
 - Branching time
 - * *Tree*: set of moments with an irreflexive, transitive ordering relation $<$, a unique root moment denoted by 0
 - * History: maximal linearly ordered set of moments from *Tree*
 - * Given a moment $m \in Tree$, the set of histories that go through m is $H_m := \{h | m \in h\}$
 - * Moment/history pairs: m/h , where $m \in Tree$ and $h \in H_m$
 - * Branching time model: With AP a set of atomic propositions, a branching time model is a tuple $\mathcal{M} = (Tree, <, v)$, where *Tree* is a tree of moments with ordering $<$ and v is a function that maps moments m in $\mathcal{M}$ to sets of atomic propositions from 2^{AP} , the set of subsets of AP
 - * $\mathcal{M}, m/h \models \phi$: ϕ holds at pair m/h , where it is always the case that $h \in H_m$
 - * $|A|_m^{\mathcal{M}} := \{h \in H_m | \mathcal{M}, m/h \models A\}$: the proposition DAU statement A expresses at moment m is the set of histories where it holds starting at m ($\mathcal{M}$ sometimes dropped from notation)
 - Choice
 - * At every moment m , agent α is faced with a choice of actions $Choice_{\alpha}^m$
 - * At m , an action K is a subset of H_m , i.e., the subset of histories that are still realizable after taking the action
 - * Thus, $Choice_{\alpha}^m \subset 2^{H_m}$
 - Agency
 - * Chellas cstit: $\mathcal{M}, m/h \models [\alpha cstit : A]$ iff $Choice_{\alpha}^m(h) \subseteq |A|_m^{\mathcal{M}}$, where $Choice_{\alpha}^m(h)$ is the unique action that contains h , A is a DAU statement, and α an agent
 - * Action K guarantees A if $K \subseteq |A|_m$
 - * Deliberative stit: $\mathcal{M}, m/h \models [\alpha dstit : A]$ iff $Choice_{\alpha}^m(h) \subseteq |A|_m^{\mathcal{M}}$ and $|A|_m^{\mathcal{M}} \neq H_m$
 - Optimal actions
 - * Value function $Value : H_0 \rightarrow \mathbb{R}$ maps histories of $\mathcal{M}$ to utility values from $\mathbb{R}$
 - * Two sets of histories Z and Y are ordered as follows: $Z \leq Y$ iff $Value(h) \leq Value(h') \forall h \in Z, h' \in Y$
 - * $State_{\alpha}^m := Choice_{Agents \setminus \{\alpha\}}^m$: set of background states against which α 's decisions are to be evaluated (i.e., the choices available to other agents)
 - * Given two action K and K' in $Choice_{\alpha}^m$, $K \preceq K'$ iff $K \cap S \leq K' \cap S \forall S \in State_{\alpha}^m$ (i.e., K dominates K' iff it is preferable regardless of what the other agents do)

- * Optimal actions: $Optimal_\alpha^m := \{K \in Choice_\alpha^m \mid \nexists K' \in Choice_\alpha^m \text{ s.t. } K \prec K'\}$
- Dominance ought: With α an agent and A an obligation in model $\mathcal{M}$, $\mathcal{M}, m/h \models \odot[\alpha \text{ cstit} : A]$ iff $K \subseteq |A|_m^\mathcal{M} \forall K \in Optimal_\alpha^m$
- Permission: $P[\alpha \text{ cstit} : A] := \neg \odot[\alpha \text{ cstit} : \neg A]$ (i.e., α can ensure A without violating any obligations)
- Conditional obligation
 - * Where X is a proposition (i.e., $X = |\phi|_m$ for some ϕ), the choice available to α at m under the condition that X holds is defined as $Choice_\alpha^m \setminus X := \{K \in Choice_\alpha^m \mid K \cap X \neq \emptyset\}$
 - * Conditional dominance is defined by comparing only histories that satisfy ϕ : for two actions K, K' from $Choice_\alpha^m$, $K \preceq_X K'$ iff $K \cap S \cap X \leq K' \cap S \cap X \forall S \in State_\alpha^m$
 - * Conditionally optimal actions: $Optimal_\alpha^m \setminus X := \{K \in Choice_\alpha^m \setminus X \mid \nexists K' \in Choice_\alpha^m \setminus X \text{ s.t. } K \prec_X K'\}$
 - * Conditional ought: Where A is an obligation and ϕ a formula, $\mathcal{M}, m/h \models \odot([\alpha \text{ cstit} : A] \setminus \phi)$ iff $K \subseteq |A|_m^\mathcal{M} \forall K \in Optimal_\alpha^m \setminus |\phi|_m^\mathcal{M}$
- Case study in modeling: RSS for self-driving cars
 - RSS is a proposal put forth by Intel's Mobileye division
 - RSS rules place an obligation to avoid aggressive driving while giving permission to drive assertively
- Obligation propagation
 - Obligations vary over time: the obligation at moment m is the set of necessary conditions satisfied by all histories optimal at m , and the set of optimal histories can change from moment to moment
 - To derive valid temporal propagation patterns, it must be assumed that the agent is acting optimally, for which the proposition a^* is defined: $m/h \models a^*$ iff $Choice_\alpha^m(h) \in Optimal_\alpha^m$
 - First possible, valid propagation pattern (where the agent always acts optimally): $\odot[\alpha \text{ cstit} : X\phi] \wedge a^* \implies X \odot [\alpha \text{ cstit} : \phi]$
 - Second possible, valid propagation pattern (where α ought to see to it that $\Diamond\phi$ at the next moment when it is impossible to satisfy ϕ now): $\odot[\alpha \text{ cstit} : \Diamond\phi] \wedge a^* \wedge \forall \neg\phi \implies X \odot [\alpha \text{ cstit} : \Diamond\phi]$
 - Third possible propagation pattern (where optimal behavior is not required, but which is only satisfied in certain models): With ϕ, ψ CTL^* formulas, let $\mathcal{M}$ be a stit model that satisfies the following constraint at every moment m : for all action $K \in Choice_\alpha^m$ s.t. $K \subseteq |\neg\phi|_m$ and that contain a history h s.t. $m^+(h) \models \exists[\alpha \text{ cstit} : \psi]$ (where $m^+(h)$ is the moment that follows m in h), it holds that all optimal action in $Optimal_\alpha^{m^+(h)}$ guarantee ψ . Then, in such a model, the following is satisfied at every index m/h : $\odot[\alpha \text{ cstit} : \phi \vee X\psi] \wedge [\alpha \text{ cstit} : \neg\phi] \wedge X\exists[\alpha \text{ cstit} : \psi] \implies X \odot [\alpha \text{ cstit} : \psi]$
 - Fourth possible, valid propagation pattern (satisfied in all models): $\odot[\alpha \text{ cstit} : \phi \vee X\psi] \wedge [\alpha \text{ cstit} : \neg\phi] \wedge a^* \implies X \odot [\alpha \text{ cstit} : \psi]$
- Model checking DAU
 - The following algorithms verify that a system has a given obligation, i.e., a given constraint on its behavior
 - Agents are cast as stit automata
 - Stit automata
 - * Stit automaton: tuple $T = (Q, q_1, \mathcal{K}, F, \Delta, L, w, \lambda)$
 - Q is a finite, non-empty set of states
 - q_1 is the initial state (note: probably should say q_0 based on the notation used subsequently)
 - $\mathcal{K}$ is a finite non-empty set of actions

- $F \subset Q$ is a set of final states
- $\Delta \subset Q \times \mathcal{K} \times Q$ is a finite transition relation such that if (q, K, q') and (q, K', q') are in Δ then $K = K'$
- $L : Q \rightarrow 2^{AP}$ is a labeling function (where AP is a finite set of atomic propositions)
- $w : \Delta \rightarrow \mathbb{R}$ is a weight function
- $\lambda : \mathbb{R}^\omega \rightarrow \mathbb{R}$ is an accumulation function (where, for a set S , S^ω denotes the set of infinite sequences $(a_i)_{i \in \mathbb{N}}$ with $a_i \in S$)
- * Execution
 - q_0 -rooted execution ξ of T : sequence of transitions of the form

$$\xi = (q_0, K_0, q_1)(q_1, K_1, q_2) \dots \in \Delta^\omega$$
 , where T is a stit automaton and $q_0 \in Q$
 - Tactic: corresponding sequence of actions $K_0, K_1, \dots \in \mathcal{K}^\omega$
 - The value of execution $\xi = \xi[0]\xi[1]\xi[2]\dots$, where $\xi[i] \in \Delta$, is defined to be

$$\lambda(w(\xi[0])w(\xi[1])w(\xi[2])\dots)$$
 , and abbreviated $\lambda(\xi)$
- * A set of agents is modeled by the product of all individual stit automata, which is itself a stit automaton
- * From stit automata to stit models: an automaton T along with a state $q_0 \in Q$ induces a stit model $\mathcal{M}_{T,q_0}$ as follows (informal description)
 - q_0 maps to the root moment 0 of $\mathcal{M}_{T,q_0}$
 - From q_0 , T has a choice of actions $\mathcal{K}(q_0)$ which map to the actions available at 0 in $\mathcal{M}_{T,q_0}$
 - Each action K in $\mathcal{K}(q_0)$ non-deterministically causes one or more transitions, each of which maps to a moment in $\mathcal{M}_{T,q_0}$, and so on from each next state
 - $\eta : \Delta \rightarrow Tree$ is the map from transitions to moments (for executions: $\eta(\xi) := \eta(\xi[0])\eta(\xi[1])\dots$)
 - $\eta(\xi)$ is a history in $\mathcal{M}_{T,q_0}$, whose utility $Value(\eta(\xi))$ is the λ -value of the generating execution $\lambda(\xi)$
 - Atoms labeling $\eta((q, K, q'))$ are $v(\eta((q, K, q')))) := L(q')$
- * $\mathcal{M}_{T,q_0}$ is a utilitarian stit model with finite $Choice_\alpha^m$ for every agent α and moment m
- * Given an automaton T , one of its states q , the induced model $\mathcal{M}_{T,q}$, and an obligation A , T, q satisfies $\odot[\alpha \text{ cstit} : A]$ ($T, q \models \odot[\alpha \text{ cstit} : A]$) iff $\mathcal{M}_{T,q}, 0 \setminus h \models \odot[\alpha \text{ cstit} : A]$ for an arbitrary history $h \in H_0$
- * For certain accumulation functions, the choice of root moment does not matter
- Model checking for unconditional obligations
 - * Problem of cstit model checking: Given a stit automaton T that models agent α , a state $q \in T.Q$, and a formula A (either CTL^* formula or statement of the form $[\alpha \text{ dstit} : \phi]$ or $\neg[\alpha \text{ dstit} : \phi]$, where ϕ is a CTL^* formula), determine whether $\mathcal{M}_{T,q}, 0 \setminus h \models \odot[\alpha \text{ cstit} : A]$ for some arbitrary $h \in H_0$
 - * Problem can be broken down into two parts: (1) what is the set of optimal actions at H_0 ; and (2) do all of these optimal actions guarantee the truth of A ? If all optimal actions guarantee A , then $\mathcal{M}_T$ has obligation A at $0 \setminus h$
 - * Algorithm 1: returns True iff $\mathcal{M}, root/h \models \odot[\alpha \text{ cstit} : A]$ (*pseudo-code available in original paper*)
 - * Finding the maximum-valued and minimum-valued executions of an automation T'_n : e.g., $\lambda = \min$ or discounted sum function

- Model checking obligations
 - * Problem of condition cstit model checking: Given a stit automaton T that models an agent α , a state $q \in T.Q$, and a formula A , and a finite-horizon formula B (which means that there exists a $\tau \geq 0$ such that every history of length τ either satisfies or violates B), determine whether $\mathcal{M}_{T,q}, 0/h \models \odot([\alpha \text{ cstit} : A] \setminus B)$ for some $h \in H_0$.
 - * Algorithm 2: returns True iff $\mathcal{M}, \text{root}/h \models \odot([\alpha \text{ cstit} : A] \setminus B)$ (*pseudo-code available in original paper*)
 - * Algorithm 3 shows how to incrementally construct automata $T'_{n,l}$ (used in Algorithm 2): every such automaton has one initial action K_n , has a single execution up to the horizon τ (which satisfies B), and behaves like the original automaton after τ (*pseudo-code available in original paper*)
- Case study in model checking self-driving cars obligations
- Related work
- Conclusion

Table 49: First dimension

	Ethical theory	Categorization
Ethical theory type	Deontological ethics	
	Consequentialism	✓
	Virtue ethics	
	Particularism	
	Hybrid	Hierarchically specific
		Hierarchically nonspecific
	Configurable ethics	
	Ambiguous	
Conflict sensitivity	B/w AMAs	
	Within AMA	
	None	✓
Source	Developer	
	User	
	Expert	
	Lay people	
	(Legal) code/guidelines	✓
	Stakeholders	
	None	
Moral patient sensitivity	Yes	
	No	✓
	Unclear	

Table 50: Second dimension

Non-technical aspect		Categorization
Approach	Top-down	✓
	Bottom-up	
	Hybrid	
Contribution type	Model representation	
	Model selection	
	Judgment provision	
	Action selection/execution	✓
Evaluation	Test	Non-expert
		Expert
	Prove	Laws
		Model checker
		Logical proof
	Informal	Example scenario ✓
		Face validity
	None	
Domain-specific	yes/no	✗

Table 51: Third dimension

Technical aspect		Categorization
Tech type	Logical reasoning	Deductive logic ✓
		Non-monotonic logic
		Abductive logic
		Deontic logic
		Rule-based system
		Event calculus
		Knowledge representation and ontologies
		Inductive logic
	Probabilistic reasoning	Bayesian approach
		Markov models ✓
		Statistical inference
	Learning	Inductive logic
		Decision tree
		Reinforcement learning
		Neural networks
		Evolutionary computing
	Optimization	
	Case-based reasoning	
Input	Case	Logical representation ✓
		Numerical representation
		(Structured) language representation
Implementation availability	Sensor data	
	Specification details	
	Implementation details	
	Code (link) provided	
Other		
		https://github.com/sabotagelab/MC-DAU
Other	Hardware	
	Feedback	
	Explanation	
	User interface	
	Automated processing	

18 Shea-Blymyer and Abbas (2021b)

Key Points 18

- Introduction (15-6)
 - How can we tell which obligations should constrain the robot, so we can design it to have and meet these obligations?
 - Pair-wise comparisons of two behaviors of an AV are displayed to stakeholders, which indicate which of the two behaviors they prefer
 - These pair-wise comparisons are used to assign a score to each behavior such that if behaviors are ordered by score, the population's preferences are approximated
 - From the scores, weights on the robot's actions are learned
 - From the weights, policies and DAU formulas can be derived
 - Paper presents an algorithm that infers automaton weights from preferences over that automaton's behavior
- System model and formalized obligations (16)
 - System model
 - * Stit automaton: tuple $T = (Q, q_1, \mathcal{K}, F, \Delta, L, w, \lambda)$
 - Q is a finite, non-empty set of states
 - q_1 is the initial state (note: probably should say q_0 based on the notation used subsequently)
 - $\mathcal{K}$ is a finite non-empty set of actions
 - $F \subset Q$ is a set of final states
 - $\Delta \subset Q \times \mathcal{K} \times Q$ is a finite transition relation such that if (q, K, q') and (q, K', q') are in Δ then $K = K'$
 - $L : Q \rightarrow 2^{AP}$ is a labeling function (where AP is a finite set of atomic propositions)
 - $w : \Delta \rightarrow \mathbb{R}$ is a weight function
 - $\lambda : \mathbb{R}^\omega \rightarrow \mathbb{R}$ is an accumulation function (where, for a set S , S^ω denotes the set of infinite sequences $(a_i)_{i \in \mathbb{N}}$ with $a_i \in S$)
 - * Formulas of temporal logic can be interpreted over the non-deterministic executions of the automaton
 - * Set of actions available at q : $\mathcal{K}(q) = \{K \in \mathcal{K} \mid \exists (q, K, q') \in \Delta\}$
 - * q_0 -rooted execution: sequence of transitions of the form $\tau = (q_0, K_0, q_1)(q_1, K_1, q_2) \dots \in \Delta^\omega$
 - * Value of execution $\tau = \tau[0]\tau[1]\tau[2] \dots$, where $\tau[i] \in \Delta$, is defined to be

$$\lambda(w(\tau[0])w(\tau[1])w(\tau[2]) \dots)$$

, and abbreviated $\lambda(\tau)$
 - The specification language: Dominance Act Utilitarianism (DAU)
 - * Optimal action: Where Ξ_q denotes the set of executions starting with state q , and those executions that begin with action $K \in \mathcal{K}(q)$ are denoted $\Xi_q^K = \{\tau \mid \tau[0] = (q, K, q')\}$, an action $K^* \in \mathcal{K}(q)$ is optimal if there is no other action $K' \in \mathcal{K}(q)$ s.t. $\lambda(\tau') > \lambda(\tau)$ for all $\tau \in \Xi_q^{K^*}$ and $\tau' \in \Xi_q^{K'}$
 - * An execution τ either satisfies or violates A (a formula in the underlying temporal logic)
 - * Basic obligation modality: $\odot[\alpha \text{ cstit} : A]$ (robot α ought to see to it that A is true)
 - * $\odot[\alpha \text{ cstit} : A]$ holds true in q iff for all optimal actions K^* at q , and for all executions $\tau \in \Xi_q^{K^*}$, τ satisfies A
- Learning obligations from pairwise comparisons (16-8)

- Learning automaton weights
 - * Automata generation: system automata are generated randomly via the Erdos-Renyi model [12]
 - * Generating preference data
 - Given a stit automaton T , n random executions τ_i of T are generated
 - for each pair τ_i, τ_j , $i \neq j$, for each member p of the population P , values for the latent scores $s_p(\tau_i) \sim \mathcal{D}_i$ and $s_p(\tau_j) \sim \mathcal{D}_j$ (where the mean of $\mathcal{D}_i$ is $\lambda(\tau_i)$)
 - $R^{n \times n}$ is the response matrix, where $r_{ij} = \sum_{p \in P} \begin{cases} 1 & \text{if } s_p(\tau_i) > s_p(\tau_j) \\ 0 & \text{else} \end{cases}$ (i.e., the number of comparisons where τ_i is preferred over τ_j) (can also be thought of as a weighted directed graph where an edge from τ_i to τ_j has weight r_{ij})
 - * Weights from scores: weights $w \in T$ from scores, depend on accumulation function $\lambda \in T$ and executions τ (e.g., when λ is DiscSum, the weights can be estimated as follows: $\min_{\vec{w}} \sum_{i=0}^n (\vec{a}_i^T \vec{w} - s(\tau_i))^2$, where $A^{n,m}$ is a matrix with $m = |\Delta|$ and for each transition $\tau_i[k] \in \tau_i$ γ^k is added to a_{ij} , where j is the index of transition $\tau_i[k]$ in Δ)
 - * Measuring the distance between formulas: Where $\pi[i]$ is the set of actions optimal actions at state q (i.e., a policy), the distance between two policies π_1, π_2 is $D(\pi_1, \pi_2) = \|\vec{d}(\pi_1, \pi_2)\|_2$, where $d_i(\pi_1, \pi_2) = \frac{|\pi_1[i] \cap \pi_2[i]| - |\pi_1[i] \cup \pi_2[i]|}{|\pi_1[i] \cap \pi_2[i]|}$
- The four scoring methods for turning pairwise comparisons of alternatives into scores for each alternative
 - * Bradley-Terry
 - * Borda
 - * Deterministic depth-based scoring
 - * Stochastic depth-based scoring
- Comparing policies: baseline policy (value iteration applied to the true weights of T) vs. ideal (no-comparison) method (minimization of DiscSum equation)
- Theoretical sensitivity analysis: Setting theoretical bounds as to how much can the learned weights change when there is an opinion change in the response
- Experimental results
 - Policy sensitivity to changes in preferences (*figure available in original paper*)
 - Robustness to automaton connectedness (*figure available in original paper*)
 - Robustness to automaton connectedness (with greater standard deviation of sampling distribution) (*figure available in original paper*)
 - Robustness to population size (*figure available in original paper*)
 - Robustness to number of behaviors (*figure available in original paper*)

Table 52: First dimension

	Ethical theory	Categorization
Ethical theory type	Deontological ethics	
	Consequentialism	✓
	Virtue ethics	
	Particularism	
	Hybrid	Hierarchically specific
		Hierarchically nonspecific
	Configurable ethics	
	Ambiguous	✓ (folk morality)
Conflict sensitivity	B/w AMAs	
	Within AMA	
	None	✓
Source	Developer	
	User	
	Expert	
	Lay people	
	(Legal) code/guidelines	
	Stakeholders	✓
	None	
Moral patient sensitivity	Yes	✓
	No	
	Unclear	

Table 53: Second dimension

Non-technical aspect		Categorization
Approach	Top-down	
	Bottom-up	✓
	Hybrid	
Contribution type	Model representation	
	Model selection	
	Judgment provision	
	Action selection/execution	✓
Evaluation	Test	Non-expert
		Expert
	Prove	Laws
		Model checker
		Logical proof
	Informal	Example scenario
		Face validity
	None	✓
Domain-specific	yes/no	✗

Table 54: Third dimension

Technical aspect		Categorization
Tech type	Logical reasoning	Deductive logic
		✓
		Non-monotonic logic
		Abductive logic
		Deontic logic
		Rule-based system
		Event calculus
		Knowledge representation and ontologies
	Probabilistic reasoning	Inductive logic
		Bayesian approach
		Markov models
	Learning	Statistical inference
		Inductive logic
		Decision tree
		Reinforcement learning
		Neural networks
Input	Case	Evolutionary computing
		Optimization
		✓
	Case-based reasoning	
Implementation availability	Case	Logical representation
		✓
		Numerical representation
		✓
Other	(Structured) language representation	
	Sensor data	
	Specification details	✓
		Implementation details
		Code (link) provided
Other	Hardware	
	Feedback	
	Explanation	
	User interface	
	Automated processing	

19 Tassella *et al.* (2023)

Key Points 19

- Introduction
 - Paper suggests some ways in which AI systems could be used as tools to enhance a user’s moral decision-making ability
- Moral dilemmas
 - (Onto)logical dilemmas: In moral dilemmas, agents ought to do both A and B , but they cannot achieve both. In (onto)logical/symmetrical dilemmas, the agent ought to do both A and B for the same reason.
 - Ethical dilemmas: An agent has to take a stance about irreconcilable ethical frameworks (and it is unclear how to prioritize different frameworks).
 - Epistemic dilemmas: An agent’s decision-making is psychologically bounded due to either (1) a limited ability to consider the whole context or (2) an inability to meaningfully predict consequences of actions. This, in turn, hinders settling moral dilemmas.
- Artificial moral advisors
 - Moral observer
 - * Able to identify moral dilemmas
 - * MORL algorithm could learn the interest (i.e., the gained advantage of taking an action) for each of the moral values in each situation through a Q -function ($Q : \mathcal{S} \times \mathcal{A} \times \mathcal{M} \rightarrow \mathbb{R}$, where $\mathcal{S} = \{s_1, s_2, \dots\}$ is the set of states, $\mathcal{A} = \{a_1, a_2, \dots\}$ is the set of actions, and $\mathcal{M} = \{m_1, m_2, \dots\}$ is the set of moral values)
 - * Theoretical interests Q^{th} : interests an action would have if it always received the maximum possible reward
 - * Ethical threshold: vector $\zeta \in \mathbb{Z} = [0, 1]^{|\mathcal{M}|}$, where $|\mathcal{M}|$ is the number of moral values
 - * Acceptable action: action a is acceptable in state s with respect to ethical thresholds ζ if the action’s interests compared to its theoretical interests are higher than the thresholds on each dimension (i.e., $acceptable(s, a, \zeta) \iff \forall m \in \mathcal{M} \frac{Q(s, a, m)}{Q^{th}(s, a, m)} \geq \zeta_m$)
 - * Dilemma: situation s is a dilemma with respect to ethical thresholds ζ if, for all possible action $a \in \mathcal{A}$, none of them is acceptable with respect to ζ (i.e., $dilemma(s, \zeta) \iff \nexists a \in \mathcal{A} \text{ s.t. } acceptable(s, a, \zeta)$)
 - Moral organizer
 - * Goal to present a dilemma, where a user interface should describe current state s and available actions as well as their consequences once a dilemma has been identified by a MORL algorithm
 - Moral expert
 - * Able to settle a dilemma by (1) asking the user to choose which action they prefer, or by (2) preemptively asking the user for their preferences and using them automatically to find a suitable action, or by (3) automatically deriving user preferences from user data
 - * Able to reflect on ethical preferences before and after a dilemma has been settled

Table 55: First dimension

	Ethical theory	Categorization
Ethical theory type	Deontological ethics	
	Consequentialism	✓
	Virtue ethics	
	Particularism	
	Hybrid	Hierarchically specific
		Hierarchically nonspecific
	Configurable ethics	
	Ambiguous	
Conflict sensitivity	B/w AMAs	
	Within AMA	
	None	✓(only conflict between available actions and actions deemed acceptable by the user)
Source	Developer	
	User	✓
	Expert	
	Lay people	
	(Legal) code/guidelines	
	Stakeholders	
	None	
Moral patient sensitivity	Yes	✓
	No	
	Unclear	

Table 56: Second dimension

Non-technical aspect		Categorization
Approach	Top-down	
	Bottom-up	✓
	Hybrid	
Contribution type	Model representation	
	Model selection	
	Judgment provision	
	Action selection/execution	✓
Evaluation	Test	Non-expert
		Expert
	Prove	Laws
		Model checker
		Logical proof
	Informal	Example scenario
		Face validity
	None	✓
Domain-specific	yes/no	✗

Table 57: Third dimension

Technical aspect		Categorization
Tech type	Logical reasoning	Deductive logic
		Non-monotonic logic
		Abductive logic
		Deontic logic
		Rule-based system
		Event calculus
		Knowledge representation and ontologies
		Inductive logic
	Probabilistic reasoning	Bayesian approach
		Markov models
		Statistical inference
	Learning	Inductive logic
		Decision tree
		Reinforcement learning ✓
		Neural networks
		Evolutionary computing
	Optimization	
	Case-based reasoning	
Input	Case	Logical representation
		Numerical representation ✓
		(Structured) language representation
Implementation availability	Sensor data	
	Specification details	✓
	Implementation details	
	Code (link) provided	
Other	Hardware	✗
	Feedback	✗
	Explanation	✗
	User interface	✗
	Automated processing	✗

20 Agarwal *et al.* (2021)

Key Points 20

- Introduction
 - Important that GNNs are not perpetrating undesirable discriminatory biases (i.e., they are fair), and that they are robust to attacks resulting from small perturbations to the graph structure and node attributes (i.e., they are stable)
- Related work
- Preliminaries
 - Notation
 - * Undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{X})$, where $\mathbf{X}$ is the set of node attribute vectors corresponding to all nodes in $\mathcal{V}$
 - * Adjacency matrix $\mathbf{A} \in \mathbb{R}^{N \times N}$, where $N = |\mathcal{V}|$
 - * Incidence vector $\mathbf{I}_u \in \{0, 1\}^N$ captures all edges incident on node u
 - * Concatenation of node attribute vector and incidence vector for node u : $\mathbf{b}_u = [\mathbf{x}_u; \mathbf{I}_u]$
 - * Augmented graph $\mathcal{G}' = (\mathcal{V}, \mathcal{E}', \tilde{\mathbf{X}})$
 - * GNN with K layers, where $\mathbf{h}_u^1, \mathbf{h}_u^2, \dots, \mathbf{h}_u^K$ is the representations output by each layer for node u (notation: $\mathbf{z}_u = \mathbf{h}_u^K$)
 - * Similarity between two node representations given by distance metric $D : \mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}$
 - * Goal: learn an encoder function ENC which maps a given node u to a representation $\mathbf{z}_u$, i.e., $\text{ENC}(u) = \mathbf{z}_u$
 - * Downstream classifier f that maps node representation $\mathbf{z}_u$ to class label $\hat{y}_u$
 - Graph neural networks
 - * Many GNNs can be formulated as message passing networks specified by trainable operators MSG, AGG, and UPD, which are recursively applied on $\mathcal{G}$, specifying how neural messages are exchanged between nodes, aggregated, and transformed to arrive at final node representations in the last layer of transformations
 - * Message between a pair of nodes (u, v) in layer k : $\mathbf{m}_{uv}^k = \text{MSG}(\mathbf{h}_u^{k-1}, \mathbf{h}_v^{k-1})$
 - * Aggregation of messages from $\mathcal{N}_u$: $\mathbf{m}_u^k = \text{AGG}(\mathbf{m}_{uv}^k | u \in \mathcal{N}_u)$
 - * u 's representation for layer k : $\mathbf{h}_u^k = \text{UPD}(\mathbf{m}_u^k, \mathbf{h}_u^{k-1})$
 - Fairness and stability
 - * A function is counterfactually fair if its output is independent of the sensitive attribute
 - * Encoder function ENC satisfies counterfactual fairness if $\text{ENC}(u) = \text{ENC}(\tilde{u}^s)$ for any given node u , where $\tilde{u}^s$ is a node in the augmented graph which is generated by modifying/flipping the sensitive attribute s of node u while keeping everything else constant
 - * Stability via Lipschitz continuity: a function is stable if slightly perturbing any given instance does not drastically change its output
 - * Encoder function ENC is stable according to Lipschitz continuity if

$$\|\text{ENC}(\tilde{u}) - \text{ENC}(u)\|_p \leq L \|\tilde{\mathbf{b}}_u - \mathbf{b}_u\|_p$$

, where L is the Lipschitz constant

- Our framework NIFTY
 - Problem: Given a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{X})$, NIFTY aims to generate d -dimensional embeddings $\mathbf{z}_u \in \mathbb{R}^d$ that are counterfactually fair and stable to attribute and structural perturbations of $\mathcal{G}$
 - Enforcing fairness and stability in the objective function: overall objective function is given by $\min_{\theta_{\text{ENC}}, \theta_t, \theta_f} \mathbb{E}_u[(1 - \lambda)\mathcal{L}_c] + \lambda\mathcal{L}_s$, where $\theta_{\text{ENC}}, \theta_t, \theta_f$ are trainable parameters of ENC,

- predictor t , and classifier f ; $\mathcal{L}_c$ is the binary cross entropy (BCE) loss, the expectation is taken over training nodes in $\mathcal{G}$, and λ is a regularization coefficient
- Enforcing fairness and stability in GNN architecture
 - * NIFTY modifies the UPD step of each GNN layer, where a typical GNN layer is given by $\mathbf{h}_u^k = \text{UPD}(\text{AGG}(\text{MSG}(\mathbf{h}_u^{k-1}, \mathbf{h}_v^{k-1}) | v \in \mathcal{N}_u), \mathbf{h}_u^{k-1})$
 - * Considering the AGG operator to be a fully-connected layer and UPD to be a non-linear activation function σ , the message-passing step can be rewritten as $\mathbf{h}_u^k = \sigma(\mathbf{W}_a^k \mathbf{h}_u^{k-1} + \mathbf{W}_n^k \sum_{v \in \mathcal{N}(u)} \mathbf{h}_v^{k-1})$, where $\mathbf{W}_n^k$ is the weight matrix associated with the neighbors of node u at layer k and $\mathbf{W}_a^k$ is the self-attention weight matrix at layer k
 - * Lipschitz constant L is used to normalize $\tilde{\mathbf{W}}_a^k = \frac{\mathbf{W}_a^k}{\sigma(\mathbf{W}_a^k)}$
 - * Lipschitz-normalized weight matrix is used to modify the UPD step: $\mathbf{h}_u^k = \sigma(\tilde{\mathbf{W}}_a^k \mathbf{h}_u^{k-1} + \mathbf{W}_n^k \sum_{v \in \mathcal{N}(u)} \mathbf{h}_v^{k-1})$
 - NIFTY algorithm (*pseudo-code available in original paper*)
 - Theoretical analysis of NIFTY
 - Theorem 1: Given a non-linear activation function σ that is Lipschitz continuous, the representations learned by NIFTY are stable: $\|\text{ENC}(\tilde{u}) - \text{ENC}(u)\|_p \leq \prod_{k=1}^K \|\mathbf{W}_a^k\|_p \|\tilde{\mathbf{b}}_u - \mathbf{b}_u\|_p$
 - Theorem 2: Given a non-linear activation function σ that is Lipschitz continuous and a binary valued sensitive attribute s , the (counterfactual) unfairness of the representations learned by NIFTY can be bounded as follows: $\|\text{ENC}(\tilde{u}^s) - \text{ENC}(u)\|_p \leq \prod_{k=1}^K \|\mathbf{W}_a^k\|_p$
 - Proposition 1: If the representations learned by NIFTY satisfy counterfactual fairness, then a downstream classifier $f : \mathbf{z}_u \rightarrow \hat{y}_u$ which leverages these representations also satisfies counterfactual fairness
 - Experiments
 - Across three datasets and five GNNs, NIFTY-augmented GNNs learn fairer and more stable embeddings than unmodified GNNs
 - NIFTY indirectly improves group fairness of GNNs (even though the explicit aim is to capture counterfactual fairness)
 - As regularization λ increases, error rates for counterfactual fairness and stability steadily decrease
 - The objective function and the layer-wise normalization of GNN’s architecture using the Lipschitz constant are both necessary to generate embeddings that are simultaneously fair and stable

Table 58: First dimension

	Ethical theory	Categorization
Ethical theory type	Deontological ethics	
	Consequentialism	✓
	Virtue ethics	
	Particularism	
	Hybrid	Hierarchically specific
		Hierarchically nonspecific
	Configurable ethics	
	Ambiguous	
Conflict sensitivity	B/w AMAs	
	Within AMA	
	None	✓
Source	Developer	
	User	
	Expert	
	Lay people	
	(Legal) code/guidelines	
	Stakeholders	
	None	✓
Moral patient sensitivity	Yes	
	No	
	Unclear	✓

Table 59: Second dimension

Non-technical aspect		Categorization
Approach	Top-down	
	Bottom-up	✓
	Hybrid	
Contribution type	Model representation	
	Model selection	
	Judgment provision	
	Action selection/execution	✓
Evaluation	Test	Non-expert
		Expert
	Prove	Laws
		Model checker
		Logical proof ✓
	Informal	Example scenario ✓
		Face validity
	None	
Domain-specific	yes/no	✗

Table 60: Third dimension

Technical aspect		Categorization
Tech type	Logical reasoning	Deductive logic
		Non-monotonic logic
		Abductive logic
		Deontic logic
		Rule-based system
		Event calculus
		Knowledge representation and ontologies
		Inductive logic
	Probabilistic reasoning	Bayesian approach
		Markov models
		Statistical inference
	Learning	Inductive logic
		Decision tree
		Reinforcement learning
		Neural networks ✓
		Evolutionary computing
	Optimization	
	Case-based reasoning	
Input	Case	Logical representation
		Numerical representation ✓
		(Structured) language representation
Implementation availability	Sensor data	
	Specification details	
	Implementation details ✓	
	Code (link) provided	
Other	Hardware ✗	
	Feedback ✗	
	Explanation ✗	
	User interface ✗	
	Automated processing ✗	

Key Points 21

- Introduction
 - Fair classification problems: protected group defined by a shared sensitive attribute and one wishes not to be biased against any one group with the same sensitive attribute
 - Selective classification: a model is allowed to abstain from making a decision (classification/no classification decision usually done via a threshold applied to a prediction confidence measure)
 - Recent work: selective classification can magnify disparities between groups as the coverage decreases, even though overall performance increases
 - Contributions
 - * Prove that sufficiency can be used to train fairer selective classifier which ensure that precision always increases as coverage is decreased for all groups
 - * Derive a novel upper bound of the conditional mutual information which can be used as a regularizer to enforce the sufficiency criterion
- Background
 - The fair classification problem
 - * Supervised learning: predicting the value of target variable $Y \in \mathcal{Y}$ using a set of predictive variables $X \in \mathcal{X}$ with training samples $\{(x_1, y_1), \dots, (x_n, y_n)\}$, i.e., finding features $\Phi(x) \in \mathbb{R}^{d_\Phi}$ which are predictive about Y so that a predictor $\hat{y} = T(\Phi(x))$ of y can be constructed under some loss criteria $L(\hat{y}, y)$, where T is the final classifier on the learned features
 - * Positive predictive parity (PPV) is satisfied when the precision for each group is the same: $\forall a, b \in \mathcal{D}, \mathbb{P}(Y = 1 | \hat{Y} = 1, D = a) \mathbb{P}(Y = 1 | \hat{Y} = 1, D = b)$, where $D \in \mathcal{D}$ is a sensitive attribute, Y is a target variable, $\hat{Y}$ is a prediction
 - Selective classification
 - * Decision to make a prediction $\hat{Y}$ or not based on confidence score $\mathcal{K}(x)$ on a given sample x
 - * Confidence using softmax $s(x)$: $\mathcal{K} = \frac{1}{2} \log \left(\frac{s(x)}{1-s(x)} \right)$
 - * Coverage: fraction of samples for which a prediction is made
 - * Measuring the effectiveness of selective classification: plot accuracy-coverage curve and compute the area under the curve to capture the performance across different coverages
 - Biases in selective classification
 - * Margin M of a classifier: $M = \begin{cases} \mathcal{K}(x) & \text{if } \hat{y}(x) = y \\ -\mathcal{K}(x) & \text{otherwise} \end{cases}$
 - * A selective classifier makes the correct prediction when $M(x) \geq \tau$ and incorrect predictions when $M(x) \leq -\tau$, where τ is a threshold
 - * Probability density function (PDF) f_M and cumulative density function (CDF) F_M
 - * Selective accuracy: $A_F(\tau) = \frac{1 - F_M(\tau)}{F_M(-\tau) + 1 - F_M(\tau)}$
 - * Selective precision: $PPV_F(\tau) = \frac{1 - F_{M|\hat{Y}=1}(\tau)}{F_{M|\hat{Y}=1}(-\tau) + 1 - F_{M|\hat{Y}=1}(\tau)}$
- Fair selective classification with sufficiency
 - Sufficiency and fair selective classification
 - * When Y is binary, sufficiency requires that $\mathbb{P}(Y = 1 | \Phi(x), D = a) = \mathbb{P}(Y = 1 | \Phi(x), D = b), \forall a, b \in \mathcal{D}$
 - * Calibration by group requires that there exists a score function $R = s(x)$ such that for all $r \in (0, 1)$: $\mathbb{P}(Y = 1 | R = r, D = a) = r, \forall a \in \mathcal{D}$

- * Theorem 1: If a classifier has sufficient features Φ , then there exists a mapping $h(T(\Phi)) : [0, 1] \rightarrow [0, 1]$ such that $h(T(\Phi))$ is calibrated by group
- * Theorem 2: If a classifier has a score function $R = s(x)$ which is calibrated by group, and selective classification is performed using confidence $\mathcal{K}$, then for all groups $d \in \mathcal{D}$ both $A_F(\tau)$ and $PPV_{F_d}(\tau)$ are monotonically increasing with respect to τ ; and $A_{F_d}(0) > 0.5$ and $PPV_{F_d}(0) > 0.5$
- Imposing the sufficiency condition
 - * Sufficiency constraint using mutual information: $\min_{\theta} L(\hat{y}, y) + \lambda I(Y; D | \Phi(X))$
 - * Upper bound for mutual information: $I(Y; D | \Phi(X)) \leq \mathbb{E}_{P_{XYD}}[\log P(Y | \Phi(X), D)] - \mathbb{E}_{P_D}[\mathbb{E}_{P_{XY}}[\log P(Y | \Phi(X), D)]]$
 - * Since $P(x | \Phi(x), d)$ is unknown, it is approximated with parameter θ_d using a neural net for each group $d \in \mathcal{D}$ as $q(y | \Phi(x); \theta_d)$, leaving regularizer: $L_R \triangleq \frac{1}{n} \sum_{i=1}^n (\log q(y_i | \Phi(x_i); \theta_{d_i}) - \log q(y_i | \Phi(x_i); \theta_{\tilde{d}_i}))$, where $\tilde{d}_i$ are drawn independently and identically from marginal distribution P_D , and for $d \in \mathcal{D}$, $\theta_d = \arg \max_{\theta} \sum_{i: d_i=d} \log q(y_i | \Phi(x_i); \theta)$
 - * Overall loss function:

$$\min_{\theta_T, \theta_{\Phi}} \frac{1}{n} \sum_{i=1}^n (L(T(\Phi(x_i)), y_i) + \lambda \log q(y_i | \Phi(x_i); \theta_{d_i}) - \lambda \log q(y_i | \Phi(x_i); \theta_{\tilde{d}_i}))$$

where T is a joint classifier over all groups that is used to make final predictions such that $\hat{y} = T(\Phi(x))$

- * Idea: minimize the difference between the log-probability of the output of the correct model and the randomly-chosen one trying to enforce that $\Phi(x)$ to have the property that all group-specific models trained on it will be the same: $q(y | \Phi(x); \theta_a) = q(y | \Phi(x); \theta_b), \forall a, b \in \mathcal{D}$
- Training with sufficiency-based regularizer (*pseudo-code available in original paper*)
- Computation of sufficiency-based loss when D is binary (*figure available in original paper*)
- Experimental results
 - Four binary classification datasets: Adult, CelebA, Civil Comments, CheXpert
 - Comparisons: (a) baseline method optimizing cross-entropy loss; (b) group DRO method, which has been shown to mitigate the disparity in recall rates between groups in selective classification; (c) sufficiency-regularized (present paper)
 - Group-specific precision-coverage curves for datasets Adult and CheXpert (*figure available in original paper*)
 - Area under the curve results (*table available in original paper*)

Table 61: First dimension

	Ethical theory	Categorization
Ethical theory type	Deontological ethics	
	Consequentialism	✓
	Virtue ethics	
	Particularism	
	Hybrid	Hierarchically specific
		Hierarchically nonspecific
	Configurable ethics	
	Ambiguous	
Conflict sensitivity	B/w AMAs	
	Within AMA	
	None	✓
Source	Developer	
	User	
	Expert	
	Lay people	
	(Legal) code/guidelines	
	Stakeholders	
	None	✓
Moral patient sensitivity	Yes	
	No	
	Unclear	✓

Table 62: Second dimension

Non-technical aspect		Categorization
Approach	Top-down	
	Bottom-up	✓
	Hybrid	
Contribution type	Model representation	
	Model selection	
	Judgment provision	
	Action selection/execution	✓
Evaluation	Test	Non-expert
		Expert
	Prove	Laws
		Model checker
		Logical proof
		✓ (for upper bound)
	Informal	Example scenario
		✓
	None	Face validity
Domain-specific	yes/no	✗

Table 63: Third dimension

Technical aspect		Categorization
Tech type	Logical reasoning	Deductive logic
		Non-monotonic logic
		Abductive logic
		Deontic logic
		Rule-based system
		Event calculus
		Knowledge representation and ontologies
		Inductive logic
	Probabilistic reasoning	Bayesian approach
		Markov models
		Statistical inference
	Learning	Inductive logic
		Decision tree
		Reinforcement learning
		Neural networks ✓
		Evolutionary computing
	Optimization	
	Case-based reasoning	
Input	Case	Logical representation
		Numerical representation ✓
		(Structured) language representation
Implementation availability	Sensor data	
	Specification details	
	Implementation details ✓	
Other	Code (link) provided	
	Hardware ✗	
	Feedback ✗	
	Explanation ✗	
	User interface ✗	
	Automated processing ✗	

22 Ecoffet and Lehman (2021)

Key Points 22

- Introduction
 - When ethical reward functions (of different ethical theories) can be compared on a shared cardinal scale, multi-objective RL can be employed in a straightforward way
 - Multi-objective RL aims to solve the problem of finding the set of efficient trade-offs among competing objectives without addressing how to choose which trade-off policy to deploy
 - Solution proposed by the paper: an ethical theory should influence outcomes proportional to its credence, irrespective of the scale of its rewards
- Philosophical background
 - MacAskill (2014) explores how to handle the comparability of moral theories and how to combine preferences across ordinal and cardinal theories, i.e., whether a given theory's preferences assign a specific score to various options, or simply order them from best to worst
- Formalism of moral uncertainty
 - Choice-worthiness function W of an ethical theory: preference ordering over actions and their immediate outcomes across different states of the world (assumed to be complete, i.e., defined for all possible state-action pairs)
 - Modified version of the standard MDP framework, where, instead of a reward function $R(s, a, s')$, a choice-worthiness function $W_i(s, a, s')$ is used for theory i
 - A morally uncertain agent must find a compromise between maximizing multiple choice-worthiness functions (rather than a single reward function)
 - Expected discounted sum of future choice-worthiness for theory i starting from taking action a at state s with all future actions given by the current policy π : $Q_i(s, a) = \mathbf{E} \left[\sum_{t=0}^{\infty} \gamma^t W_i(s_t, a_t, s_{t+1}) | s_0 = s, a_0 = a \right]$, where $\gamma_i \in [0, 1]$ is a discount factor
 - Credence: each theory i has a level of credence C_i indicating the degree of belief the agent (or its designer) has in the theory (sum to 1 across theories for a given agent, and are assumed to be set and fixed)
- Methods
 - Maximizing expected choice-worthiness (MEC), i.e., $R(s, a, s') = \sum_i C_i W_i(s, a, s')$ is scale-sensitive
 - All theories that are comparable are first aggregated into a single virtual theory using MEC before handling the set of remaining incomparable theories
 - Voting systems
 - * Principle of proportional say: theories should have an influence that is proportional to their credence and not to the particular details of their choice-worthiness function, i.e., their scale
 - * Principle points to a voting systems
 - * Desirable properties of a voting system (Arrow's desirability axioms)
 - Non-dictatorship: outcome should not only reflect the preferences of a single theory
 - Pareto efficiency: if all theories prefer action A over action B at a given state, action B should not be chosen at that state
 - Independence of irrelevant alternatives (IIA): if, in a given state, A is chosen over B , then adding a new action C to the action set should not result in B being chosen instead
 - * Arrow's impossibility theorem: any deterministic voting system which satisfies Pareto

- efficiency and IIA must be a dictatorship (thus either Pareto efficiency or IIA is broken systematically when designing a deterministic voting system)
- Nash voting
 - * Principle of proportional say: Theories have proportional say if they are each allocated an equal voting budget and vote following the same cost structure, after which their votes are scaled proportionally to their credences
 - * Nash-voting: each theory tries to maximize its sum of choice-worthiness
 - At each time step, each theory provides a continuous-valued vote for or against each of the available actions
 - Action with the largest credence-weighted vote at each time step is executed
 - Cost of the theories' votes is subtracted from their remaining budget (if a theory overspends its remained budgets, votes are scaled to match the remaining budget, and it has 0 budget for all subsequent steps)
 - * Nash-voting can be thought of as a competitive, multi-agent setting, where each theory is treated as a sub-agent that influence controls of the singular object-level RL agent
 - * Theories can be trained under Nash voting using multi-agent RL, which aims towards convergence to a Nash equilibrium among competing theories
 - * Nash voting can exhibit two flaws
 - Stakes insensitivity: increasing the stakes for one theory and not the other does not increase the relative say of the theory for which more is at stake
 - No compromise: if an action is not any theory's most preferred action, it cannot be chosen, even if it is seemingly the best middle ground option
- Variance voting
 - * Forcing votes to be a true representation of theories' preferences (rather than being tactical) would alleviate the issues of stakes insensitivity and no compromise
 - * Preference of theory i for action a in state s given policy π is given as $Q_i(s, a)$
 - * Variance voting: preferences of theories are variance-normalized across decision options
 - * Expected value of the Q_i function's variance across time steps: $\sigma_i^2 = \mathbf{E}_{s \sim S} \left[\frac{1}{k} \sum_a (Q_i(s, a) - \mu_i(s))^2 \right]$, where $\mu_i(s) = \frac{1}{k} \sum_a Q_i(s, a)$ and S is the distribution of states the agent can encounter under the policy
 - * Policy, where the Q_i function is normalized by the expected value of its variance across time steps: $\pi(s) = \arg \max_a \sum_i C_i \frac{Q_i(s, a) - \mu_i(s)}{\sqrt{\sigma_i^2 + \epsilon}}$, where ϵ is a small constant to handle theories with $\sigma_i^2 = 0$
 - * Variance-SARSA: for learning, target y_i is used for theory i : $y_i = W_i(s, a, s') + \gamma_i Q_i(s', a')$, where a' is the action taken by policy π in state s' which is either produced by variance voting or an ϵ -greedy action (Variance-SARSA converges if σ_i^2 converges)
- Updating credences
 - * It is possible for the designer to update credences without retraining the policy by credence-conditioning the sub-policies of the theories and training them in simulation under a wide variety of credences
- Experiments
 - Four gridworld environments derived from the trolley dilemma
 - Nash voting and stakes insensitivity: Nash voting does not exhibit stake sensitivity in the classical trolley dilemma, but it can be stakes sensitive if it expects to make multiple decisions in sequence
 - Nash voting and no compromise: whichever theory has the upper hand in terms of credence is capable of imposing its preference, even in the presence of a compromise action

- Variance voting and IIA: adding irrelevant options can have a significant impact on the final outcome
- Related work
- Discussion
- Results

Table 64: First dimension

	Ethical theory	Categorization
Ethical theory type	Deontological ethics	
	Consequentialism	
	Virtue ethics	
	Particularism	
	Hybrid Hierarchically specific	✓(hierarchy defined by designer/stakeholders; D + C used in experiments)
	Hierarchically nonspecific	
	Configurable ethics	
	Ambiguous	
Conflict sensitivity	B/w AMAs	
	Within AMA	✓
	None	
Source	Developer	✓
	User	
	Expert	
	Lay people	
	(Legal) code/guidelines	
	Stakeholders	✓
	None	
Moral patient sensitivity	Yes	✓(credence levels specified by stakeholders)
	No	✓(credence levels specified by designer)
	Unclear	

Table 65: Second dimension

Non-technical aspect		Categorization
Approach	Top-down	
	Bottom-up	✓
	Hybrid	
Contribution type	Model representation	
	Model selection	
	Judgment provision	
	Action selection/execution	✓
Evaluation	Test	Non-expert
		Expert
	Prove	Laws
		Model checker
		Logical proof
	Informal	Example scenario ✓
		Face validity
	None	
Domain-specific	yes/no	✗

Table 66: Third dimension

Technical aspect		Categorization
Tech type	Logical reasoning	Deductive logic
		Non-monotonic logic
		Abductive logic
		Deontic logic
		Rule-based system
		Event calculus
		Knowledge representation and ontologies
		Inductive logic
	Probabilistic reasoning	Bayesian approach
		Markov models
		Statistical inference
	Learning	Inductive logic
		Decision tree
		Reinforcement learning ✓
		Neural networks
		Evolutionary computing
	Optimization	
	Case-based reasoning	
Input	Case	Logical representation
		Numerical representation ✓
		(Structured) language representation
Implementation availability	Sensor data	
	Specification details	
	Implementation details	
	Code (link) provided	
Other		
	Hardware	
	Feedback	
	Explanation	
	User interface	
	Automated processing	

23 Frisch *et al.* (2021)

Key Points 23

- Introduction (69)
 - Paper aims to produce fair recommendations using a co-clustering of users and items that respects statistical parity of users with respect to some sensitive attributes
- Model
 - Data used to build recommender systems can be aggregated in a matrix where rows are users, columns are items and entries are feedback
 - The Latent block models
 - * Generative probabilistic models that enable to cluster the rows and columns of a data matrix $\mathbf{R}$: assume a homogeneous block structure in data matrix $\mathbf{R}$ which is revealed by reordering rows and columns according to their respective cluster index
 - Model proposed (70-1)
 - * Users most often express their interest in an item using a discrete rating scale, which is why the method proposed is based on a statistical co-clustering using ordered probit regression (which allows to include sensitive attributes as covariates)
 - * Ordered probit model is used within a latent block model, where the former assumes the existence of a continuous, Gaussian distributed latent random variable $\mathbf{R}^*$ representing the underlying rating value
 - * The assumption of the latent block model that users in a given cluster share exactly the same opinion is relaxed
 - * Sensitive attribute s_i is a binary feature of user i that should not intervene in the recommendation of items
 - * Latent variable C_j for each object j interacts with different strength with the sensitive attribute, with the interaction being added to the conditional distribution of R_{ij}^*
 - * Assumption that missing ratings are missing at random or missing completely at random
- Inference and fair recommendations (71-3)
 - A stochastic batch gradient descent of the variational criterion
 - * Log-likelihood of the model is not traceable, which is why a variational inference procedure is used
 - * Latent stochastic optimization is used to maximize the variational criterion using noisy estimates of its gradient
 - Fair recommendations
 - * In collaborative filtering, usually recommendations are produced by estimating a relevance score for each item, which is then used to define a total order through numerical comparisons
 - * Fair comparison of items: given user i and items j and j' , the comparison of items j and j' is said to be fair if it is freed from the evaluation bias regarding the sensitive attribute s
 - * Clustering ϵ -parity: clustering of users respects ϵ -parity with respect to sensitive attribute s if sub-populations of users defined by identical sensitive attributes are represented approximately equally in each clusters
 - * ϵ -fair recommendation: a recommender system is ϵ -fair if for any two items j and j' , the proportion of users with the same preference is approximately identical in all the subpopulations of users defined by identical sensitive attributes
 - * Fair recommendation from clustering parity (theorem): If the clustering of users in k_1

groups respects ϵ -parity, then the recommender system relying on the relevance score introduced previously is $(k_1\epsilon)$ -fair

- Experiment on MovieLens dataset
 - Model (Parity LBM) is compared to (i) baseline LBM (Standard LBM), (ii) weighted Bregman co-clustering (Bregman co-clust), (iii) Singular Value Decomposition (SVD)
 - User gender as sensitive attribute
 - Comparison of the gap ϵ from exactly fair recommendations (*figure available in original paper*)
 - Ranking performance averaged over all users, for a recommendation list of 10 items (*figure available in original paper*)
- Conclusion

Table 67: First dimension

	Ethical theory	Categorization
Ethical theory type	Deontological ethics	
	Consequentialism	✓
	Virtue ethics	
	Particularism	
	Hybrid	Hierarchically specific
		Hierarchically nonspecific
	Configurable ethics	
	Ambiguous	
Conflict sensitivity	B/w AMAs	
	Within AMA	
	None	✓
Source	Developer	
	User	
	Expert	
	Lay people	
	(Legal) code/guidelines	
	Stakeholders	
	None	✓
Moral patient sensitivity	Yes	
	No	
	Unclear	✓

Table 68: Second dimension

Non-technical aspect		Categorization
Approach	Top-down	
	Bottom-up	✓
	Hybrid	
Contribution type	Model representation	
	Model selection	
	Judgment provision	
	Action selection/execution	✓
Evaluation	Test	Non-expert
		Expert
	Prove	Laws
		Model checker
		Logical proof ✓
	Informal	Example scenario ✓
		Face validity
	None	
Domain-specific	yes/no	✗

Table 69: Third dimension

Technical aspect		Categorization
Tech type	Logical reasoning	Deductive logic
		Non-monotonic logic
		Abductive logic
		Deontic logic
		Rule-based system
		Event calculus
		Knowledge representation and ontologies
		Inductive logic
	Probabilistic reasoning	Bayesian approach
		Markov models
		Statistical inference
	Learning	Inductive logic
		Decision tree
		Reinforcement learning
		Neural networks
		Evolutionary computing
	Optimization	✓(unsupervised learning)
	Case-based reasoning	
Input	Case	Logical representation
		Numerical representation
		(Structured) language representation
Implementation availability	Sensor data	
	Specification details	
	Implementation details	
	Code (link) provided	
Other	Hardware	
	Feedback	
	Explanation	
	User interface	
	Automated processing	

Key Points 24

- Introduction (1054-5)
 - Personalized and causal fairness for recommendation systems using counterfactual fairness
 - Related work (1055-6)
 - Preliminaries and notations (1056-7)
 - A recommendation system is counterfactually fair if for any possible user u with features $\mathbf{X} = \mathbf{x}$ and $\mathbf{Z} = \mathbf{z}$: $P(L_{\mathbf{z}}|\mathbf{X} = \mathbf{x}, \mathbf{Z} = \mathbf{z}) = P(L_{\mathbf{z}}|\mathbf{X} = \mathbf{x}, \mathbf{Z} = \mathbf{z}')$ for all L and for any value $\mathbf{z}'$ attainable by $\mathbf{Z}$, where $\mathbf{Z}$ is the set of sensitive features and $\mathbf{z}$ is the set of associated values, $\mathbf{X}$ is the set of remaining features (that are not causally dependent on $\mathbf{Z}$) and $\mathbf{x}$ is the set of associated values, and where L denotes the Top- N recommendation list for user u (i.e., for a given user u , the distribution of the generated recommendation results L should be the same if only $\mathbf{Z}$ is changed from $\mathbf{z}$ to $\mathbf{z}'$, while holding the remaining features $\mathbf{X}$ constant)
 - Fair recommendation framework (1057-9)
 - Problem formulation
 - * Given a user u , for any item $v \in \mathcal{V}$, the predicted score S_{uv} for user-item pair (u, v) is independent from user sensitive features $\mathbf{Z}$
 - * Given that a user u has selected a set of sensitive features $\mathbf{Q}_u \subseteq \{1, \dots, K\}$ that they care about, $\mathbf{r}_u \perp z_u^k$ needs to be guaranteed for all $k \in \mathbf{Q}_u$, where $\mathbf{r}_u$ is an embedding for u
 - The model
 - * Main idea: train a predictor and an adversarial classifier simultaneously, where the predictor aims at learning informative representations for the recommendation task, while the adversarial classifier aims at minimizing the predictor's ability to predict the protected features from the learned representation, meaning that information about sensitive features is removed from the representations to mitigate discrimination
 - * Adversary network consisting of two modules: (1) filter module which aims at filtering out the information about sensitive features from user embeddings, and (2) discriminator module which aims to predict the sensitive features from the learned user embeddings
 - * Filter module
 - Filter function $f : \mathbb{R}^d \mapsto \mathbb{R}^d$
 - Separate method (SM): the user selects a few sensitive features and a separate filter function is trained for each feature and each feature combination: filtered embeddings of user u in terms of the selected sensitive features $\mathbf{Q}_u$ are denoted $\mathbf{r}_u^* = f_{\mathbf{Q}_u}(\mathbf{r}_u)$
 - Combination method (CM): a filter function is trained for each sensitive feature: filtered embeddings are denoted
- $$\mathbf{r}_u^* = G(f_{q_1}(\mathbf{r}_u), f_{q_2}(\mathbf{r}_u), \dots, f_{q_{|\mathbf{Q}_u|}}(\mathbf{r}_u)),$$
- where $\mathbf{Q}_u = \{q_1, q_2, \dots, q_{|\mathbf{Q}_u|}\}$ and where G is a combination function that takes the $|\mathbf{Q}_u|$ filtered embeddings as input, and outputs the embedding which is independent from all features in $\mathbf{Q}_u$ without changing the embedding dimension
- * Discriminator module
 - For each sensitive feature Z_k a classifier $C_k : \mathbb{R}^d \mapsto [0, 1]$ is trained which attempts to predict Z_k from the user embeddings

- Goal of the filter functions is to make it hard for the classifiers to predict the sensitive features from the user embeddings, while the goal of the discriminators is to fail the filter functions
- The training process tries to jointly optimize both goals
- * Adversary training
 - Adversary learning loss: $\mathcal{L} = \sum_{u,v,\mathbf{Q}_u} \left(\mathcal{L}_{\text{Rec}}(u,v,\mathbf{Q}_u) - \lambda \sum_{k \in \mathbf{Q}_u} \sum_{z_k \in Z_k} \mathcal{L}_C(\mathbf{r}_u^*, z_k) \right)$, where $\mathcal{L}_{\text{Rec}}$ is the loss of the recommendation task, $\mathcal{L}_C$ is the loss of the discriminators (which is a cross-entropy loss), and λ controls the trade-off between recommendation performance and fairness
 - Theorem: If (1) the filter functions and discriminators are implemented with sufficient capacity, and (2) at each step of the following algorithm, the discriminators are allowed to reach their optimum given the filter functions, and (3) the filter functions are optimized according to the loss function with discriminators fixed, then for any user u : $\mathbf{r}_u \perp Z_u$ as $\lambda \rightarrow \infty$
 - * Training algorithm (*pseudo-code available in original paper*)
- Experiments (1059-62)
 - MovieLens and Insurance data sets
 - Fairness improvement: for the baseline methods, sensitive information of users can be learned from the data even if it is not explicitly used, but for both SM and CM it is hard for attackers to distinguish the sensitive features from the filtered user embeddings
 - Recommendation performance: both SM and CM can still achieve high recommendation quality, but there is a little sacrifice on recommendation performance
 - Ablation: with the increase of λ , recommendation performance is declining, but the system is getting fairer
- Conclusion (1062)

Table 70: First dimension

	Ethical theory	Categorization
Ethical theory type	Deontological ethics	
	Consequentialism	✓
	Virtue ethics	
	Particularism	
	Hybrid	Hierarchically specific
		Hierarchically nonspecific
	Configurable ethics	
	Ambiguous	
Conflict sensitivity	B/w AMAs	
	Within AMA	
	None	✓
Source	Developer	
	User	✓
	Expert	
	Lay people	
	(Legal) code/guidelines	
	Stakeholders	
	None	
Moral patient sensitivity	Yes	✓
	No	
	Unclear	

Table 71: Second dimension

Non-technical aspect		Categorization
Approach	Top-down	
	Bottom-up	✓
	Hybrid	
Contribution type	Model representation	
	Model selection	
	Judgment provision	
	Action selection/execution	✓
Evaluation	Test	Non-expert
		Expert
	Prove	Laws
		Model checker
		Logical proof ✓
	Informal	Example scenario ✓
		Face validity
	None	
Domain-specific	yes/no	✓

Table 72: Third dimension

Technical aspect		Categorization
Tech type	Logical reasoning	Deductive logic
		Non-monotonic logic
		Abductive logic
		Deontic logic
		Rule-based system
		Event calculus
		Knowledge representation and ontologies
		Inductive logic
	Probabilistic reasoning	Bayesian approach
		Markov models
		Statistical inference
	Learning	Inductive logic
		Decision tree
		Reinforcement learning
		Neural networks
		✓(adversary learning)
	Optimization	Evolutionary computing
		Case-based reasoning
Input	Case	Logical representation
		Numerical representation
		✓
Implementation availability	Sensor data	(Structured) language representation
		Specification details
		Implementation details
		✓
Other	Code (link) provided	
		Hardware
		Feedback
		Explanation
		User interface
	Automated processing	✓
		✗
		✗
		✗
		✗

Key Points 25

- Introduction (695-6)
 - Paper introduces graph counterfactual fairness, which considers the potential biases regarding sensitive attributes of each node and its neighboring nodes, as well as the biases led by the causal effect from sensitive attributes on the other variables
- Problem definition (696-7)
 - Notation (*table available in original paper*)
 - An encoder $\phi(\cdot)$ satisfies graph counterfactual fairness if for any node i : $P((Z_i)_{S \leftarrow s'} | X = \mathbf{X}, A = \mathbf{A}) = P((Z_i)_{S \leftarrow s''} | X = \mathbf{X}, A = \mathbf{A})$ for all $s' \neq s''$, where $s', s'' \in \{0, 1\}^n$ are arbitrary sensitive attribute values of all nodes, $Z_i = (\phi(X, A))_i$ denotes the node representations for node i (i.e., $\phi(\cdot)$ should minimize the distribution discrepancy between the representations $(\phi(X_{S \leftarrow s'}, A_{S \leftarrow s'}))_i$ and $(\phi(X_{S \leftarrow s''}, A_{S \leftarrow s''}))_i$ for any node i)
- The proposed framework: GEAR (697-9)
 - Subgraph generation
 - * For each node i , a context subgraph $\mathcal{G}^{(i)}$ is generated, based on which representations for the central node is learned
 - Counterfactual data augmentation
 - * Assumption: the sensitive attribute has no parent variable and it only causally influences the other node features, the graph structure, and the labels
 - * A graph variational auto-encoder module takes each context subgraph as input and encodes each node in the subgraph into a latent embedding $\mathbf{h}_i$, and a decoder reconstructs the original subgraph with the latent embeddings $\mathbf{H} = \{\mathbf{h}_1, \dots, \mathbf{h}_k\}$ and the sensitive attribute values of the k nodes in this subgraph
 - * Adversary learning is used to learn embeddings that are invariant to different sensitive attribute values of each node and their neighbors, with the expectation being that embeddings $\mathbf{H}$ can capture the latent variables which are informative of the unput subgraph but not causally influenced by the sensitive attribute of the nodes in the subgraph
 - * Two kinds of perturbations: (1) self-perturbation (flipping value of the sensitive attribute value of the central node and feeding the embeddings into the decoder to generate a set of counterfactual subgraphs $\overline{\mathcal{G}}^i$), (2) neighbor-perturbation (randomly flip the sensitive attribute value of any nodes except the central nodes to generate a set of counterfactual subgraphs $\underline{\mathcal{G}}^i$)
 - Fair representation learning
 - * Siamese networks are used to encode the original subgraphs $\mathcal{G}^{(i)}$ and counterfactual subgraphs $\overline{\mathcal{G}}^i$ and $\underline{\mathcal{G}}^i$ for each node i
 - * Subgraph encoder $\phi(\cdot)$ is trained to generate representations $z_i, \bar{z}_i, \underline{z}_i$, and the distance between the central node representations learned from the original subgraph and from the counterfactuals is minimized
- Experiments (699-702)
 - Datasets: (1) synthetic, (2) bail, (3) credit default
 - GEAR shows comparable prediction performance with state-of-the-art node representation learning methods, and it outperforms all fair node representation learning methods in terms of prediction performance
 - GEAR outperforms fair node representation learning methods in terms of fairness metrics that consider neighboring sensitive attributes and model prediction, and it performs well

in other fairness metrics

- Related work (702)
- Conclusion (702)

Table 73: First dimension

	Ethical theory	Categorization
Ethical theory type	Deontological ethics	
	Consequentialism	✓
	Virtue ethics	
	Particularism	
	Hybrid	Hierarchically specific
		Hierarchically nonspecific
	Configurable ethics	
	Ambiguous	
Conflict sensitivity	B/w AMAs	
	Within AMA	
	None	✓
Source	Developer	
	User	
	Expert	
	Lay people	
	(Legal) code/guidelines	
	Stakeholders	
	None	✓
Moral patient sensitivity	Yes	
	No	
	Unclear	✓

Table 74: Second dimension

Non-technical aspect		Categorization
Approach	Top-down	
	Bottom-up	✓
	Hybrid	
Contribution type	Model representation	
	Model selection	
	Judgment provision	
	Action selection/execution	✓
Evaluation	Test	Non-expert
		Expert
	Prove	Laws
		Model checker
		Logical proof
	Informal	Example scenario ✓
		Face validity
	None	
Domain-specific	yes/no	✗

Table 75: Third dimension

Technical aspect		Categorization
Tech type	Logical reasoning	Deductive logic
		Non-monotonic logic
		Abductive logic
		Deontic logic
		Rule-based system
		Event calculus
		Knowledge representation and ontologies
	Probabilistic reasoning	Inductive logic
		Bayesian approach
		Markov models
	Learning	Statistical inference
		Inductive logic
		Decision tree
		Reinforcement learning
		Neural networks
		✓(adversarial learning and Siamese networks)
		Evolutionary computing
	Optimization	
	Case-based reasoning	
Input	Case	Logical representation
		Numerical representation
		(Structured) language representation
Implementation availability	Sensor data	
	Specification details	
	Implementation details	
Other	Code (link) provided	
	Hardware	
	Feedback	
	Explanation	
	User interface	
	Automated processing	

Key Points 26

- Introduction (1423-4)
 - Graph embedding methods map each node to a low-dimensional embedding vector that reflects the node’s structural information, but the observed node connections in a graph are affected by certain sensitive attributes, which means that learned node embeddings can inherit undesired sensitive information and lead to severe bias and fairness concerns
 - Paper proposes to learn node embeddings from an underlying bias-free graph whose edges are generated without influence from sensitive attributes
 - Two alternative methods are proposed to uncover the bias-free graph from the given graph for learning node embeddings: (1) weighting-based method, and (2) regularization
- Related work (1424)
- Preliminaries (1424-5)
 - Graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{A})$
 - Embeddings are learned via the link prediction task, in which a scoring function is defined to predict the probability of an edge $(u, v) \in \mathcal{E}$
 - Given a node u , its embedding $\mathbf{z}_u$ is unbiased with respect to attribute i if it is independent from i
- Effect of attributes in graph generation (assumed generative process behind an observed graph) (1425-6)
 - There is an underlying structural graph behind an observed graph, whose edge distribution is governed by global graph structural properties (e.g., degree distributions, diameter, clustering coefficients)
 - Attributes in $\mathcal{A}$ will modify the structural edge distribution
 - The modified edge distribution is used to generate the observed graph
- Unbiased graph embedding from a bias-free graph (1426-7)
 - Main idea: get rid off the sensitive attributes and modify the structural edge probabilities by only conditioning on non-sensitive attributes, which gives edge probabilities of a bias-free graph, from which unbiased node embeddings can be learned (i.e., generate $\tilde{\mathcal{G}}$ and learn embeddings from $\tilde{\mathcal{G}}$ instead of $\mathcal{G}$)
 - Weighting-based UGE (UGE-W): reweighing the loss term for $\mathcal{G}$ on each edge, where the weights are based on the conditional probability of an edge between u and v given the corresponding non-sensitive attribute values on them and the structural parameters
 - Regularization-based UGE (UGE-R): adding a regularization term to the loss function which requires the produced edge distributions from the embeddings learned from $\tilde{\mathcal{G}}$ to be as similar as possible to each other with and without the sensitive attributes
 - UGE-W is a sufficient condition for unbiased graph embedding, since it directly learns node embeddings from a bias-free graph, and UGE-R is a necessary condition, as it requires the learned embeddings to satisfy the properties of a bias-free graph
 - Combined method (UGE-C): combination of UGE-W and UGE-R
- Experiments (1427-30)
 - Embeddings from UGE methods have the least bias against all baselines with respect to all sensitive attributes and datasets, with UGE-C usually producing the best debiasing effect
 - UGE-C can usually achieve a better debiasing effect, without sacrificing too much utility
 - Using the debiased embeddings in downstream tasks, UGE methods achieve significantly better fairness in terms of demographic parity and equalized opportunity (except for equalized opportunity on the age attribute), with UGE-C achieving the most fair predictions

- UGE-C achieves a better trade-off between unbiasedness and utility (than the baseline method)

Table 76: First dimension

	Ethical theory	Categorization
Ethical theory type	Deontological ethics	
	Consequentialism	✓
	Virtue ethics	
	Particularism	
	Hybrid	Hierarchically specific
		Hierarchically nonspecific
	Configurable ethics	
	Ambiguous	
Conflict sensitivity	B/w AMAs	
	Within AMA	
	None	✓
Source	Developer	
	User	
	Expert	
	Lay people	
	(Legal) code/guidelines	
	Stakeholders	
	None	✓
Moral patient sensitivity	Yes	
	No	
	Unclear	✓

Table 77: Second dimension

Non-technical aspect		Categorization
Approach	Top-down	
	Bottom-up	✓
	Hybrid	
Contribution type	Model representation	
	Model selection	
	Judgment provision	
	Action selection/execution	✓
Evaluation	Test	Non-expert
		Expert
	Prove	Laws
		Model checker
		Logical proof
	Informal	Example scenario ✓
		Face validity
	None	
Domain-specific	yes/no	✗

Table 78: Third dimension

Technical aspect		Categorization
Tech type	Logical reasoning	Deductive logic
		Non-monotonic logic
		Abductive logic
		Deontic logic
		Rule-based system
		Event calculus
		Knowledge representation and ontologies
		Inductive logic
	Probabilistic reasoning	Bayesian approach
		Markov models
		Statistical inference
	Learning	Inductive logic
		Decision tree
		Reinforcement learning
		Neural networks
		✓(link prediction in GNNs)
	Evolutionary computing	
	Optimization	
	Case-based reasoning	
Input	Case	Logical representation
		Numerical representation
		✓
	(Structured) language representation	
	Sensor data	
Implementation availability	Specification details	
	Implementation details	
	Code (link) provided	
Other	Hardware	
	Feedback	
	Explanation	
	User interface	
	Automated processing	

Key Points 27

- Introduction (4462-3)
 - Paper proposes a fairness-aware news recommendation approach (FairRec)
- Related work (4463)
- Methodology (4463-5)
 - Problem definition
 - * Recommendation unfairness: if sensitive attribute z of user u can be predicted from the recommended candidate news set more accurately, the recommendation result is more unfair
 - Framework of FairRec
 - * Architecture (*figure available in original paper*)
 - * News model: learns embeddings of candidate news
 - * Bias-free user model: learns bias-free user embeddings of users
 - * Click scoring model: computes the fairness-aware news raking scores based on the bias-free user embedding and candidate news embeddings
 - Decomposed adversarial learning with orthogonality regularization
 - * User interest model is decomposed into two components: (1) bias-aware one which aims to learn bias-aware user embeddings that capture the bias information on sensitive user attributes, (2) bias-free one which only encodes attribute-independent information into bias-free user embeddings
 - * To make the bias-aware user embeddings more attribute-discriminative, a sensitive attribute prediction task is applied to the bias-aware user embeddings
 - * In order to eliminate bias information about sensitive user attributes from the bias-free user embeddings, adversarial learning and an orthogonality regularization method are used
 - Model training
 - * Bias-free and bias-aware user embeddings are added together to form unified user embeddings, which are used for training the recommendation model
- Experiments (4465-8)
 - Compared with methods that do not consider, the fairness of recommendation, fairness-aware methods yield better recommendation fairness, with FairRec achieving substantially better recommendation fairness than two of the three methods which do consider fairness
 - FairRec can improve recommendation fairness, while keeping a good recommendation performance

Table 79: First dimension

	Ethical theory	Categorization
Ethical theory type	Deontological ethics	
	Consequentialism	✓
	Virtue ethics	
	Particularism	
	Hybrid	Hierarchically specific
		Hierarchically nonspecific
	Configurable ethics	
	Ambiguous	
Conflict sensitivity	B/w AMAs	
	Within AMA	
	None	✓
Source	Developer	
	User	
	Expert	
	Lay people	
	(Legal) code/guidelines	
	Stakeholders	
	None	✓
Moral patient sensitivity	Yes	
	No	
	Unclear	✓

Table 80: Second dimension

Non-technical aspect		Categorization
Approach	Top-down	
	Bottom-up	✓
	Hybrid	
Contribution type	Model representation	
	Model selection	
	Judgment provision	
	Action selection/execution	✓
Evaluation	Test	Non-expert
		Expert
	Prove	Laws
		Model checker
		Logical proof
	Informal	Example scenario ✓
		Face validity
	None	
Domain-specific	yes/no	✓

Table 81: Third dimension

Technical aspect		Categorization
Tech type	Logical reasoning	Deductive logic
		Non-monotonic logic
		Abductive logic
		Deontic logic
		Rule-based system
		Event calculus
		Knowledge representation and ontologies
		Inductive logic
	Probabilistic reasoning	Bayesian approach
		Markov models
		Statistical inference
	Learning	Inductive logic
		Decision tree
		Reinforcement learning
		Neural networks ✓
		Evolutionary computing
	Optimization	
	Case-based reasoning	
Input	Case	Logical representation
		Numerical representation ✓
		(Structured) language representation
Implementation availability	Sensor data	
	Specification details	✓
	Implementation details	
	Code (link) provided	
Other	Hardware	✗
	Feedback	✗
	Explanation	✗
	User interface	✗
	Automated processing	✗

Key Points 28

- Introduction and motivation (2-4)
 - Paper treats multi-stakeholder fairness-aware recommendation as a multi-objective optimization (MOO) and proposes a scalable framework Multi-FR, where both producer and consumer fairness are considered
- Related work (4-6)
- Preliminaries (6-7)
 - Notation (*table available in original paper*)
 - Problem formulation
 - * Top- K item recommendation task: Given the training item set $\mathcal{S}_u^+$, and the non-empty test item set $\mathcal{T}_u^+$ of user u , the model aims to recommend an ordered set of K items $\mathcal{X}_u$ such that $|\mathcal{X}_u| = K$ and $\mathcal{X}_u \cap \mathcal{S}_u^+ = \emptyset$, where $\mathcal{I}_u^+ := \{i \in \mathcal{I} | \mathbf{Y}_{u,i} = 1\}$ is the set of items user u has interacted with (preference data) that is split into training set $\mathcal{S}_u^+$ and test set $\mathcal{T}_u^+$ requiring that $\mathcal{S}_u^+ \cup \mathcal{T}_u^+ = \mathcal{I}_u^+$ and $\mathcal{S}_u^+ \cap \mathcal{T}_u^+ = \emptyset$
 - * Goal to simultaneously balance the following objectives: (1) recommendation accuracy), (2) consumer-sided fairness, (3) producer-sided fairness
- Objective construction (7-12)
 - Objective for recommendation quality
 - * ...
 - Fairness objective for multi-stakeholder
 - * Consumer fairness constraint
 - Aim to make the satisfaction of different groups with sensitive features be equal
 - Use of NDCG@K to measure consumer satisfaction, but since items ranked at higher positions generally receive more attention from consumers, the fairness at each prefix of the ranking is also significant
 - $\mathbf{s}^{G_i} \in \mathbb{R}^K$ is a satisfaction vector for group G_i among the consumers, where the k th entry equals to NDCG@ k , $k = 1, \dots, K$
 - Assuming n^C consumer groups, the fairness constraint on a group level can be defined as the mean of the pair-wise satisfaction difference across all groups: $\mathcal{L}^{C-Fair} = \frac{1}{\binom{n^C}{2}} \sum_{1 \leq i < j \leq n^C} \|\mathbf{s}^{G_i} - \mathbf{s}^{G_j}\|_2^2$, where $\binom{n^C}{2}$ is the number of combinations between different groups
 - Considering a batch of b consumers, the satisfaction of group G_i is defined as: $\mathbf{s}^{G_i} = \frac{\mathbf{m}^{G_i} \cdot \mathbf{N}}{\|\mathbf{m}^{G_i}\|_1}$, where $\mathbf{m}^{G_i} \in \mathbb{R}^b$ is a multi-hot vector, $\mathbf{N} \in \mathbb{R}^{b \times K}$ is a matrix containing NDCG@1 to NDCG@ K for all users in a batch
 - Consideration of gender-based and age-based fairness
 - * Producer fairness constraint
 - Goal: find a ranking strategy that can offer a fair probability of exposure on items based on their merits
 - Assuming n^P groups among all items, the fairness constraint on the producer side can be defined as: $\mathcal{L}^{P-Fair} = \|\epsilon - \epsilon^*\|_2^2$, where $\epsilon \in \mathbb{R}^{n^P}$ is a vector representing the distribution of exposure on items from different groups (system exposure), and $\epsilon^* \in \mathbb{R}^{n^P}$ is the ideal exposure distribution proportional to the true relevance of items (target exposure)
 - Computing system exposure ϵ : ranking of all $|\mathcal{I}|$ items for b users in a batch based on the relevance score

- Computing target exposure ϵ^* : assumption that all relevant items should have the same high probability of being selected to the top of the raking, while other irrelevant items should equally share the rest amount of exposure at a low level
- Consideration of popularity-based fairness and genre-based fairness
- Differentiable approximation of the ranking
 - * A continuous approximation of the ranking function is used to make it differentiable and amenable to gradient-based optimization
 - * Given the approximated differentiable ranks of items, an objective for standard relevance metrics (e.g., discounted cumulative gain) can be directly optimized using gradient descent
 - * Non-differentiable ranking of items is also addressed for the producer side
- Multi-FR: Multi-stakeholder fairness-aware recommendation (12-16)
 - Multi-objective optimization (MOO)
 - * MOO aims to find a solution that can optimize a set of objectives simultaneously:

$$\min_{\theta} \mathcal{L}(\theta) = \min_{\substack{\theta^C \\ \theta^{S_1}, \dots, \theta^{S_t}}} \mathcal{L}(\theta^C, \theta^{S_1}, \dots, \theta^{S_t}) = \min_{\substack{\theta^C \\ \theta^{S_1}, \dots, \theta^{S_t}}} \begin{bmatrix} \mathcal{L}_1(\theta^C, \theta^{S_1}) \\ \mathcal{L}_2(\theta^C, \theta^{S_2}) \\ \vdots \\ \mathcal{L}_t(\theta^C, \theta^{S_t}) \end{bmatrix},$$

where $\mathcal{L}$ is the full objective, $\mathcal{L}_1, \dots, \mathcal{L}_t$ are t different objectives, θ^C is the common parameter shared by all objectives, while $\theta^{S_1}, \dots, \theta^{S_t}$ are object-specific parameters

- * Pareto optimality
 - A solution θ_1 dominates another solution θ_2 if for all objectives $\mathcal{L}_i(\theta_1^C, \theta_1^{S_i}) \leq \mathcal{L}_i(\theta_2^C, \theta_2^{S_i})$ where $i \in \{1, \dots, t\}$, and there exists at least one objective $j \in \{1, \dots, t\}$, where $\mathcal{L}_j(\theta_1^C, \theta_1^{S_j}) < \mathcal{L}_j(\theta_2^C, \theta_2^{S_j})$
 - A solution is Pareto optimal if there does not exist any other solution that dominates it
 - The set of Pareto optimal solutions is called the Pareto set, and the curve of the points in the Pareto set is called the Pareto frontier
- Multiple gradient descent algorithm (MGDA)
 - * Overall objective of solving an MOO problem by MGDA is usually a weighted summation of t single objectives: $\mathcal{L}(\theta) = \mathcal{L}(\theta^C, \theta^{S_1}, \dots, \theta^{S_t}) = \sum_{i=1}^t \alpha_i \cdot \mathcal{L}_i(\theta^C, \theta^{S_i})$, where the coefficients of all the objectives satisfy $\sum_{i=1}^t \alpha_i = 1$ and $\alpha_i \geq 0$ for $i = 1, \dots, t$
 - * A Pareto optimal solution must be Pareto stationary, i.e., it must satisfy the Karush-Kuhn-Tucker (KKT) conditions
- Solving the MOO problem (where Procedure 1 is based on the analytical solution of the case with only two objectives in the loss function) (*pseudo-code available in original paper*)
- The Multi-FR framework and the overall objective
 - * Algorithm (*pseudo-code available in original paper*)
 - * Since there are no task-specific parameters, θ^S can be omitted, and $\theta = \theta^C = \Theta = \llbracket \Theta^U, \Theta^I \rrbracket$, where Θ includes user and item embeddings

* Overall training objective can be formulated as:

$$\mathcal{L} = \mathcal{L}^{Accuracy} + \sum_{i=1}^m \alpha_i^C \cdot \mathcal{L}_i^{C-Fair} + \sum_{j=1}^n \alpha_j^P \cdot \mathcal{L}_j^{P-Fair},$$

$$\text{s.t.}, \alpha^A + \sum_{i=1}^m \alpha_i^C + \sum_{j=1}^n \alpha_j^P = 1.$$

$$\alpha^A \geq 0, \alpha_i^C \geq 0, \alpha_j^P \geq 0, \forall i, j,$$

where m is the number of fairness constraints on the consumer side, and n is the number of fairness constraints on the producer side

– Solution selection

* To select a solution from the Pareto set of Pareto optimal solution, the Least Misery Strategy is adopted, i.e., the solution coming from the q' th round that minimizes the highest loss value across all objectives is chosen: $\min_{1 \leq q' \leq q} \max\{\mathcal{L}_1^{q'}, \mathcal{L}_2^{q'}, \dots, \mathcal{L}_t^{q'}\}$, where t is the number of objectives and q is the number of running rounds

— Experiments (16-25)

– Datasets: MovieLens100K, MovieLens1M, FM-reduced

– Evaluation metrics

* Metrics for measuring recommendation accuracy: (1) Recall@ K , (2) NDCG@ K

* Self-defined metrics for measuring fairness: (1) Disparity _{u} (user side), (2) Disparity _{i} (item side)

* General metrics for measuring fairness and diversity: (1) Gini Index, (2) Popularity Rate, (3) Simpson's Diversity Index

– Method studied

* Three models as recommendation backbones: (1) Bayesian Personalized Ranking-based Matrix Factorization (BPRMF), (2) Weighted Regularized Matrix Factorization (WRMF) (3) Neural Graph Collaborative Filtering (NGCF)

* Three fairness-aware approaches: (1) Fairness of Exposure in Rankings (FOEIR), (2) FairRec, (3) TFRM

– Experiment settings

– Experimental results and analysis

* Multi-FR achieves significant improvements regarding all the fairness and diversity metrics

* There is a conflict between recommendation accuracy and fairness, but Multi-FR can largely improve fairness and diversity with little drop in accuracy

* Multi-FR has reasonable training time, especially when the number of fairness constraints increases

Table 82: First dimension

	Ethical theory	Categorization
Ethical theory type	Deontological ethics	
	Consequentialism	✓
	Virtue ethics	
	Particularism	
	Hybrid	Hierarchically specific
		Hierarchically nonspecific
	Configurable ethics	
	Ambiguous	
Conflict sensitivity	B/w AMAs	
	Within AMA	
	None	✓
Source	Developer	
	User	
	Expert	
	Lay people	
	(Legal) code/guidelines	
	Stakeholders	
	None	✓
Moral patient sensitivity	Yes	
	No	
	Unclear	✓

Table 83: Second dimension

Non-technical aspect		Categorization
Approach	Top-down	
	Bottom-up	✓
	Hybrid	
Contribution type	Model representation	
	Model selection	
	Judgment provision	
	Action selection/execution	✓
Evaluation	Test	Non-expert
		Expert
	Prove	Laws
		Model checker
		Logical proof
		Example scenario ✓
	Informal	Face validity
		None
Domain-specific	yes/no	✓

Table 84: Third dimension

Technical aspect		Categorization
Tech type	Logical reasoning	Deductive logic
		Non-monotonic logic
		Abductive logic
		Deontic logic
		Rule-based system
		Event calculus
		Knowledge representation and ontologies
		Inductive logic
	Probabilistic reasoning	Bayesian approach
		Markov models
		Statistical inference
	Learning	Inductive logic
		Decision tree
		Reinforcement learning
		Neural networks
		Evolutionary computing
	Optimization	✓
	Case-based reasoning	
Input	Case	Logical representation
		Numerical representation
		(Structured) language representation
Implementation availability	Sensor data	
	Specification details	
	Implementation details	
	Code (link) provided	
Other	Hardware	
	Feedback	
	Explanation	
	User interface	
	Automated processing	

29 Sholla *et al.* (2021)

Key Points 29

- Introduction (1487-8)
 - Paper proposes a neuro-fuzzy system that is able to implement ethics pertinent to the context of a smart thing and also learn appropriate ethical behavior
- Background (1488-9)
- Methodology (1489-90)
- Neuro fuzzy system implementing ethics (1490-4) (*figure available in original paper*)
 - Neuro fuzzy system architecture
 - * Layer 1 (ethics input layer): nodes represent input FEVs (fuzzy ethics variables) to express the EOP (ethics of operation) of a device, and the outgoing links represent the number of MFs (membership functions) used to represent the input FEVs
 - * Layer 2 (fuzzification layer): maps input FEVs to linguistic labels used to represent them, and calculates membership strength to the corresponding label based on the MFs used; nodes represent possible rule antecedents for the next layer
 - * Layer 3 (ethics rule layer): specifies fuzzy ethics rules, calculates SEV (scaled ethics value) for each rule, and represents the ethics mode applicable; incoming links represent rule antecedents and outgoing links represent rule consequents (cf. Sholla (2021a) summary for more details)
 - * Layer 4 (ethics output layer): nodes represent ethics rule consequents, signifying various possible outcomes (linguistic labels) that can result due to the interaction between the device and its context
 - * Layer 5 (defuzzification layer): represents the final output, where among the various possible linguistic labels corresponding to an output FEV, an ethically consistent label is used as the final output
 - Learning strategies
 - * NFS specifies initial ethical requirements in terms of fuzzy rules and learns appropriate ethical responses via neural networks
 - * Learning feature enables to implement ethics depending on personal preferences of individuals
 - * Ethics dialect: variation of ethics among members of a society from the general ethical trend prevalent in a society
 - * Fuzzy rules represent human (expert) knowledge, and neural networks are only used to fine-tune the ethical response already specified by means of fuzzy logic rules
 - * Learning mechanism adjusts the SEV, but rules belonging to forbidden and obligatory categories are kept out of the learning process (thus, $R_{max} = 0.8$ and $R_{min} = 0.2$)
 - * In case of positive reinforcement, a random number between the current SEV of the rule and offset is added, and subtracted in case of negative reinforcement
- Implementation of the proposed system (1494-7)
 - Health care example (cf. summary of Sholla (2021a))
 - Learning healthcare ethics: for each rule, a sample of 100 university students is asked whether they agree, disagree, or have no preference for the corresponding scenario (reflecting R_{min}), and for each question, they are asked to rate the confidence of their opinion (reflecting ethics strength m)
 - Use of MATLAB along with nftool for implementing the neuro fuzzy system
- Analysis of results (1497-8) (cf. summary of Sholla (2021a))
- Discussion (1498-9)

Table 85: First dimension

	Ethical theory	Categorization
Ethical theory type	Deontological ethics	✓
	Consequentialism	
	Virtue ethics	
	Particularism	
	Hybrid	Hierarchically specific
		Hierarchically nonspecific
	Configurable ethics	
	Ambiguous	✓(expert knowledge and folk morality)
Conflict sensitivity	B/w AMAs	
	Within AMA	✓
	None	
Source	Developer	
	User	
	Expert	✓
	Lay people	✓
	(Legal) code/guidelines	
	Stakeholders	
	None	
Moral patient sensitivity	Yes	✓
	No	
	Unclear	

Table 86: Second dimension

Non-technical aspect		Categorization
Approach	Top-down	
	Bottom-up	
	Hybrid	✓
Contribution type	Model representation	
	Model selection	
	Judgment provision	✓
	Action selection/execution	
Evaluation	Test	Non-expert
		Expert
	Prove	Laws
		Model checker
		Logical proof
		Example scenario ✓
	Informal	Face validity
		None
Domain-specific	yes/no	✗

Table 87: Third dimension

Technical aspect		Categorization
Tech type	Logical reasoning	Deductive logic
		Non-monotonic logic
		Abductive logic
		Deontic logic
		Rule-based system
		✓(fuzzy logic)
	Probabilistic reasoning	Event calculus
		Knowledge representation and ontologies
		Inductive logic
		Bayesian approach
		Markov models
		Statistical inference
	Learning	Inductive logic
		Decision tree
		Reinforcement learning
		Neural networks
		✓
	Optimization	Evolutionary computing
	Case-based reasoning	
Input	Case	Logical representation
		✓
		Numerical representation
Implementation availability	Sensor data	✓
		(Structured) language representation
Other	Implementation details	Code (link) provided
Other	Automated processing	Hardware
		Feedback
		Explanation
		User interface
		Automated processing

30 Stenseke (2024)

Key Points 30

- Introduction
 - Paper presents an experimental implementation of AMAs that focuses on virtue ethics
 - Virtues can be understood as traits that support an agent to achieve eudaimonia (i.e. flourishing, well-being)
 - Phronesis describes moral wisdom gained from the experience a virtuous adult has
- Materials and methods
 - Artificial virtuous agents
 - * Idea behind the model
 - Dispositional virtues can be functionally carried out by artificial neural networks that learn from experience in light of an eudaimonic reward by means of a phronetic learning system
 - Virtues determine the agent's action based on input from the environment
 - Learning feedback is received based on whether the performed action increased or decreased the agent's eudaimonia
 - * Six main components of the model (*figure available in original paper*)
 - Input network: environmental input is received and parsed by the input network
 - Virtue networks: a virtue network classifies the input so as to produce the most suitable action (in the most simple case, a binary classifier is used to determine whether an agent acts in one way or another and, in more complex cases, a virtue can be an entire network of nodes that outputs one of several possible actions)
 - Action output: the agent executes the action determined by the invoked virtue network
 - Outcome network: new environmental input is received in order to classify the outcome of the performed action
 - Eudaimonic reward system: the classified outcome is evaluated, for which the following distinction is used: eudaimonic type (e-type), i.e., the values or goals the agent strives toward, vs. eudaimonic value (e-value), i.e., a quantitative measure of the amount of e-type an agent has obtained
 - Phronetic learning system: if a certain action led to an increase of e-value, the virtue that produced the action will receive positive reinforcement, whereas if e-value is decreased, the virtue receives punishment
 - * An AVA as MDP: $\langle S, A, \Phi, P, \gamma \rangle$
 - S : set of states
 - A : set of actions
 - $\Phi_a(s, s')$: eudaimonic reward the agent receives transition from s to s' by performing action a
 - γ : discount factor ($0 \leq \gamma \leq 1$), specifying whether the agent prefers short- or long-term rewards
 - * To maximize the eudaimonic reward Φ over time, the AVA needs to learn a policy $\pi(s, a)$ that determines what action to perform given a specific state
 - * Three practical considerations that are critical for implementation success
 - Static vs. dynamic: which features should be hard-coded and remain unchanged, and which features should be able to change over time
 - Learning of situations and outcomes: to not treat rewards and states as given (i.e., to allow for learning of situations and outcomes), (1) partially observable Markov

- decision processes (POMDP), or (2) designated subsystems (e.g., actor-critic methods) could be used, or (3) designated input and output networks could be trained through supervised learning
 - Learning from observation and moral exemplars: if an agent can observe that the outcome of a specific action performed by another agent increased some recognizable e-value, it could be used for learning
 - Ethical environment: BridgeWorld
 - * Setting
 - Five islands connected by four bridges: a central island where agents live and four surrounding islands where food grows
 - Agents need to cross bridges to collect food, but there is a chance of them falling into the ocean
 - A fallen agent can survive if it is saved by another agent, with a chance of falling into the ocean themselves
 - An agent dies from starvation when its energy reserve R is less than 1 at the time of eating
 - Food only exists on one island at a given time, and agents are able to remember the current location of food if they have found it
 - * Each simulation cycle, the agents do the following four things
 - Move from the central island to one of the four surrounding islands
 - If the agent moved to the island where food grows, it receives food (R increases by one food value FV)
 - Move back to the central island
 - Eat (R decreases by 1)
 - * Each simulation cycle, agents are able to interact with each other in three ways
 - Ask about food location, where an agent can react by (i) telling where the food is or (ii) lying
 - Call for help, where an agent can react by (i) trying to rescue another agent or (ii) ignore
 - Beg for food, where an agent can react by (i) giving food or (ii) ignore
 - * Three moral dilemmas that may arise
 - If agents only act in their own self-interest, resources would not be utilized efficiently at a collective level
 - If agents always act selflessly, it could result in more overall casualties due to self-sacrificial behavior
 - In a mixed population, selfless agents could be exploited by self-interested agents
 - * In contrast to conventional game-theoretic models, the behavior of virtuous agents does not depend on pre-given strategies that could be analyzed in terms of Nash equilibria for the different payoff matrices and goods, since an agent's behavior depends on dispositional virtues, which may develop over time according to a certain conception of eudaimonia
 - Virtuous agents in BridgeWorld
 - * Virtues and virtuous actions
 - Each agent is equipped with three virtues that relate to the three types of interactions: (1) courage, (2) generosity, and (3) honesty
 - Each virtue represents a balance between deficiency and excess, modeled as a value between -1 and 1
 - Each virtue is a threshold function $f(x)$ that determines, given some input x , weight

w , and threshold ψ , whether an agent acts in one way or another^a:

$$f(x) = \begin{cases} 1 & \text{if } wx \geq \psi \\ 0 & \text{if } wx \leq \psi \end{cases}$$

* Phronetic learning system

- The learning system is invoked each time an agent responds to an action
- It takes three input parameters: (1) the action taken by the agent, (2) the relevant virtue affected, (3) the magnitude of an event
- If the action increased the agent's e-value in light of its e-type, the relevant virtue weight is reinforced, and vice versa if the e-value decreased
- The learning algorithm updates the weight in the threshold function, with the aim of finding the optimal state-action policy that maximizes cumulative e-value:

$$\operatorname{argmax}_{\pi} E \left[\sum_{t=0}^{\infty} \Phi(s_t, a_t) \middle| \pi \right]$$

* Eudaimonia: three different e-types used in the experiments

- Selfish (S): an agent with a selfish e-type rewards actions that are beneficial only for the agent itself and punishes actions that go against the self-interest of the agent
- Praise/blame (P/B): an agent with the praise/blame e-type rewards actions that maximize praise and minimize blame received from others
- Selfish/selfless (S/S)^b
- Additional learning from exemplars (E)

* Simulation cycle

- Initialization: virtue values assigned randomly, e-value set to 0, e-type set to one of the defined e-types
- Phases that occur at each simulation cycle: (1) ask for food location, (2) move to food, (3) help fallen, (4) collect food, (5) move back, (6) beg for food, (7) eating and starving, (8) moral exemplar, (9) death and rebirth

— Results

- Mean death rate and standard deviation at iteration 1000 of ten repeated in seven different conditions, where NL serves as baseline without any learning nor e-type (*table available in original paper*)
- Average death rate and virtue values (*figures available in original paper*)

— Discussion

— Conclusion

^a $wx = \psi$ included in both cases in the original paper

^bNot listed in the original paper, because the paragraph seems to have gone missing.

Table 88: First dimension

	Ethical theory	Categorization
Ethical theory type	Deontological ethics	
	Consequentialism	
	Virtue ethics	✓
	Particularism	
	Hybrid	Hierarchically specific
		Hierarchically nonspecific
	Configurable ethics	
	Ambiguous	
Conflict sensitivity	B/w AMAs	✓
	Within AMA	
	None	
Source	Developer	
	User	
	Expert	
	Lay people	
	(Legal) code/guidelines	
	Stakeholders	
	None	✓
Moral patient sensitivity	Yes	
	No	
	Unclear	✓

Table 89: Second dimension

Non-technical aspect		Categorization
Approach	Top-down	
	Bottom-up	✓
	Hybrid	
Contribution type	Model representation	
	Model selection	
	Judgment provision	
	Action selection/execution	✓
Evaluation	Test	Non-expert
		Expert
	Prove	Laws
		Model checker
		Logical proof
		Example scenario
	Informal	Face validity
		None
Domain-specific	yes/no	✗

Table 90: Third dimension

Technical aspect		Categorization
Tech type	Logical reasoning	Deductive logic
		Non-monotonic logic
		Abductive logic
		Deontic logic
		Rule-based system
		Event calculus
		Knowledge representation and ontologies
		Inductive logic
	Probabilistic reasoning	Bayesian approach
		Markov models
		Statistical inference
	Learning	Inductive logic
		Decision tree
		Reinforcement learning ✓
		Neural networks ✓
		Evolutionary computing
	Optimization	
	Case-based reasoning	
Input	Case	Logical representation
		Numerical representation ✓
		(Structured) language representation
Implementation availability	Sensor data	
	Specification details	✓
	Implementation details	
	Code (link) provided	
Other	Hardware	✗
	Feedback	✗
	Explanation	✗
	User interface	✗
	Automated processing	✗

31 Bibliography

- Agarwal C, Lakkaraju H, Zitnik M (2021). “Towards a unified framework for fair and stable graph representation learning.” In C de Campos, MH Maathuis (eds.), “Proceedings of the Thirty-Seventh Conference on Uncertainty in Artificial Intelligence,” volume 161 of *Proceedings of Machine Learning Research*, pp. 2114–2124. PMLR, Online. URL <https://proceedings.mlr.press/v161/agarwal21b.html>.
- Baumann J, Hannák A, Heitz C (2022). “Enforcing Group Fairness in Algorithmic Decision Making: Utility Maximization Under Sufficiency.” In “Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency,” FAccT ’22, p. 2315–2326. Association for Computing Machinery, New York, NY, USA. ISBN 9781450393522. doi:<https://doi.org/10.1145/3531146.3534645>.
- Baumann J, Heitz C (2022). “Group Fairness in Prediction-Based Decision Making: From Moral Assessment to Implementation.” In “2022 9th Swiss Conference on Data Science (SDS),” pp. 19–25. doi:<https://doi.org/10.1109/SDS54800.2022.00011>.
- Bourgne G, Sarmiento C, Ganascia JG (2022). “ACE modular framework for computational ethics: dealing with multiple actions, concurrency and omission.” In “International Workshop on Computational Machine Ethics, Nov. 2021,” .
- Bringsjord S, Govindarajulu NS, Giancola M (2021). “Automated argument adjudication to solve ethical problems in multi-agent environments.” *Paladyn, Journal of Behavioral Robotics*, **12**(1), 310–335. doi:<https://doi.org/10.1515/pjbr-2021-0009>.
- Cardoso RC, Ferrando A, Dennis LA, Fisher M (2022). “Implementing Ethical Governors in BDI.” In N Alechina, M Baldoni, B Logan (eds.), “Engineering Multi-Agent Systems,” pp. 22–41. Springer International Publishing, Cham. ISBN 978-3-030-97457-2. doi:https://doi.org/10.1007/978-3-030-97457-2_2.
- Chaput R, Duval J, Boissier O, Guillermin M, Hassas S (2021). “A Multi-Agent Approach to Combine Reasoning and Learning for an Ethical Behavior.” In “Proceedings of the 2021 AAAI/ACM Conference on AI, Ethics, and Society,” AIES ’21, p. 13–23. Association for Computing Machinery, New York, NY, USA. ISBN 9781450384735. doi:<https://doi.org/10.1145/3461702.3462515>.
- Chen S, Arkin R, Borenstein J, Wagner AR (2022). “Case-based Robotic Architecture with Multiple Underlying Ethical Frameworks for Human-Robot Interaction.” In “7th International Conference on Robot Ethics and Standards, Seoul, South Korea,” .
- Dennis LA, Bentzen MM, Lindner F, Fisher M (2021). “Verifiable machine ethics in changing contexts.” In “Proceedings of the AAAI Conference on Artificial Intelligence,” volume 35, pp. 11470–11478. doi:<https://doi.org/10.1609/aaai.v35i13.17366>.
- Ecoffet A, Lehman J (2021). “Reinforcement Learning Under Moral Uncertainty.” In M Meila, T Zhang (eds.), “Proceedings of the 38th International Conference on Machine Learning,” volume 139 of *Proceedings of Machine Learning Research*, pp. 2926–2936. PMLR. URL <https://proceedings.mlr.press/v139/ecoffet21a.html>.
- Erdoğan G (2022). “The ethical shortlisting problem.” *Computers Operations Research*, **138**, 105593. ISSN 0305-0548. doi:<https://doi.org/10.1016/j.cor.2021.105593>. URL <https://www.sciencedirect.com/science/article/pii/S0305054821003166>.
- Frisch G, Leger JB, Grandvalet Y (2021). “Stereotype-aware collaborative filtering.” In “2021 16th Conference on Computer Science and Intelligence Systems (FedCSIS),” pp. 69–79. doi:<https://doi.org/10.15439/2021F117>.

- Giancola M, Bringsjord S, Govindarajulu NS, Varela C (2022). *Making Maximally Ethical Decisions via Cognitive Likelihood and Formal Planning*, pp. 127–142. Springer International Publishing, Cham. ISBN 978-3-031-09823-9. doi:https://doi.org/10.1007/978-3-031-09823-9_10.
- Kim TW, Hooker J, Donaldson T (2021). “Taking principles seriously: A hybrid approach to value alignment in artificial intelligence.” *Journal of Artificial Intelligence Research*, **70**, 871–890. doi:<https://doi.org/10.1613/jair.1.12481>.
- Kolker S, Dennis L, Fraga Pereira R, Xu M (2023). “Uncertain Machine Ethical Decisions Using Hypothetical Retrospection.” In N Fornara, J Cheriyan, A Mertzani (eds.), “Coordination, Organizations, Institutions, Norms, and Ethics for Governance of Multi-Agent Systems XVI,” pp. 161–181. Springer Nature Switzerland, Cham. ISBN 978-3-031-49133-7. doi:https://doi.org/10.1007/978-3-031-49133-7_9.
- Lee JK, Bu Y, Rajan D, Sattigeri P, Panda R, Das S, Wornell GW (2021). “Fair Selective Classification Via Sufficiency.” In M Meila, T Zhang (eds.), “Proceedings of the 38th International Conference on Machine Learning,” volume 139 of *Proceedings of Machine Learning Research*, pp. 6076–6086. PMLR, Online. URL <https://proceedings.mlr.press/v139/lee21b.html>.
- Li Y, Chen H, Xu S, Ge Y, Zhang Y (2021). “Towards Personalized Fairness based on Causal Notion.” In “Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval,” SIGIR ’21, p. 1054–1063. Association for Computing Machinery, New York, NY, USA. ISBN 9781450380379. doi:<https://doi.org/10.1145/3404835.3462966>.
- Ma J, Guo R, Wan M, Yang L, Zhang A, Li J (2022). “Learning Fair Node Representations with Graph Counterfactual Fairness.” In “Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining,” WSDM ’22, p. 695–703. Association for Computing Machinery, New York, NY, USA. ISBN 9781450391320. doi:<https://doi.org/10.1145/3488560.3498391>.
- Medda G, Fabbri F, Marras M, Boratto L, Fenu G (2024). “GNNUERS: Fairness Explanation in GNNs for Recommendation via Counterfactual Reasoning.” *ACM Trans. Intell. Syst. Technol.* ISSN 2157-6904. doi:<https://doi.org/10.1145/3655631>.
- Pagnucco M, Rajaratnam D, Limarga R, Nayak A, Song Y (2021). “Epistemic Reasoning for Machine Ethics with Situation Calculus.” In “Proceedings of the 2021 AAAI/ACM Conference on AI, Ethics, and Society,” AIES ’21, p. 814–821. Association for Computing Machinery, New York, NY, USA. ISBN 9781450384735. doi:<https://doi.org/10.1145/3461702.3462586>.
- Peschl M, Zgonnikov A, Oliehoek FA, Siebert LC (2022). “MORAL: Aligning AI with Human Norms through Multi-Objective Reinforced Active Learning.” In “Proceedings of the 21st International Conference on Autonomous Agents and Multiagent Systems,” AAMAS ’22, p. 1038–1046. International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC. ISBN 9781450392136. doi:<https://doi.org/10.48550/arXiv.2201.00012>.
- Rosen E, Hsiung E, Chi VB, Malle BF (2022). “Norm Learning with Reward Models from Instructive and Evaluative Feedback.” In “2022 31st IEEE International Conference on Robot and Human Interactive Communication (RO-MAN),” pp. 1634–1640. doi:<https://doi.org/10.1109/RO-MAN53752.2022.9900563>.
- Shea-Blymyer C, Abbas H (2021a). “Algorithmic Ethics: Formalization and Verification of Autonomous Vehicle Obligations.” *ACM Trans. Cyber-Phys. Syst.*, **5**(4). ISSN 2378-962X. doi:<https://doi.org/10.1145/3460975>.

- Shea-Blymyer C, Abbas H (2021b). “Learning a Robot’s Social Obligations from Comparisons of Observed Behavior.” In “2021 IEEE International Conference on Advanced Robotics and Its Social Impacts (ARSO),” pp. 15–20. doi:<https://doi.org/10.1109/ARSO51874.2021.9542846>.
- Sholla S, Mir RN, Chishti MA (2021). “A neuro fuzzy system for incorporating ethics in the internet of things.” *Journal of Ambient Intelligence and Humanized Computing*, **12**(1), 1487–1501. doi:<https://doi.org/10.1007/s12652-020-02217-2>.
- Stenseke J (2024). “Artificial virtuous agents in a multi-agent tragedy of the commons.” *AI & SOCIETY*, **39**(3), 855–872. doi:<https://doi.org/10.1007/s00146-022-01569-x>.
- Tassella M, Chaput R, Guillermin M (2023). “Artificial Moral Advisors: enhancing human ethical decision-making.” In “2023 IEEE International Symposium on Ethics in Engineering, Science, and Technology (ETHICS),” pp. 1–5. doi:<https://doi.org/10.1109/ETHICS57328.2023.10155026>.
- Wang N, Lin L, Li J, Wang H (2022). “Unbiased Graph Embedding with Biased Graph Observations.” In “Proceedings of the ACM Web Conference 2022,” WWW ’22, p. 1423–1433. Association for Computing Machinery, New York, NY, USA. ISBN 9781450390965. doi:<https://doi.org/10.1145/3485447.3512189>.
- Wu C, Wu F, Wang X, Huang Y, Xie X (2021). “Fairness-aware news recommendation with decomposed adversarial learning.” In “Proceedings of the AAAI conference on artificial intelligence,” volume 35, pp. 4462–4469. doi:<https://doi.org/10.1609/aaai.v35i5.16573>.
- Wu H, Ma C, Mitra B, Diaz F, Liu X (2022). “A Multi-Objective Optimization Framework for Multi-Stakeholder Fairness-Aware Recommendation.” *ACM Trans. Inf. Syst.*, **41**(2). ISSN 1046-8188. doi:<https://doi.org/10.1145/3564285>.