Mohsen Hassan Nejad

Towards an Ontology of Normative Role Design for LLM Agent Interactions in Multi-Agent Systems

Tallinn University · Tampere University · Lusófona University
Erasmus Mundus Programme in Artificial Intelligence for Sustainable Societies (AISS)
Master's Thesis – April 2026
Research page

Abstract

Mohsen Hassan Nejad

Towards an Ontology of Normative Role Design for LLM Agent Interactions in Multi-Agent Systems

Master's Thesis – Research Page: https://technejad.github.io/MA-thesis-ongoing/

Tallinn University, Tampere University, and Lusófona University

Master's Programme in Artificial Intelligence for Sustainable Societies (AISS)

Supervisor: José Braga de Vasconcelos, PhD, Associate Professor, Lusófona University

Co-supervisor: Danial Hooshyar, PhD, Research Professor, Tallinn University

Large Language Models are increasingly deployed as agents in multi-agent systems, where their behavior is shaped largely by the roles encoded in their system prompts. These roles carry normative content, including obligations, permissions, and prohibitions that govern how agents interact in a collective setting. Left unexamined, such design choices can quietly shape how agents behave. Yet no framework exists for identifying normative roles and mapping how they unfold across domains. This thesis develops a conceptual ontology for examining how such roles are designed, what normative content they carry, and what outcomes they produce. A systematic literature review of 724 papers yielded 40 studies, of which 38 were coded as the core of the ontology under a four-layer schema into 93 role conditions, 212 entities, and 516 relational triples. The results reveal partial normative content in most role designs, meaningful differences across simulation domains, and failure modes in nearly half of all role conditions. The thesis contributes (i) a reframing of system prompts as normative documents, (ii) a conceptual ontology of normative role design, and (iii) a reusable coding schema for analysing roles across multi-agent LLM studies. Together, they offer a basis for designing, evaluating, and governing normative role design for LLM agents.

The originality of this thesis has been checked using the Turnitin OriginalityCheck service.

Keywords: LLM agents, multi-agent systems, normative role design, ontology, systematic literature review, system prompts, AI ethics, AI governance

Use of AI in Thesis

I have utilised AI tools in my thesis: ☐ No ☒ Yes

The AI tools utilised in my thesis and their purposes are described below:

Names and versions of AI tools:

Anthropic Claude (Sonnet, Opus variants via Co-work and Claude Code CLI); OpenAI (ChatGPT, Codex); Google Gemini; Mistral AI; xAI Grok; Grammarly. Most of these tools were used with modified custom extensions, memory, and access to files and research context, rather than as generic chatbot versions.

Purpose of using AI tools:

AI tools were used in selected parts of the thesis process as support for review, coding, technical development, reference management, and writing improvement. In the systematic literature review, after the author completed screening and assessment independently, AI tools were used in a cross-checking capacity to flag possible oversights or inconsistencies in the application of exclusion criteria. In the ontology coding stage, they were used to help locate relevant system prompt-related evidence in the supplementary materials and code repositories of the studies; support the writing and editing consistency across coding analysis notes; and audit coding sheets for possible duplicates, missing evidence, or inconsistencies in coded attributes, triples, and supporting evidence. AI software development tools were also used in the development of the thesis research page (https://technejad.github.io/MA-thesis-ongoing/), including an interactive knowledge graph based on the ontology data. During the writing process, AI tools were used for editorial support, including rewriting for clarity, improving structure and readability, checking grammar and phrasing, identifying possible logical gaps, and helping review citation consistency, source use, and overall coherence.

Sections where AI tools were used:

Systematic literature review; ontology coding and evidence audit; development of the thesis research page and interactive knowledge graph; editorial revision of the thesis text; and reference management.

I acknowledge that I am fully responsible for the entire content of my thesis, including the parts generated by AI, and accept accountability for any violations of ethical standards in publications.

Table of Contents

List of Tables

List of Figures

List of Abbreviations

CQ	Competency Question
EC	Exclusion Criterion
IC	Inclusion Criterion
IRR	Inter-Rater Reliability
LLM	Large Language Model
MAS	Multi-Agent System
NI	Normative Indicator
PRISMA	Preferred Reporting Items for Systematic Reviews and Meta-Analyses
QA	Quality Assessment
RC	Role Condition
RLHF	Reinforcement Learning from Human Feedback
RQ	Research Question
SLR	Systematic Literature Review

Glossary of Key Terms

Competency question (CQ) — A question the ontology must be able to answer. Competency questions serve as design targets during ontology construction and as validation criteria for the completed ontology.

Conceptual ontology — A structured vocabulary of concepts, entities, relations, and rules that organises knowledge within a domain for human interpretation. Distinguished in this thesis from formal ontologies, which are logic-based and machine-executable.

Deontic — Relating to obligation, permission, and prohibition. In normative theory and multi-agent systems, deontic logic formalises what agents must, may, or must not do.

Failure mode — An unintended breakdown in which an agent's behavior deviates from the normative expectations embedded in its role design. Organised in the ontology across three layers: substrate (model-level), design (prompt-level), and interaction (dynamic).

Incentive structure — A prompting method that operationalises a role through payoffs, penalties, and strategic objectives, grounding behavioral expectations in game mechanics rather than identity.

LLM agent — A language-model-based agent: a system in which a large language model, coordinated by an orchestrator and optionally equipped with memory, planning, and external tools, is prompted to play a specified role and interact with other agents or humans.

Multi-agent system (MAS) — An environment in which two or more autonomous agents interact—coordinating, competing, or cooperating—toward individual or shared objectives. LLM-based MAS replaces the fixed symbolic specifications of classical MAS with natural-language role prompts.

Norm origin — Whether a role's norms are designer-specified (norm_designed), arise through agent interaction (norm_emergent), or combine both (norm_hybrid).

Normative frame — The type of normative logic that organises a role's expectations. The ontology identifies nine frames, including ethical (deontological, consequentialist, virtue), institutional, social convention, strategic-instrumental, value-oriented, and ideological framings.

Normative indicator (NI) — One of five binary features used to score the normative content of a role condition: social reference, prescriptive framing, violability, conditionality, and social standing. Their sum (the NI Count, 0–5) classifies a role condition as FUNCTIONAL, PARTIAL, or FULL normative.

Normative role — A role specification that goes beyond functional duties by embedding behavioral and social expectations about how the agent should act toward other agents or humans.

Persona framing — A prompting method that constructs an agent's identity—name, occupation, personality, social context—and relies on the model to infer behavioral expectations from that identity.

Prompting method — The technique used to operationalise a normative role within a system prompt. The ontology identifies eleven methods, including persona framing, incentive structure, principle statement, narrative priming, and cultural transmission.

Role condition (RC) — The unit of analysis used in this thesis. An RC is any distinct agent type, persona, or normative framing that receives its own prompt or behavioral specification within a study. A single paper can contain multiple role conditions.

Role type — The normative role assigned to, or emerging in, an agent (for example, situated_persona, adversarial, game_strategic, prosocial). The ontology catalogues role types as a domain-general vocabulary.

Simulation domain — The context in which a multi-agent simulation takes place (for example, negotiation, tragedy of the commons, debate, peer review). Coded at the study level.

System prompt — The natural-language instruction layer that tells an LLM what behavior, perspective, and approach to adopt in a given context. The primary artefact through which roles are encoded and, in this thesis, the primary empirical evidence analysed.

Triple — The atomic unit of the ontology: a statement of the form subject → relation → object (for example, *role_type_antagonistic → grounded_in → norm_frame_social_convention*).

Introduction

Large language models (LLMs) are no longer passive text generators. They are active agents that reason, plan, and interact with other agents and humans in large-scale systems (Xi et al., 2025). Their ability to interpret natural-language instructions and generalise across tasks has made them attractive vehicles for multi-agent simulation, where behavior and system-level interaction can be studied in domains such as negotiation, resource sharing, and strategic game play (Lu et al., 2024).

A widely used approach to building LLM agents is to assign them roles through a system prompt: natural language instructions that encode the agent's identity and govern its conduct in a given context (Chen et al., 2024; Tseng et al., 2024). A foundational language model has no default persona; it is a simulator that can take on any of the roles it has absorbed from its training data, and the system prompt suggests which one (Shanahan et al., 2023).

A role gives the agent more than just functional duties by defining behavioral and social expectations. For example, in a negotiation study by Bianchi et al. (2024), a single line added to the system prompt ("You must fake being desperate. Supplicate and beg to get more resources") caused agents to improve their payoffs by 20% through faked emotional appeals. The desperate negotiator is one of many such role conditions this thesis catalogues and analyses in a variety of simulation settings.

In social science, norms are the shared expectations, such as obligations, permissions, and prohibitions, that govern how members of a community should act toward one another (Bicchieri, 2006). When a system prompt prescribes what an agent must do, how it should treat other agents, and what counts as a violation, it is scripting a similar kind of normative structure.

As LLM agents move from simulations into high-stakes domains and real-world multi-agent systems, their interactions with other agents and humans introduce novel risks alongside new capabilities (Hammond et al., 2025). How we design roles for agents and position them in the real world shapes their societal impact, and although the research on role design is growing, it remains disorganized, lacking a systematic taxonomy (Tseng et al., 2024). Prompt engineering more broadly, as an agent building practice among developers and researchers, is guided by

trial-and-error rather than coherent principles (Liang et al., 2025). What is missing is a structured account of the normative content these roles carry: what behavioral expectations they encode, what obligations they distribute, and what consequences they produce across settings.

To address these gaps, this thesis develops a prototype ontology of normative role design (§1.3 and §2.2), a structured knowledge base mapping the concepts, entities, relations, and rules that guide this field. The core research question is: How are normative roles designed and operationalized across different applications of multi-agent LLM systems? Sub-questions, set out in §2.1, address what normative frames and prompting methods designers use, what outcomes those designs produce, where they fail, and what impact the domain they operate in has.

To answer these questions, a systematic literature review was first conducted to assemble a corpus of relevant simulation studies, which was then analysed through an extensive coding process that produced the ontology's core. The overall workflow is visualised in Figure 1.

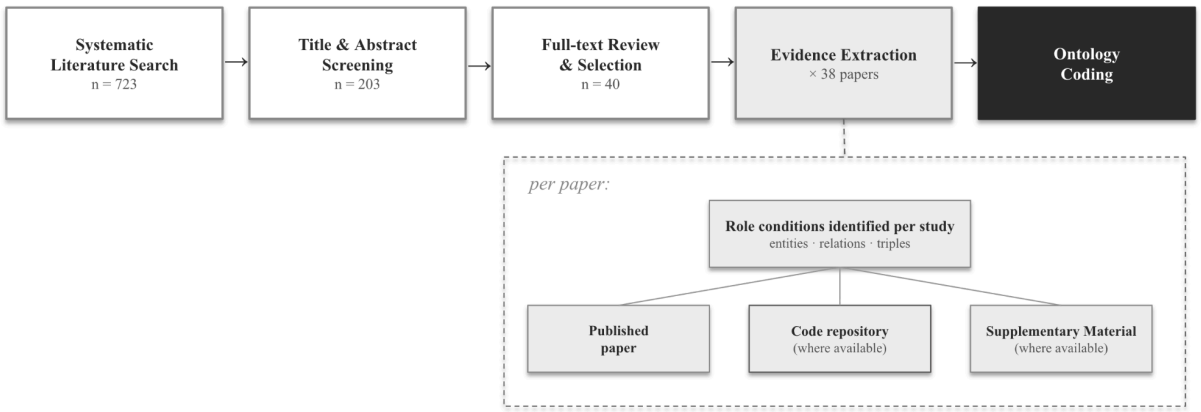


Figure 1. *Research pipeline from systematic literature review to ontology coding.*

This thesis makes three contributions. (i) A conceptual ontology of normative role design that maps how roles are specified in multi-agent LLM systems, how they shape outcomes, and where they fail. (ii) A reframing that places system prompts within normative theory, treating them as scripts that encode behavioral expectations, distribute obligations, and carry consequences. This reframing bridges the engineering of multi-agent LLM systems and the study of norms in philosophy and social science. (iii) A four-layer coding schema and a set of competency questions (§2.2) that offer a reusable protocol for analysing normative role design, applicable beyond this corpus to new studies, deployed systems, and prompt libraries.

The analysis yields five findings that anchor the chapters to come. Across 93 role conditions in 38 studies, role design is widespread, but the normative vocabulary that grounds it is narrow, dominated by strategic-instrumental and social-convention framings while moral philosophy's broader repertoire remains underutilized. Behavior is rarely a function of the role alone; it emerges from the joint action of role, prompting method, model capability, domain, and interaction, so identical roles can diverge when any one of these shifts. Nearly half (45%) produce failure modes, from model-capacity limits that prompting cannot address to emergent dynamics in which one agent reshapes the normative environment for the others. And the domain is not a passive stage but an active shaper of which normative tools carry weight.

Three commitments define the scope of this inquiry: language-based agents, multi-agent configurations, and roles specified primarily through system prompts. Language is the medium through which LLM agents interpret and enact their roles. System prompts are where those roles are encoded in practice. And multi-agent settings are the conditions under which norms and interaction dynamics become observable. Single-agent deployments cannot exhibit the emergent behavior this thesis sets out to study. Further inclusion and exclusion criteria are set out in §2 as part of the systematic literature review protocol.

The remainder of this thesis is organized as follows. Chapter 1 reviews the literature on LLM agents and multi-agent systems, the theoretical foundations of norms in agent systems, and the ontology approach. Chapter 2 describes the methodology in two phases: the systematic literature review used to build the empirical corpus, and the ontology development approach. Chapter 3 presents the results across the five research questions. Chapter 4 discusses the findings in relation to the research questions, alongside implications and limitations. Chapter 5 concludes with contributions and directions for future work.

1. Literature Review

This chapter reviews the foundations of the thesis across three domains. Section 1.1 introduces LLM agents and their deployment in multi-agent systems. Section 1.2 examines the theoretical underpinnings of norms: what makes a norm a norm, how norms have been formalized in classical agent systems, and how they might apply to LLM agents. Section 1.3 reviews the ontology approach as a method for structuring and mapping the space of normative role design.

1.1 LLM Agents and Multi-Agent Systems

1.1.1 From Language Models to LLM Agents

Foundational Large Language Models (LLMs) that were initially developed as word predictors (Devlin et al., 2019; Brown et al., 2020) have evolved into agents capable of multi-step reasoning (Wei et al., 2022), tool use (Schick et al., 2023), and autonomous task execution (Wang et al., 2024). State-of-the-art LLM architecture typically shares four core modules: a profile module (role and identity), a memory module (context over time), a planning module (task decomposition and reasoning), and an action module (outputs and tool calls) (Xi et al., 2025; Wang et al., 2024). Figure 2 presents the package that makes up an LLM agent.

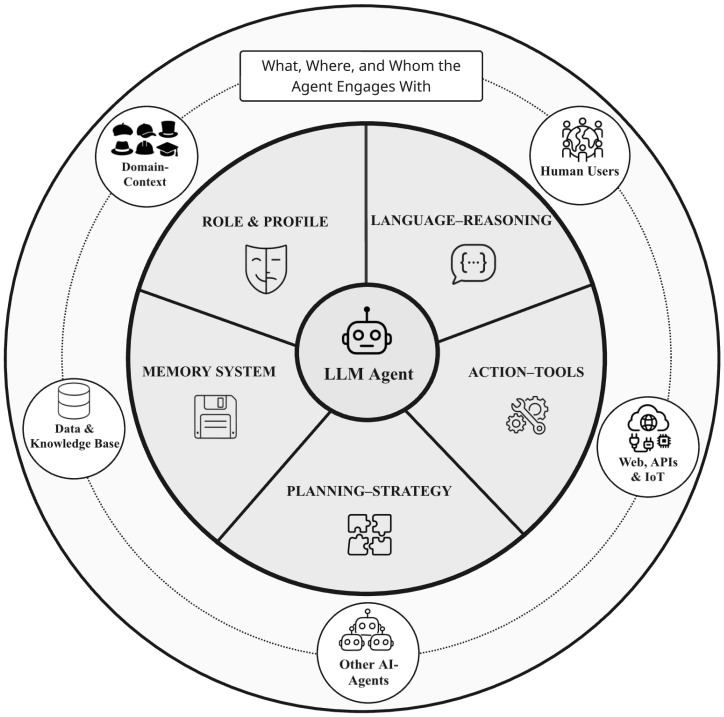


Figure 2. *The Anatomy of an LLM Agent*

Figure 2 offers a simplified view of that architecture. An LLM agent is a layered and dynamic system. Its components are coordinated by an orchestrator: conventional software that connects the language model with tools and memory, executes code or API calls on its behalf, and feeds results back to it for reasoning and planning. Actions may involve multiple tools, roles adapt to context, and memory can take different forms. This modular setup makes LLM agents flexible but also unpredictable. What's more, these agents are now deployed not in isolation but together, in multi-agent simulations designed to observe how they behave toward one another (Park et al., 2023), and in real-world multi-agent systems to harness their collective intelligence (Hammond et al., 2025).

1.1.2 Multi-Agent Systems

A multi-agent system is any environment in which multiple autonomous agents interact: coordinate, compete, and work toward shared objectives. The concept predates LLMs by decades, rooted in classical AI research on distributed problem-solving and game theory, but the arrival of LLM agents has transformed what these systems can do.

Classical multi-agent systems research built agents with explicitly defined states, utility functions, and communication protocols; coordination and roles were typically formalized through logic-based specifications or organizational models, which made agent behavior mostly systematic and predictable (Russell & Norvig, 2021; Dignum, 2004).

LLM-based MAS depart from this paradigm in two main ways. First, decision-making operates over natural language rather than fixed symbolic vocabularies. Second, roles and norms are less formalized, making agents more flexible but harder to analyze and more sensitive to subtle variations in the instructions they receive (Xi et al., 2025). Lu et al. (2024) describe this shift as the rise of Generative Agent-Based Models, in which agent behavior is no longer derived from fixed rules but from prompts encoding social roles, personality traits, and contextual information.

The unpredictable and adaptive nature of LLMs makes them an appealing proxy for humans. Recent experiments have used multi-agent simulations to reproduce classical social-science phenomena such as cooperation and collapse in commons dilemmas, repeated games, opinion dynamics, and strategic behavior in negotiations (Bianchi et al., 2024; Piatti et al., 2024; Akata et al., 2025; Chuang et al., 2024). The diversity of this field is visible in Gao et al.'s (2024) survey,

which catalogues LLM-empowered simulations across social, physical, cyber, and hybrid domains, each posing distinct requirements for how agents are deployed and how they interact.

Emerging safety literature highlights that interactions among LLM-agents introduce novel systemic risks, miscoordination, conflict, and collusion beyond single-agent alignment. These risks are amplified by information asymmetries, network effects, and emergent agency (Hammond et al., 2025). Rigorous analysis of how norms and roles are specified and built into these autonomous entities is one critical pathway into AI safety and governance.

1.1.3 Role Design as a Central Concern

LLM agents are operationalized in context through system prompts: natural language instructions that specify what role the agent plays, how it should reason, what to prioritise, and how to engage with humans or other agents. This marks a fundamental departure from traditional software engineering. Emerging evidence suggests that roles structure what agents attend to, how they interpret situations, and how they balance objectives against constraints (Tseng et al., 2024; Xi et al., 2025).

A foundational language model has no personality; it is a prediction engine capable of simulating the many characters it has seen in the vast corpora of human text it is trained on. In that sense, the model generates responses consistent with the identity, goals, and constraints the script describes. The role is therefore not decorative; it is the mechanism that enables a generic model to play the part of a particular agent in a given context. Shanahan et al. (2023) call the base model a simulator, a system with no beliefs or goals of its own, and argue that role-play is the LLM's native mode of operation, or as they put it: *"With a dialogue agent, it is role-play all the way down."*

At the implementation level, Chen et al. (2024) describe prompt-based role specification as consisting of persona data and role-playing instructions. Persona data has two elements: descriptions, which specify the agent's attributes such as name, background, personality, and tone; and demonstrations, which illustrate representative behavior through example dialogues or interactions. Role-playing instructions are treated as a separate element that encourages or restricts specific behaviors.

Role design has measurable behavioral consequences across the literature. For example, in game-theoretic experiments across four LLMs and five languages, Buscemi et al. (2025) find that a single label ("cooperative" or "selfish") in the agents' system prompt shifts strategic choices. Orner et al. (2025) observe similar effects with richer personas: swapping a politician for a con-artist role changes cooperation dynamics in a mixed-motive game. Großmann et al. (2026) push the mechanism further still, showing that agents whose system prompts include cooperation-themed folktales give significantly more in a networked public goods game than agents who receive no story or a nonsensical one.

Role design also has various failure modes. Tseng et al. (2024) document that role assignment produces not only the intended task behavior but also emergent effects such as conformity, toxicity amplification, and shifts in social disposition that the designer may not have anticipated. Beyond unintended side effects, role conditions can fail as designed. Li et al. (2023), creators of the CAMEL role-playing framework, document characteristic breakdowns such as role flipping, instruction repetition, and conversational collapse, in which the assigned role fails to hold across the simulation.

Gao et al. (2024), surveying LLM-empowered agent-based simulations, highlight a core challenge: while LLMs make it easy to generate heterogeneous agents, ensuring each reliably fulfills its assigned role remains difficult. Li and Wu (2025) map this challenge on a spectrum of prompt design, from descriptive prompts, where agents infer behavior from abstract identities, to instructional prompts that explicitly define decision rules, constraints, and reasoning. Descriptive prompts grant agents autonomy but risk inconsistent behavior and low internal validity; instructional prompts, in contrast, enhance control but risk over-determining outcomes, a tension Li and Wu term the “over-control” problem.

Simulation studies are beginning to pay attention to the importance of this space, but the field has no shared vocabulary for describing what the different role conditions contain, how they produce their effects, and where they break down (Tseng et al., 2024; Liang et al., 2025). The next section turns to the dimension of role design that this thesis foregrounds: the normative content embedded in those specifications.

1.2 Norms in Agent Systems

Most social behavior runs on scripts, socially learned sequences of action that guide how we enter a classroom, order coffee, or grieve at a funeral. As Bicchieri (2006) argues, these scripts form the backbone of normative behavior: they shape not only what we do but how we expect and interpret what others do. A system prompt given to an LLM agent appears to perform a similar function: it names a situation, assigns a part, and embeds implicit prescriptions about an appropriate response. This section begins from normative theory itself, from what norms are, how they differ, and what kinds of authority they carry (§1.2.1). It then traces how the multi-agent systems field has tried to encode these ideas into computational agents, from deontic logic to organisational frameworks (§1.2.2), and how the transformation towards LLM-based agents relocates where norms live in the system (§1.2.3). Finally, it synthesises these threads into five heuristic indicators for identifying normative content in agent role conditions (§1.2.4).

1.2.1 What Makes a Norm a Norm

Weber (1922/1978) established that action qualifies as 'social' only insofar as it takes account of the behavior of others. His distinction between value-rational action (commitment to shared values) and instrumental-rational action (optimising means to ends) informs a central concern for this thesis: norm-following involves orientation toward shared expectations and standards, not merely individual optimisation.

Bicchieri (2006) confirms this by observing that conformity to norms is conditional on beliefs about what others do and expect: a social norm exists when individuals believe most others follow the behavior (empirical expectations), and when individuals believe others expect them to follow it (normative expectations).

Searle (1995) adds a distinction between regulative rules ('Do X') that constrain pre-existing activities, and constitutive rules ('X counts as Y in context C') that create new institutional possibilities. Institutional rules carry deontic force not from ethical reasoning or social habit but from organisational or regulatory authority, what Searle (1995) calls institutional facts, created and sustained by collective agreement.

Norms also differ in the kind of authority they carry. Social norms, unlike moral norms, regulate interaction through mutual expectations rather than moral authority. Brennan et al. (2013) argue that moral norms appeal to considerations of harm, fairness, and rights, while social norms rest on coordination and mutual expectation alone.

These distinctions give us a working vocabulary of norms: what makes them normative, how they vary, and what kinds of authority they carry. What they do not yet tell us is how such norms can be transmitted to, or arise within, an agent that has no human-like social standing or moral awareness.

1.2.2 Norms in Computational Agent Design

Translating normative scripts into useful digital systems has been a long-running problem in computer science. Autonomous software agents do not share our history, our embarrassments, or our stake in being well regarded, yet we have increasingly asked them to operate in settings where those things matter. In doing so, the field has reached across disciplines from moral philosophy to social theory, legal institutionalism, and social psychology, in pursuit of assembling a working repertoire of frames for encoding normative content into agent behavior.

The MAS tradition's first answer to this problem was formal. For computer agents to coordinate in a shared environment, designers needed a precise way to specify what each one was required to do, allowed to do, and forbidden from doing. The tools came from deontic logic: the branch of formal reasoning concerned with obligation, permission, and prohibition. How to translate these ideas into working agents, however, split the field. Jones and Sergot (1993) argued that computer systems should be modelled as norm-governed rather than rule-bound: using the deontic operators of obligation, permission, and prohibition (O, P, F), they insisted on a gap between *actuality* (what is) and *ideality* (what ought to be). Norms, on their view, are precisely the kinds of things that can be violated; a system that forecloses violation is not regulating behavior but eliminating it. This is where the field's distinction between *regimentation* (making violation impossible) and *enforcement* (allowing violation, with consequences) crystallised.

Shoham and Tennenholtz (1995) took the opposing view, defining social laws as hard constraints on the action space. A cleaner engineering solution, but one that eliminates meaningful choice for the agents cannot violate, rather than should not.

A decade later, the OMNI framework (Dignum et al., 2005) tried to reconcile these traditions, combining deontic rules with organisational roles and a shared vocabulary for coordination, while preserving autonomy by keeping an organisation's values separate from the rules and code that implemented them.

Underlying these formal traditions is a more basic distinction. Haynes et al. (2017), in their review of norm engineering in multi-agent systems, distinguish conventions from norms. A convention is a stable behavioral pattern within a society: it describes what agents do, but carries no obligation and no sanction for deviation. A norm, by contrast, imposes an obligation to act or refrain from acting in a particular way, and its violation risks punishment. The boundary is not always sharp: conventions can evolve into norms when a community begins to sanction non-conformity, turning a behavioral regularity into a social expectation. This gradient between convention and norm provides a useful lens for examining the normative content of agent role specifications, where assignments range from purely functional task descriptions to instructions that embed explicit obligations and consequences.

How to enforce norms is one challenge; the grounding frameworks used are another. Woodgate and Ajmeri (2024) identify three main philosophical traditions that have been operationalised in AI ethics: (1) Deontological principles, which evaluate actions by whether they conform to rules, duties, and rights. (2) Consequentialist principles, which judge actions by their outcomes. (3) Virtue ethics, which locates moral worth in the character of the agent rather than in the act or its consequences. These three traditions served as the initial seed for the normative frame dimension of the ontology developed in this thesis (§2.2.6), though the coding process revealed that the normative vocabulary of existing role designs extends well beyond them (§3.3.2).

Cutting across all of these is the distinction between descriptive and injunctive norms (Cialdini et al., 1991). Descriptive norms reflect what most people actually do in a given situation, such as tipping. Injunctive norms prescribe what people ought to do, such as not smoking indoors. The key difference is in what happens when someone deviates. Breaking a descriptive norm is merely

unusual, while breaking an injunctive norm draws social disapproval. Ren et al. (2024) confirm the importance of this distinction in generative agent societies, where injunctive norms spread within the first simulated day while the descriptive norm took twice as long, precisely because its violations triggered weaker corrective social pressure from other agents.

1.2.3 Norms in Language Models

As we enter the era of powerful language models, the foundation has shifted. Norms no longer have to be written as deontic formulae over finite action spaces; they can be expressed in the same natural language used by humans, and interpreted by a model whose behavior was trained rather than fully engineered or specified. This change relocated where norms live in the system.

Current approaches embed norms in LLM agents at three levels. At the *training level*, Constitutional AI trains models on natural-language principles that encode normative preferences, including constraints on harmful outputs (Bai et al., 2022). At the *inference level*, system prompts specify role boundaries, behavioral guidelines, and interaction protocols when the model generates responses. At the *emergent level*, norms can arise from agent interaction itself: Park et al. (2023) showed generative agents developing social behaviors from minimal specifications, and Chuang et al. (2024) demonstrated that LLM populations can spontaneously develop shared conventions through repeated interaction. Furthermore, Ren et al. (2024) propose CRSEC, a normative architecture for generative agent societies, and demonstrate that LLM agents can spontaneously develop and internalise social norms through conversation and observation, including norms not present in any agent's initial description.

These insights complicate a purely top-down account. Role design sets the initial normative conditions of a multi-agent system, but interaction can generate norms that no designer scripted. In classical MAS, this capacity for bottom-up norm emergence was a central research concern (Haynes et al., 2017). Before tracing those dynamics further, however, a prior question must be settled: how to tell whether a role specification carries normative content at all. The next section introduces five heuristic indicators for identifying when a role specification carries normative content.

1.2.4 Normative Indicators

Not all role conditions in multi-agent systems are normative in the same way. Some simply assign tasks (e.g., "maximize profit" or "collect resources"). Others embed expectations about how agents ought to behave toward one another. To distinguish these cases, this thesis identifies five components that could help us identify a normative role. Drawing on theoretical frameworks described in the previous sections, these indicators attempt to capture the range with which normative expectations appear in agent instructions:

I. Social reference indicates whether a role frames behavior in relation to other agents or shared standards. Normative behavior, in Weber's sense, is oriented toward others. Prompts such as "consider what would happen if everyone acted this way" explicitly invoke such collective standards.

II. Prescriptive framing signals that the role guides behavior rather than merely describing a task. Normative roles typically use expectation-laden language such as "should," "must," or "are expected to."

III. Violability captures whether agents retain meaningful freedom within the role. Norms guide behavior but do not eliminate choice; agents can comply or deviate (Shoham & Tennenholtz, 1995; Dignum et al., 2005).

IV. Conditionality reflects Bicchieri's (2006) insight that norms often depend on beliefs about others' behavior. In simulations, this appears when agents respond to reciprocity, reputation, or actions of the other agents.

V. Social standing implications refer to consequences for violating expectations, such as reputation loss, sanctions, or relational penalties within the simulated community.

These indicators do not define norms in a strict sense. Instead, they provide an analytical lens for identifying and comparing normative role designs across studies. Roles may display some indicators but not others, resulting in varying degrees of normative orientation. The methodology section §2.2.4 describes how these indicators are operationalised in the context of the ontology coding framework. Identifying normative content is one challenge; the next one is organising the

full design space of roles, norms, and outcomes into a systematic framework. That particular challenge requires a tool — an ontology.

1.3 The Ontology Approach

Normative role design in LLM-based multi-agent systems is important but under-studied and fragmented (§1.1), while embedding social norms into artificial agents remains a long-standing challenge with a rich but heterogeneous vocabulary (§1.2). Bridging the two fields of multi-agent AI and normative theory requires a tool that can help us identify, organise, and compare knowledge. The ontology serves that purpose.

An ontology is an explicit specification of a conceptualization (Gruber, 1995), a formal account of the terms and relations among them that defines a common vocabulary for sharing information in a domain (Noy & McGuinness, 2001; Uschold & Grüninger, 1996). By making knowledge explicit, ontologies enable better organization, comparison, and communication in different contexts (Patel & Debnath, 2024). A standard way to represent the knowledge an ontology captures is through semantic triples: atomic statements of the form subject–relation–object, where two entities are connected by a named relation. The concept originates in knowledge representation and the Semantic Web, where the Resource Description Framework (RDF) encodes all information as such triples, forming directed labelled graphs that can be queried, linked, and extended (Hogan et al., 2021).

Ontologies range along a spectrum of formality, from informal vocabularies that organise and clarify knowledge for human interpretation, through to logic-based frameworks encoded in languages like OWL that enable computational reasoning and interoperability (Guarino et al., 2009; Uschold & Grüninger, 1996). This thesis operates at the conceptual end of that spectrum: the ontology is built for human analysis and cross-study comparison, not for automated inference. However, by representing its knowledge as semantic triples of the form subject–relation–object (Hogan et al., 2021), it adopts a formal representational structure that could serve as the foundation for a machine-readable knowledge base in future work.

Ontologies have been used in the past to define agent roles in rule-based multi-agent systems (MAS), specifying how agents communicate, act, and fulfill their responsibilities. Frameworks

like OMNI, for example, demonstrated that roles could be defined by a set of objectives, rights, and norms. (Dignum et al., 2005). This could serve as a foundation for building ontologies for LLM role design, even as the context shifts from rule-based to generative agents.

Recent advances in LLMs have created two key junctions with ontologies. The first uses LLMs to assist ontology construction (LLMs for Ontologies): LLMs can accelerate development but produce inconsistencies that require human oversight (Saeedizade & Blomqvist, 2024). The second uses ontologies to support LLM applications (Ontologies for LLMs): structured ontological knowledge can ground retrieval and reduce hallucination in LLM-based systems (DeBellis et al., 2024). Underlying both junctions is a fundamental asymmetry: LLMs navigate meaning probabilistically, whereas ontologies require explicit, unambiguous definitions (Neuhaus, 2023). The two approaches are thus complementary rather than competing; ontologies provide the kind of structured, inspectable knowledge that LLMs alone cannot guarantee.

This work applies the ontology principle to systematically map the design space for normative roles in multi-agent LLM simulations. The goal is a knowledge graph: a network of entities connected by labelled relations, expressed as triples for comparing and evaluating normative role designs across studies.

1.4 Synthesis

Role design is a key mechanism by which LLM systems are instantiated (§1.1), and those roles frequently carry normative content that is challenging to implement and evaluate (§1.2). The field remains, in practice, behaviorally rich but normatively underspecified: researchers study what effects roles produce, but no systematic account maps what normative commitments they embed or how those commitments interact with context to shape outcomes. An ontology can close that gap by making design choices explicit, comparable, and analytically tractable (§1.3).

The ontology is therefore built as a diagnostic tool to make visible what current practice leaves implicit: what types of normative roles appear across the corpus, how they are operationalised through system prompts, what normative frames they invoke, how they translate into behavior and collective outcomes, and where they fail. These are the questions that structure the remainder of this thesis.

The methodology for constructing this ontology, including the systematic literature review that assembled the corpus, and the coding approach that built the ontology from it, is described in detail in Chapter 2.

2. Methodology

The research methodology follows a two-phase logic. First, a systematic literature review (SLR) identifies, assesses, and selects the corpus of multi-agent LLM simulation studies. Second, the selected studies are coded and formalized into a conceptual ontology of normative role design. The ultimate goal of this dual process is to answer the main research question: How are normative roles designed and operationalized across different applications of multi-agent LLM systems? Which is further detailed into five sub-questions:

RQ1: What types of normative roles are designed across multi-agent LLM systems?
RQ2: What prompting methods and normative framings operationalize these roles?
RQ3: How do the designed roles relate to individual agent behavior and collective simulation outcomes?
RQ4: Where and how do the designed roles fail to produce intended behavior and outcomes?
RQ5: How does the simulation domain mediate the normative design of agent roles?

This chapter describes both phases in detail.

2.1 Systematic Literature Review (SLR)

The purpose of this SLR is not to produce a standalone review article, but to systematically identify, assess, and select the corpus of multi-agent LLM simulation studies from which the ontology of normative role design is constructed. The review is therefore instrumental: it ensures that the studies forming the basis of the ontology are selected through a rigorous and transparent process. The SLR follows established guidelines for systematic reviews in software engineering and information systems (Kitchenham et al., 2015; Tranfield et al., 2003). The following subsections discuss every stage of the process.

2.1.1 Search Strategy

Two electronic databases were selected for the systematic search: Scopus, for its comprehensive coverage of CS/AI venues including major conferences (NeurIPS, ICML, AAAI, ICLR); and Web of Science, for its established role in systematic reviews and its cross-disciplinary citation coverage. The search period was defined as 2017–January 2026. The starting year was selected to coincide with the introduction of the Transformer architecture (Vaswani et al., 2017), which serves as the foundational technology for modern LLMs. The final search was executed in January 2026. Search terms were derived from the conceptual chain, research questions, and iterative testing against known relevant studies. Table 1 presents the term clusters used.

Table 1. *Search term clusters.*

Category	Primary Terms	Alternatives
Core Technology	“large language model”, “LLM”	“language model”, “GPT”
Agent Framing	“agent”, “LLM agent”, “multi-agent.”	“generative agent”, “autonomous agent”, “agency”
Role Design	“role”, “persona”	“profile”, “character”, “social behavior”
Domain / Setting	“simulation”, “game”, “negotiation”	“cooperation”, “collaboration”, “competition”
Normative Content	“normative”, “ethical”	“moral”, “norm”

The following Boolean search string was applied across both databases, adapted to each platform's query syntax (full queries are provided in Appendix 3):

(LLM OR "language model*") AND (agent*) AND (simulation* OR game*) AND (behavio*r* OR social OR interaction* OR "agency")*

Results were filtered to English-language publications from 2017 to 2026. Scopus returned 714 records and Web of Science returned 342 records, yielding a combined total of 1,056 records imported into Zotero 7, a reference management tool, for deduplication. After deduplication, the library was reduced to 724 unique records.

2.1.2 Screening and Quality Assessment

Screening proceeded in three stages: (1) title and abstract screening, (2) full-text screening, and (3) quality assessment, reducing the corpus from 724 records to 40 at the end of the SLR process. Figure 3 provides an overview of the complete review flow.

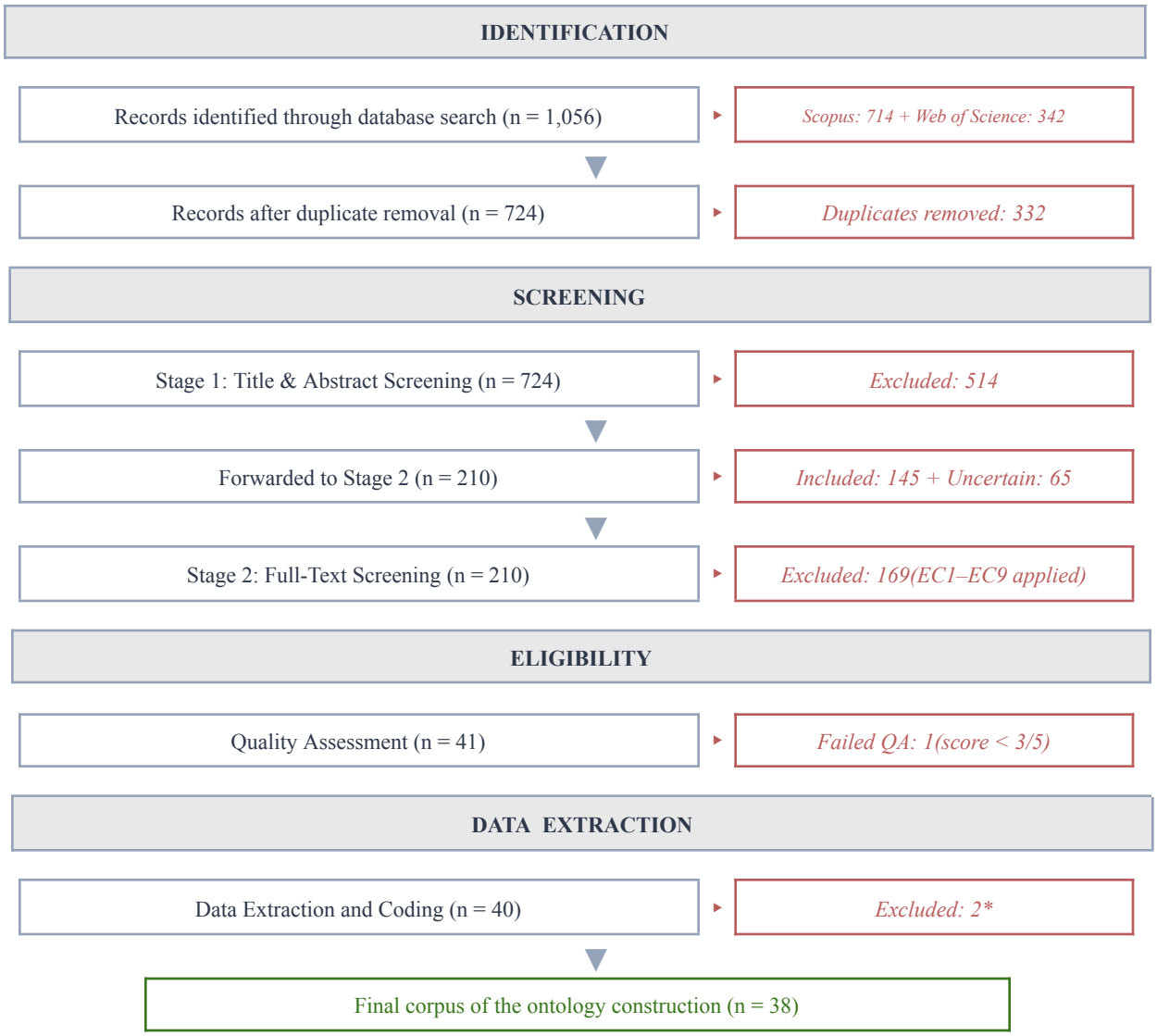


Figure 3. *PRISMA-style SLR flow summary. As the final output, 40 studies passed quality assessment. *2 studies were excluded during data extraction for ontology; details discussed in §2.2.*

Stage 1: Title and Abstract Screening. The goal was to remove clearly irrelevant papers based on title, abstract, and metadata. Each paper was scored on a three-point scale: 0 = exclude (clearly out of scope), 0.5 = uncertain (possible relevance, requires full-text review), 1 = include (clearly relevant). All papers scoring 0.5 or 1.0 were forwarded to Stage 2. Of 724 records screened, 514 were excluded, 65 were uncertain, and 145 were included, yielding 210 papers forwarded to Stage 2.

Stage 2: Full-Text Screening. Each paper was assessed following a stop-at-first-violation logic: the first applicable exclusion criterion was recorded as the reason for exclusion. EC7–EC9 were applied exclusively at this stage, requiring access to full text, system prompts, and repositories. Of 210 papers reviewed, 169 were excluded, and 41 were forwarded to quality assessment. Figure 4 shows the distribution of exclusion reasons across the six criteria.

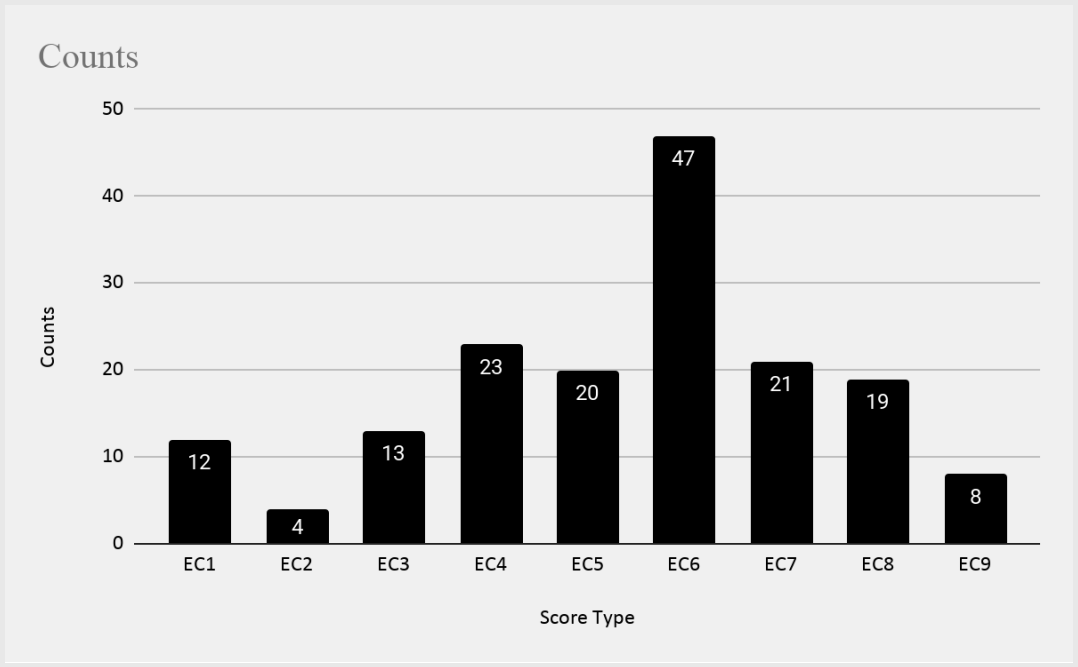


Figure 4. *Distribution of Stage 2 exclusion reasons by criterion.*

Table 2 lists the inclusion criteria, and Table 3 lists the exclusion criteria.

Table 2. *SLR inclusion criteria.*

ID	Category	Description	Stage
IC1	Technology	Agents are architecturally driven by Large Language Models (LLMs).	Query + Stage 1
IC2	Architecture	The system features multi-agent dynamics (2+ autonomous LLM agents).	Query + Stage 1
IC3	Context	Agents directly interact within a shared environment, game, or interactive scenario.	Query + Stage 1
IC4	Outcomes	The study analyses how LLM agents make decisions, interact, and produce collective outcomes.	Query + Stage 1
IC5	Timeframe	Publication date from 2017 to 2026.	Query filter
IC6	Language	Full text is available in English.	Query filter

Table 3. *SLR exclusion criteria.*

ID	Category	Description	Stage
EC1	Publication Type	Surveys, review papers, editorials, workshop abstracts, posters, or proceedings volumes.	Stage 1 & 2
EC2	Conceptual	Purely conceptual or theoretical studies that do not conduct a simulation experiment.	Stage 1 & 2
EC3	Agent Type	Non-LLM agents (rule-based, symbolic AI, pure RL).	Stage 1 & 2
EC4	MAS	Single LLM agent or human–LLM interactions without multi-agent system dynamics.	Stage 1 & 2
EC5	Interaction Type	Agents interact indirectly through intermediary mechanisms (market prices, mathematical functions, system aggregates, automated matching, order books) with no direct communication or interaction.	Stage 1 & 2
EC6	Object of Study	The MAS is merely a technical tool for benchmarking, optimisation, or generating synthetic data.	Stage 1 & 2
EC7	Prompt Access	Studies that do not provide the full system prompt, or an equivalent complete description (e.g., in an appendix or a public repository), prevent direct analysis of role design as an object of study	Stage 2
EC8	Normative Roles	Roles are functional/task-based with no reference to social expectations, obligations, or behavioral demands toward other agents.	Stage 2
EC9	Outcome Analysis	The study does not analyse how roles relate to individual agent behavior and collective outcomes.	Stage 2

EC1 restricts the corpus to primary research, excluding surveys, reviews, and editorials. EC2–EC4 build towards the topic: the review targets experimental studies deploying LLM agents in multi-agent simulations. EC5 requires direct interaction because normative roles orient agents toward one another; if agents interact only through market signals or system aggregates, there is no observable social behavior to analyze normative role design from.

EC6 distinguishes studies that treat multi-agent systems as objects of behavioral inquiry from those that use them as instruments for generating synthetic data, benchmarking models, or optimizing outputs. This review retains only the former, because the thesis examines how role design shapes what agents do, following Rahwan et al. (2019), who argue that AI systems can and should be studied as behavioral subjects in their own right.

EC7 requires access to the full system prompt or equivalent, because prompts are the primary evidence for identifying role conditions as the unit of analysis for ontology construction; without them, role design cannot be examined directly. EC8 excludes studies

whose roles are entirely functional, carrying no reference to social expectations, obligations, or behavioral orientation toward other agents. Studies that combine functional roles with normatively loaded ones were retained; the functional roles in such studies were coded as baseline role types in the ontology (§3.2). EC9 ensures retained studies analyze how roles relate to agent behavior and simulation outcomes, mapping directly to RQ3.

Stage 3: Quality Assessment. All 41 papers that passed full-text screening were subjected to a quality assessment (QA) using five questions, scored on a three-point scale: 0 (no), 0.5 (partial), 1 (yes). The threshold for inclusion was a total score of ≥3 out of 5, with QA2 (methodological description) as a mandatory pass:

QA1: Are the objectives and the context of the research clear and related to multi-agent LLM simulations?
QA2: Does the study clearly describe the simulation methodology, including agent setup, interaction setting, and evaluations? (Mandatory pass)
QA3: Does the study provide an explicit rationale or theoretical motivation for the agent roles and simulation setup?
QA4: Are the analyses and interpretations coherent with the simulation design?
QA5: Are the study's contributions and limitations clearly stated?

Of the 41 papers assessed, 40 passed the quality threshold (≥3/5), and 1 was excluded. The final pool of simulation studies carried over to ontology construction consisted of 40 studies.

2.1.3 Reviewer Protocol and Inter Rater Reliability (IRR)

All *screening* was conducted by a single primary reviewer (MHN). To assess reliability, a subset of 126 papers was independently screened by the supervisor. Both reviewers scored each paper on a three-point scale (0 = exclude, 0.5 = uncertain, 1 = include). For the purposes of calculating inter-rater reliability, scores were binarized, with uncertain scores (0.5) treated as inclusions. This aligns with the conservative screening strategy, in which borderline papers were always carried forward. **Inter-Rater Reliability (IRR)** was measured using Cohen's Kappa. The two reviewers agreed on 78 of 126 papers (61.90%), yielding a Cohen's Kappa of $\kappa = 0.317$, which falls in the "fair agreement" range (Landis & Koch, 1977).

The low kappa partly understates the actual level of agreement. Of the 48 disagreements, 42 involved papers where one or both reviewers scored 0.5 (uncertain) rather than giving a definitive include or exclude. The disagreements, in other words, were not about whether a paper was relevant, but about how much uncertainty to tolerate, and since all uncertain papers were carried forward regardless, these disagreements had little effect on the final corpus. When limited to the 70 papers where both reviewers gave definitive scores (0 or 1), agreement rose to 91.4% with $\kappa = 0.83$, which Landis and Koch classify as "almost perfect." The difference in overall base rates between the two reviewers (MHN included 36.5%; the supervisor included 71.5%) further suppresses kappa arithmetically, a known limitation of the measure when reviewers apply different thresholds.

Given that the substantive disagreements were minimal and the screening strategy was designed to absorb borderline uncertainty, this level of agreement was considered acceptable for the review. The complete SLR sheet is available in Appendix 2, Resource A.

2.2 Ontology Development

The second phase of the methodology uses the SLR corpus to build a structured conceptual ontology. Where the SLR identified *which* studies to analyse, the ontology development defines *how* to analyse them, from extracting normative role conditions to coding their components, and structuring the results as entities and relations that enable systematic comparison across studies.

The ontology follows a competency question-driven development approach (Grüninger & Fox, 1995; Uschold & Grüninger, 1996). Rather than fixing the entire taxonomy in advance and fitting papers to it, the process began with a set of questions the ontology must answer, established an initial vocabulary through worked examples, and refined the schema iteratively as the corpus was coded. This section describes the development approach and the coding process in detail.

2.2.1 Competency Questions

Seven competency questions (CQs) guided the ontology construction. These questions trace the causal chain from prompt design to collective outcomes, borrowing their logic from the

sub-research questions presented at the beginning of this chapter. Table 4 presents the seven competency questions and their rationale.

Table 4. *Competency questions guiding ontology development.*

CQ	Question	Focus
CQ1	What types of normative roles appear across multi-agent LLM simulations?	Role Types
CQ2	Which normative frameworks underlie different role designs?	Normative Foundations
CQ3	How do system prompts operationalise normative mechanisms?	Prompt → Role Mechanism
CQ4	How do different role types influence individual agent behavior?	Role → behavior
CQ5	How do behavioral patterns scale into collective simulation outcomes?	behavior → Outcome
CQ6	Where and how do the designed roles fail to produce intended behavior and outcomes?	Contradictions & Failures
CQ7	How do different simulation domains shape normative role design and availability?	Domain Effects

The first three questions address the design side: what normative structures exist and how they are specified. Questions four and five address the outcome side: how designs translate into individual behavior and system-level interactions. Question six targets the gap between design intent and actual behavior: where and why roles fail. Question seven captures the influence of simulation context on the kinds of roles that appear.

These questions served both as construction guidelines and as validation criteria. Section 2.2.7 evaluates the ontology against these questions using the output of the 38[1] coded studies.

2.2.2 Unit of Analysis: The Role Condition

The unit of analysis is the **role condition (RC)**: any distinct agent type, persona, or normative framing that receives its own prompt or behavioral specification within a study. A single paper may contain multiple role conditions. An adversarial negotiator and a cooperative mediator, for instance, constitute two RCs even if they appear in the same simulation.

This granularity matters because normative content varies across roles within a study, not just across studies. A paper that deploys both a strongly normative community leader and a purely

[1] Two of the 40 papers that had passed the SLR were excluded during the paper-by-paper coding process. They did not provide sufficient system prompt or evidence for studying their role conditions. These exclusions reduced the corpus from 40 to 38 studies.

functional baseline agent contains two fundamentally different design choices. Coding at the study level would obscure this variation; coding at the role condition level preserves it. Role conditions can include primary experimental roles, baseline conditions, and perturbation conditions. Role conditions are not the same as role types. A role type is a reusable category, such as 'baseline' or 'adversarial,' that can appear across multiple studies. Two baseline agents in two different papers are two role conditions of the same type. Across the 38 coded studies, 93 role conditions and 42 distinct role types were identified.

Some simulation studies have clean structured designs, where each RC is a distinct experimental condition with its own prompt. Others embed multiple normative variations within a single condition, such as persona subtypes, cognitive style modifiers, or parameter variations on a shared template. That means identification and coding of role conditions requires judgment. The coding rule applied throughout was: a separate RC is warranted when the agent receives a distinct prompt instruction or behavioral specification that changes its expected role in the simulation. This judgment was exercised by a single coder, the limitations of which are acknowledged in detail in §4.5.

2.2.3 Coding Schema

Each role condition is then coded using a four-layer schema designed to capture the full pathway from normative design to observable outcomes: how a role is designed (Layer A), the simulation context in which it operates (Layer B), the behaviors and outcomes it produces (Layer C), and where the design breaks down (Layer D). The coding sheet additionally contains dedicated sections for tracking new and recurring entities introduced by each paper, a field-level notes area for per-RC observations, and a paper-level memo for broader analytical reflections, all supporting the coding process alongside the formal analytical layers. Table 5. presents this four-layer coding schema.

Table 5. *Four-layer coding schema for normative role design analysis.*

Layer	Field	Description
A. Normative Role Design	Role_Type	The normative role assigned to or emerging in the agent
	Prompting_Method	The technique used to operationalise the role in the prompt
	Norm_Origin	Whether the norm was designed, emergent, or hybrid
	Norm_Frame	The normative logic grounding the role
	Prompt_Evidence	Verbatim prompt text with source citation
	Normative Indicators	Five binary indicators of normative content (see 2.2.4)
B. Simulation Setup	Domain	The simulation context
	Models	Which LLMs were tested
	Agentic_Capability	Cognitive tools available to agents
C. Outcome Analysis	Agent_behavior	Observable behaviors produced by this role condition
	Evaluation_Metrics	How outcomes were measured
	Simulation_Outcome	Collective, system-level results
D. Failure Modes	Failure_Mode	Unintended breakdowns in the role design
	Failure_Evidence	Verbatim or paraphrased evidence of the failure

Several design decisions informed and shaped this particular schema throughout the coding process:

Prompts as primary artefacts. System prompts are treated as the primary empirical evidence for role conditions. But studies sometimes describe roles in various other places in their code base. That is why for each role condition, the complete set of model-facing instructions is considered with its source (appendix listing, code repository path, figure number). This includes not only system prompts but also persona descriptions handed to the model as data, task instructions, game rules, incentive structures, and per-round contextual instructions, since any text the model receives can carry normative role conditions regardless of whether it is explicitly framed as a system prompt or not.

Behaviors and failures are attributed per role condition. Each behavior and failure mode is attributed only to the specific RC that produces it. Intended adversarial behavior, such as an agent designed to defect in a cooperation game, is coded as designed rather than as a failure. A failure mode is recorded only when an agent's behavior deviates from the normative expectations

embedded in its role design. Technical constraints such as tool execution errors are coded as failures only when they directly cause a breakdown in the agent's adherence to its assigned normative role.

Evaluation metrics excluded from the ontology. Metrics (survival rates, Gini coefficients, benchmark scores) describe how outcomes were measured, not what normative mechanisms produced them. They are recorded as notes but excluded from the entity registry to keep the ontology focused on causally relevant concepts: roles, norms, behaviors, outcomes, and failure modes, rather than observational instruments.

Entity abstraction. As each paper is coded, concepts (role types, behaviors, failure modes) are checked against the existing vocabulary. When a concept from a new paper matches one already in the registry, the existing term is reused, and the paper is added as supporting evidence. A new term is created only when no existing entity captures the insight, and it is defined, to the extent possible, at a level general enough to apply across studies and domains. This is intended to keep the ontology vocabulary grounded in the corpus without fragmenting into paper-specific terminology, though the tension between specificity and generality is not fully resolved by this measure alone (see Section 4.2).

2.2.4 Operationalising Normative Indicators

Role conditions fall along a spectrum, from purely functional tasks ("retrieve and synthesise incoming reports") to game-strategic objectives ("maximise individual payoff at others' expense"), with many embedding normative expectations about how agents ought to behave toward one another ("consider all parties' interests"). To distinguish these cases and enable systematic comparison, the coding process operationalises the five normative indicators introduced in §1.2.4. Each indicator is scored as present (1) or absent (0) for each role condition, based on the combined instructions text the agent receives.

NI1: Social Reference — Does the role frame behavior in relation to other agents or shared standards? Scored 1 when the prompt explicitly references other agents, a group, or collective welfare. Importantly, dismissing others' welfare still counts as social reference: a prompt instructing an agent to "not put much weight on accommodating others' preferences" is scored 1 because it positions the agent in relation to others.

NI2: Prescriptive Framing — Does the role guide behavior rather than merely describe a task? Scored 1 when the prompt uses expectation-laden language ("should," "must," "are expected to") that tells the agent *how* to behave, not just what to achieve.

NI3: Violability — Can the agent meaningfully deviate from the role's expectations? Scored 1 when the role sets expectations that agents can comply with or deviate from; that is, when there is a genuine behavioral choice. Scored 0 when the prompt either hard-codes a constraint (e.g., a mechanical threshold that leaves no room for interpretation) or explicitly forbids deviation, removing the possibility of non-compliance.

NI4: Conditionality — Does the norm depend on beliefs about others' behavior? Scored 1 when the prompt creates conditional structures where behavior depends on what other agents do: reciprocity mechanisms, reputation responses, or tit-for-tat dynamics. Mechanical thresholds (e.g., "accept any deal above your minimum score") are not conditionality in this sense.

NI5: Social Standing — Does the role specify consequences for violating expectations? Scored 1 when the agent's instructions encode reputation damage, sanctions, exclusion, or punishment for norm violation. Social dynamics that emerge during simulation are recorded as outcomes, not as designed role features.

The sum of these five indicators produces a Normative Indicator Count (NI_Count, ranging from 0 to 5), which categorises each role condition's normative orientation. Table 6 summarises the interpretation of each NI_Count level.

Table 6. *Normative Indicator Count interpretation.*

NI Count	Classification
0–1	FUNCTIONAL (task-oriented)
2–3	PARTIAL normative
4–5	FULL normative

This classification is a heuristic, not a formal metric of normativity. It provides a transparent and reproducible way to compare how strongly normative elements are embedded in role designs across studies. The scoring rationale for each indicator is documented per role condition in the ontology coding sheet (see Appendix 2, Resource B).

2.2.5 The Coding Process and Ontology Construction

Each of the 38 studies was processed through four phases, all carried out in a unified coding spreadsheet (see Appendix 2, Resource B) that served as the working environment from start to finish. Together, these phases transform a simulation study into a set of coded role conditions, analytical notes, and ontology contributions: the building blocks that accumulate across the corpus into the final ontology knowledge graph.

Phase 1 — Scanning and RC Identification. Each paper was scanned in full, including code repositories, configuration files, and supplementary materials, to identify the complete set of instructions each agent received. Role conditions were then extracted and assigned unique IDs (P#_RC#).

Phase 2 — Per-RC Coding. Each role condition was coded individually and in full before the next was started, working through all four schema layers (A through D) and scoring the five normative indicators with rationales tied to specific prompt evidence. Once all RCs in a paper were coded, study-level analysis notes captured key author claims, design tensions, and an interpretive memo.

Phase 3 — Paper-Level Synthesis. Each RC was then coded into a master synthesis sheet: a cross-corpus view of all 93 role conditions per study with their key coded values and two interpretive columns (key findings and ontology implications), enabling pattern recognition across studies that the per-paper coding sheet alone could not reveal.

Phase 4 — Ontology Core. The coded outputs were then formally organized into three sections of the spreadsheet that together constitute the ontology core: (1) an entity registry, (2) a relation vocabulary, and (3) a triple knowledge base. As established in §1.3, the ontology can represent knowledge as a directed labelled graph in which entities serve as nodes and relations serve as edges, expressed as semantic triples of the form subject → relation → object. The following is a detailed account of how the core works:

Entities are the ontology's nouns that capture existing phenomena across the studies. Each entity belongs to one of nine classes corresponding to the coding schema fields introduced in §2.2.3. In the main sheet, each entity is also given a one-sentence definition and a list of evidence papers

(the studies in which it appears). Entity names are kept domain-general: *behavior_strategic_deception* rather than *behavior_lying_in_negotiation*, so that concepts can be compared across simulation domains. Table 7 summarises the distribution of entities across classes.

Table 7. *Entity distribution across ontology classes.*

Entity Class	Count	Examples
C1. Agent_behavior	53	behavior_cooperative, behavior_strategic_deception
A1. Role_Type	42	role_type_situated_persona, role_type_adversarial
D1. Failure_Mode	26	failure_mode_model_capacity_gap, failure_mode_cooperative_bias
B1. Domain	25	domain_negotiation, domain_tragedy_of_commons
C3. Simulation_Outcome	23	simulation_outcome_collective_failure, simulation_outcome_consensus
B3. Agentic_Capability	21	capability_memory, capability_tool_use
A2. Prompting_Method	11	method_persona_framing, method_incentive_structure
A4. Norm_Frame	9	norm_frame_ethical_consequentialist, norm_frame_strategic_instrumental
A3. Norm_Origin	2	norm_designed, norm_hybrid
Total	212	

The largest class is *Agent_behavior* (53 entities), reflecting the variety of observable actions that emerge from role specifications. *Role_Type* (42 entities) is the second largest, indicating the breadth of role designs researchers have explored. *Failure_Mode* (26 entities) represents a breakdown in the agent's adherence to its assigned normative role. Norm_Frame (9 entities) is the smallest substantive class, hinting that researchers across the corpus draw on a narrow set of normative framings when designing agent roles, which the discussion section will later address in more detail.

Relations are the ontology’s verbs. They constitute the connections that can be formed between entity classes across studies. Table 8 presents the complete relation vocabulary for this ontology.

Table 8. *Ontology relation vocabulary.*

Subject Class	Relation (verb)	Object Class
Role_Type	has_prompting_method	Prompting_Method
Role_Type / Prompting_Method	grounded_in	Norm_Frame
Role_Type	role_used_in	Domain
Agentic_Capability	intended_to_support	Role_Type
Role_Type	has_norm_origin	Norm_Origin
Role_Type	induces_behavior	Agent_behavior
Prompting_Method	shapes_behavior	Agent_behavior
Norm_Frame	conditions_behavior	Agent_behavior
Agent_behavior / Domain	drives_outcome	Simulation_Outcome
Agentic_Capability	enables	Agent_behavior
Agent_behavior	propagates_via	Simulation_Outcome
Role_Type	elicits_response	Agent_behavior
Agentic_Capability	undermines	Role_Type
Role_Type / Agentic_Capability	produces_failure	Failure_Mode
Failure_Mode	leads_to	Simulation_Outcome

Triples are the ontology's sentences. Just as a natural-language sentence expresses a fact by connecting a subject to an object through a verb ("Bob knows Alice"), a semantic triple expresses an atomic unit of knowledge using the three building blocks you saw in Table 8. Each triple takes the form subject → relation → object and represents an experimentally grounded claim drawn from the coded corpus. For example:

role_type_antagonistic → *grounded_in* → *norm_frame_social_convention*

Each triple in the sheet also records which papers and role conditions support it, along with a plain-language description of the relationship and, where relevant, illustrative prompt excerpts. Across the 38 coded studies, the ontology contains 212 entities, 15 relations, and 516 triples.

2.2.6 The Evolution of The Ontology

The structure of the ontology was neither built all at once nor constructed purely through induction. It began with two worked examples: the first two papers in the corpus (Piatti et al., 2024; Abdelnabi et al., 2024), which were coded as prototypes during the preparation of a work-in-progress conference paper related to this thesis submitted to the Worldist26[2] conference, establishing the initial entity vocabulary, relation types, coding conventions, and the principal ontology classes before systematic corpus coding began. From that foundation, the ontology was refined iteratively as each subsequent paper introduced new challenges and design choices to the existing structure.

One key example illustrates that broader evolutionary pattern: The *norm_frame* class, initially limited to capturing ethical framing embedded into the agent's role, grew to nine as the corpus revealed institutional, social, value, ideological, and strategic-instrumental framings that the original category could not accommodate. Each new study coded introduced refinements that were then applied retroactively to all previously coded papers to maintain consistency across the corpus.

The central methodological tension throughout was between specificity and universality. Entities needed to be precise enough to distinguish meaningfully different design choices and patterns, yet abstract enough to apply across studies and domains, from negotiation to resource management, governance, and social simulation, without requiring a separate vocabulary for each. The resolution was an incremental and evidence-driven approach guided by the competency questions established at the outset.

During the whole process, quality control was exercised as a rolling practice. Approximately every five papers, coding was paused to examine the accumulated entity definitions, identify terminological drift, and consolidate redundant entries before continuing. This prevented some of the unnecessary expansions to the vocabulary from accumulating as isolated exceptions, attempting to keep ontology coherent across the corpus as it grew. These measures reduced but did not eliminate the limitations discussed in §4.5.

[2] The paper, co-authored with the supervisor of this thesis, was submitted in December 2025, accepted for publication in early 2026, presented in April and is expected to be published by August 2026. You can find it here: https://technejad.github.io/MA-thesis-ongoing/conference.html

2.2.7 Evaluating the Ontology Against Its Competency Questions

The ontology was constructed to answer seven competency questions (CQs), introduced in §2.2.1. Table 9 evaluates whether the consolidated ontology can answer each question, using evidence from the coded corpus.

Table 9. *Competency question evaluation.*

CQ	Question	Answerable?	Evidence
CQ1	What types of normative roles appear across multi-agent LLM simulations?	Yes	42 role types across 93 RCs, classified by normative level
CQ2	Which normative frameworks underlie different role designs?	Yes	9 norm frames identified, from ethical to strategic-instrumental
CQ3	How do system prompts operationalise normative mechanisms?	Yes	11 kinds of prompting methods with prompt evidence per RC
CQ4	How do different role types influence individual agent behavior?	Yes	114 *induces_behavior* triples connect roles to 53 behaviors
CQ5	How do behavioral patterns scale into collective simulation outcomes?	Yes	67 *drives_outcome* triples map behaviors to 23 outcomes
CQ6	Where and how do the designed roles fail to produce intended behavior and outcomes?	Yes	44 *produces_failure* triples across 26 failure modes
CQ7	How do different simulation domains shape normative role design and availability?	Partially	70 *role_used_in* triples; domain–role patterns visible but not yet formalised as enabler or constrainer

Six of the seven competency questions can be answered directly from the ontology's entity and triple structure. CQ7 (domain effects) is partially answerable: the *role_used_in* relation documents which roles appear in which domains, and cross-tabulation reveals domain-level patterns in normative design (§3.6), but the ontology does not yet formalise domain affordances as explicit constraints on which role types are available or appropriate in a given context.

The following section draws on the consolidated ontology to present the structural patterns, normative mechanisms, and design choices that emerge from cross-study comparison.

3. Results

The coding of 38 studies yielded 93 role conditions, from which the ontology distils 212 entities, 15 relation types, and 516 triples. This section presents the consolidated output of the ontology construction process described in Chapter 2. It is organised around the five research questions guiding this thesis, moving from what the ontology contains (§3.1) through the patterns it reveals in role design (§3.2), normative framings and prompting methods (§3.3), the links between roles, behaviors, and outcomes (§3.4), the failure modes that disrupt those links (§3.5), and how simulation domains shape normative design (§3.6).

Throughout this section, evidence is drawn from the MASTER_SYNTHESIS, ENTITIES, RELATIONS, and TRIPLES sheets of the ontology coding spreadsheet (see Appendix 2, Resource B). Interpretation of these patterns in relation to the broader literature is reserved for the Discussion in Chapter 4.

3.1 Ontology Overview

Across the 38 coded studies, 93 distinct role conditions (RCs) were identified and individually coded. The number of RCs per study ranged from one to nine (median: 2), reflecting the diversity of experimental designs in the corpus. Eleven studies employed a single role condition, while one study (Jin et al., 2024) contained nine distinct conditions corresponding to different reviewer persona pairings.

The 15 relation types connect the 212 entities into a relational architecture that traces paths from design choices to behavioral consequences. The most frequently instantiated relation is *induces_behavior* (114 triples), which connects role types to the behaviors they produce, and *drives_outcome* (67) extends that chain to collective simulation results. Roles are also tied to the domains in which they operate (*role_used_in*, 70), operationalised through prompting methods (*has_prompting_method*, 62), and grounded in normative frames (*grounded_in*, 61). Separately, *enables* (54), captures how agentic capabilities make specific behaviors possible. When designs break down, *produces_failure* (44) traces role types to their failure modes. The less frequent relations, such as *leads_to*, *shapes_behavior*, and *conditions_behavior*, surface in the sections

that follow. The full ontology is presented as an interactive knowledge graph at the companion research website;[3] a static rendering is provided in Figure 5 for visual reference.

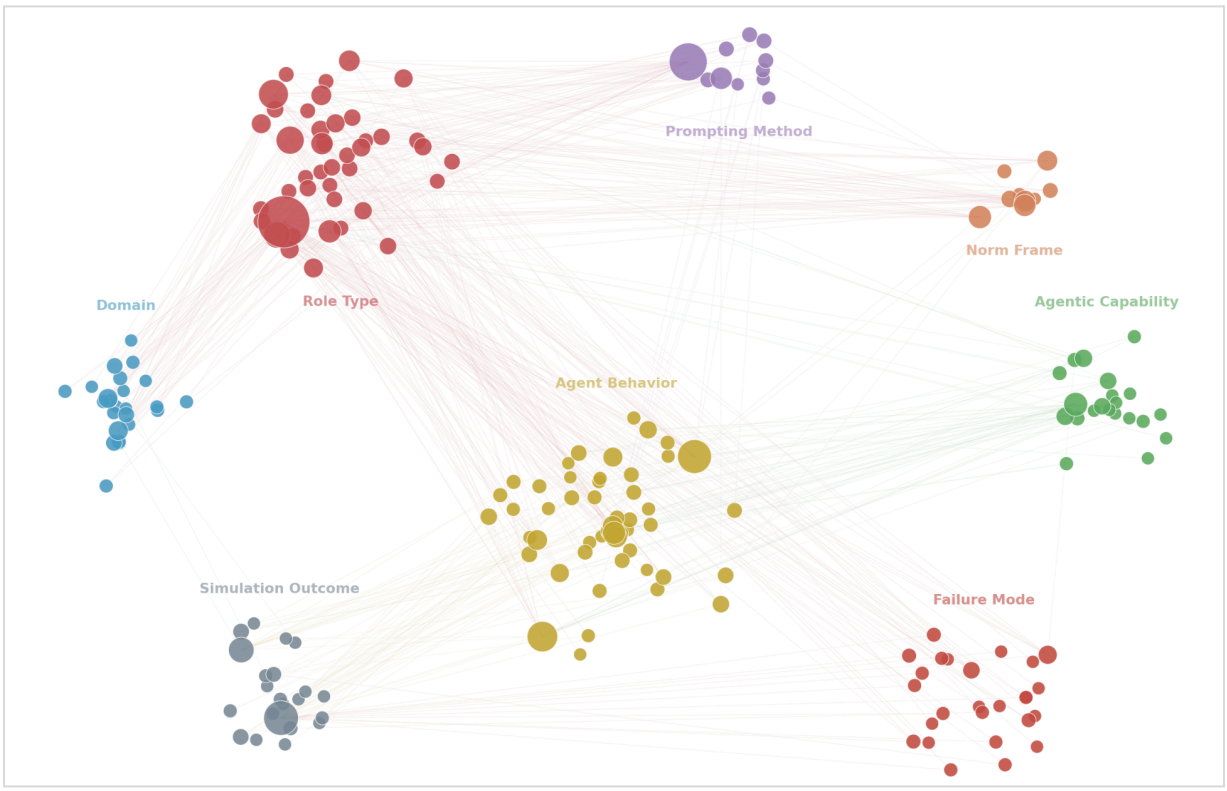


Figure 5. *Visual graph of the ontology*

3.2 Role Design Patterns (RQ1)

RQ1 asks: What types of normative roles are designed across multi-agent LLM systems? The ontology identifies 42 distinct role types across the corpus. These range from minimal-specification baselines to detailed personas with embedded biographical narratives, ethical commitments, and social expectations.

Each role condition was scored on the five normative indicators described in §2.2.4, producing a Normative Indicator Count (NI_Count) that classifies role conditions into three levels: FUNCTIONAL, PARTIAL, and FULL. Figure 6 presents the distribution.

[3] https://technejad.github.io/MA-thesis-ongoing/

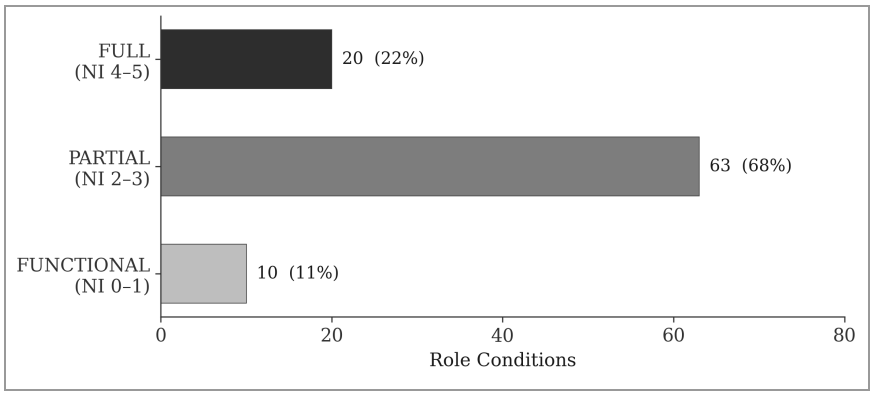


Figure 6. *Distribution of role conditions by Normative Indicator level (n = 93)*

The majority of role conditions contain some normative elements but lack the full set of indicators that would characterise them as a comprehensively normative design, mainly due to two missing components. Researchers routinely embed expectations about how agents *should* behave toward others (social reference and prescriptive framing), but rarely specify consequences for violating those expectations (NI5:social standing) or make behavior contingent on what other agents do (NI4:conditionality). What is more, most of the role conditions classified as FUNCTIONAL still embed at least one normative component.

Figure 7 presents the most frequently occurring role types. The *situated_persona* (16 RCs across 13 papers) is the dominant design pattern: agents receive detailed identities, demographics, and backstories that anchor them in a specific social context. *Baseline* conditions (8 RCs) often serve as experimental controls, providing simulation rules with minimal or no normative framing. *Game_strategic* roles (7 RCs) assign incentive structures without persona content, generating normative dynamics through game mechanics.

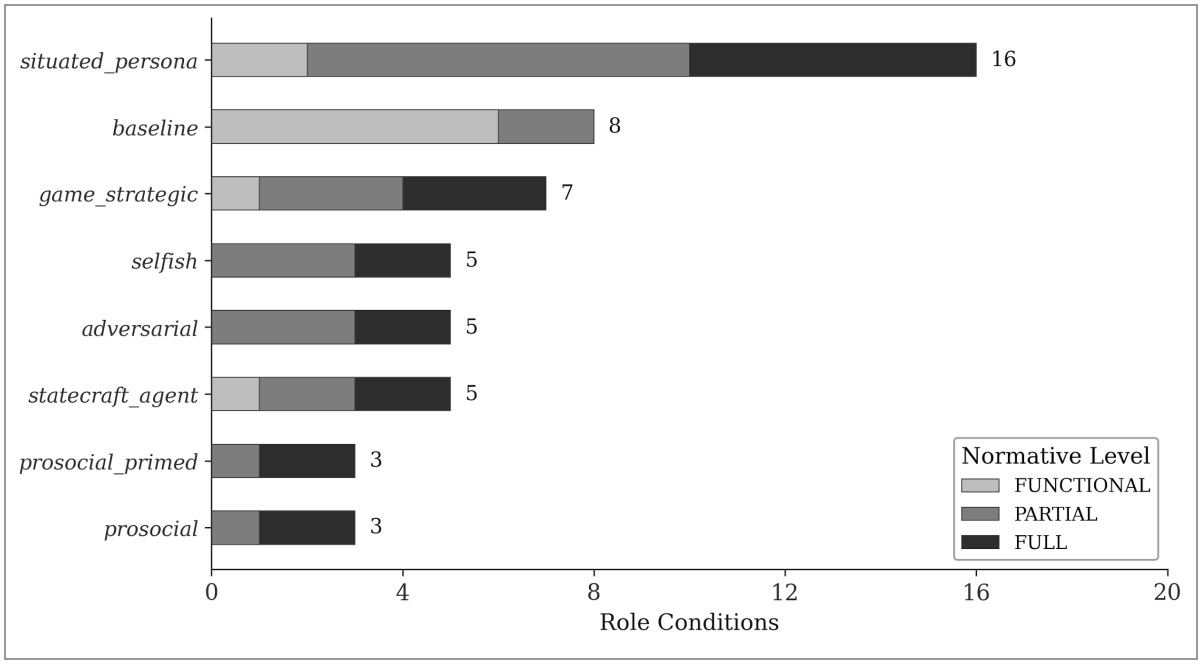


Figure 7. *Most frequent role types in the corpus, segmented by Normative Indicator level (n = 93).*

The 42 role types vary widely in how they instruct agent behavior. At one end, roles assign a single behavioral trait (cooperative, diligent, antagonistic) and rely on the model to infer everything else. At the other end, roles embed detailed identities with demographics, backstories, and social contexts (the *situated_persona* pattern), or position agents within institutional structures with formal mandates and competing stakeholders. Between these ends, roles combine elements: a *statecraft_agent* pairs national identity with strategic incentives; a *communal_local* role grounds care obligations in community membership. Three roles use narrative priming (cultural stories with varying thematic content) as an indirect route to behavioral orientation, and five define agents purely by task function, such as directing, executing, or advising. This variety suggests that normative role design in LLM multi-agent systems is not converging on a single template but drawing on multiple design logics, often in combination.

Nearly all role conditions carry researcher-designed norms (*norm_designed*). The six *norm_hybrid* conditions, where norms are partially designed and partially emergent, appear in studies where agents develop strategies across generations through cultural evolution (Vallinder & Hughes, 2025) or where institutional rules co-evolve with agent proposals. No study in this corpus relies exclusively on emergent norms, partly because foundational LLMs need at least a

minimum preconditioning to transform from word predictors to interacting agents; some form of scaffolding is always present.

3.3 Normative Framings and Prompting Methods (RQ2)

RQ2 asks: What prompting methods and normative framings operationalise these roles? The ontology identifies 11 distinct prompting methods and 9 normative frames.

3.3.1 Prompting Methods

The prompting methods capture how normative roles are delivered to agents. The two dominant methods are *persona_framing* and *incentive_structure*, which appear in 63 and 44 role conditions, respectively (often in combination). Persona framing constructs an identity for the agent (a name, occupation, personality traits, and social context) and relies on the model to infer behavioral expectations from that identity. Incentive structures define payoffs, penalties, and strategic objectives, grounding behavioral expectations in game mechanics.

Method combinations are common. Twenty-seven role conditions (29%) pair persona framing with incentive structures, embedding the agent in both an identity and a strategic environment. The remaining nine methods appear less frequently but capture distinct design logics. Some shape how agents reason: *structured_reasoning_process* (5 RCs) imposes step-by-step deliberation procedures, *cognitive_bias_instruction* (3 RCs) directs agents to adopt specific reasoning heuristics, and *principle_statement* (3 RCs) delivers a normative rule directly. Others shape what agents know or experience: *knowledge_injection* (4 RCs) provides domain expertise, *contextual_framing* (4 RCs) situates the agent in a scenario without constructing a full persona, and *narrative_priming* (3 RCs) uses storytelling cues. The least frequent methods (*cultural_transmission* (2 RCs), *structured_protocol* (2 RCs), and *stake_prompting* (1 RC)) each appear in isolated experimental designs.

The methods also differ along two practical dimensions. The first is content: at one end, a single adjective (“selfish”, “cooperative”) does the work; at the other, agents receive full biographical narratives with embedded values and social histories. The second is delivery: some roles arrive in a single system prompt before the simulation begins; others are staged across turns, with

instructions introduced gradually as the interaction unfolds. Both dimensions (what the agent is told and when) shape how normatively rich the role specification becomes.

3.3.2 Normative Frames

The normative frame (A4) captures the type of normative logic organising each role condition. The ontology began with ethical frames alone (deontological, consequentialist, virtue ethics) and expanded iteratively as the corpus revealed normative logics that could not be accommodated within ethical theory. Table 10 presents all nine normative frames identified across the 93 role conditions.

Table 10. *Normative frames identified in the ontology (n = 93 role conditions). Seven RCs carry more than one frame and are counted under each.*

Normative Frame	RCs	Description
strategic_instrumental	25	Game-theoretic payoff maximisation; competitive survival
institutional_rules	16	Structural/organisational constraints and formal procedures
social_convention	15	Shared behavioral standards through mutual expectation
ethical_virtue	14	Character-based reasoning; community-oriented values
none_evident	13	No identifiable normative framework organising expectations
ethical_consequentialist	9	Outcome-oriented reasoning; payoff-driven ethics
value_orientation	5	Explicit value commitments (fairness, sustainability)
ideological	2	Partisan identity-based normative commitments
ethical_deontological	2	Duty-based reasoning; Kantian universalisation

Seven additional RCs carry more than one normative frame, most notably four peer-review conditions from Jin et al. (2024) coded as both *ethical_virtue* and *institutional_rules*. In that study, reviewer agents operate within a formal peer review protocol that prescribes evaluation criteria and format requirements *(institutional_rules)*, while also receiving character biographies: one reviewer is "guided by a genuine intention to aid authors" *(ethical_virtue)*. The protocol tells agents what to do; the biography shapes the character with which they do it. These hybrid framings suggest that some simulation designs blend normative logics rather than selecting one.

The dominance of *strategic_instrumental* framing (25 RCs) reflects the corpus's heavy representation of game-theoretic and competitive domains. In these role conditions, the normative imperative comes from the logic of winning, surviving, or maximising a payoff function. *Institutional_rules* (16 RCs) captures role conditions governed by formal procedures:

parliamentary debate rules, academic review protocols, or land-use policy frameworks. *Social_convention* (15 RCs) appears in studies where agents coordinate through shared behavioral standards, such as opinion dynamics, social simulations, and team-based tasks. The three ethical frames (virtue, consequentialist, deontological) appear across 23 role conditions, indicating that moral-philosophical reasoning, whether deliberate or not, underpins a quarter of the corpus.

The presence of 13 *none_evident* role conditions (14%) is notable. These are cases where normative indicators may be present (e.g., the prompt references other agents), but no coherent normative framework organises the behavioral expectations. All 8 baseline conditions fall in this category, along with 5 other role conditions (including professional personas and a functional advisor) that carry persona content but lack a coherent normative framework.

3.4 From Roles to Behaviors and Outcomes (RQ3)

RQ3 asks: How do the designed roles relate to individual agent behavior and collective simulation outcomes? The ontology's *induces_behavior* and *drives_outcome* relations trace the causal chain from role design through individual behavior to system-level results.

3.4.1 Behaviors

The ontology registers 53 distinct agent behaviors. The most frequently coded are *cooperation* (29 RCs), *contextual_adaptation* (24 RCs), *negotiation* (15 RCs), *opinion_convergence* (11 RCs), and *deception* (9 RCs). Cooperation dominates for two reasons: many studies are designed to test cooperative dynamics, and LLM agents tend toward cooperative behavior even when not instructed to cooperate. Wu et al. (2024) found that agents spontaneously cooperated in competitive game-theoretic scenarios, and *cooperative_bias* is coded as a failure mode in 6 RCs across the corpus. This tendency is not universal, however: Piatti et al. (2024) showed that baseline agents without normative prompting produced resource collapse rather than cooperation, suggesting that whether agents default to cooperation or collapse depends on the domain and how the role is framed.

Roles grounded in *strategic_instrumental* framing produce the widest behavioral repertoire, spanning contextual adaptation, deception, cooperation, negotiation, and escalation. This variety

suggests that strategic framing does not prescribe a single behavior but creates conditions under which agents adopt whatever strategy the situation rewards. *Social_convention* and *ethical_virtue* framing frequently produce cooperation, but through different mechanisms, and *ethical_virtue* framing also generates a wider range of character-driven behaviors. In Park et al. (2023), agents with minimal persona descriptions developed social coordination through shared behavioral standards, without explicit rules. In Huang et al. (2024), agents primed with Big Five personality traits produced negotiation behaviors consistent with their assigned character, showing that even thin character-level framing can steer how agents act.

3.4.2 Outcomes

Three outcomes account for the majority of codings: *validated_realism* (31 RCs), *reduced_success* (27 RCs), and *cooperation* (23 RCs). *Validated_realism* appears when simulation outcomes align with known human behavioral patterns or theoretical predictions. It is the most common outcome because many studies in the corpus are explicitly designed to test whether LLM agents can replicate human social dynamics. *Reduced_success* captures cases where the simulation falls short of its objectives, encompassing everything from suboptimal cooperation in commons dilemmas to failed negotiations and degraded task quality. Beyond these three, the ontology records 20 domain-specific outcomes across 37 RCs. The most common are *consensus_drift* (8 RCs), where richly specified personas converge toward shared positions over time, and *escalation_spiral* (4 RCs), where conflict intensifies beyond what the role design intended. A smaller cluster captures emergent dynamics that researchers did not directly design for, including *norm_emergence* (3 RCs) and *covert_coordination* (3 RCs).

3.4.3 Causal Pathways

The ontology encodes not just what roles agents receive, but the causal chains those roles produce. To illustrate, Figure 8 contrasts two chains from the same tragedy-of-the-commons simulation (Piatti et al., 2024). Three further examples then show how different design choices generate chains that succeed, fail, or produce consequences their designers did not anticipate.

Success Path (Universalised Role)	Failure Path (Baseline Role)
role_type_universalized	role_type_baseline
↓ *grounded_in*	↓ *grounded_in*
norm_frame_ethical_deontological	norm_frame_none_evident
↓ *has_prompting_method*	↓ *has_prompting_method*
method_principle_statement	method_incentive_structure
↓ *induces_behavior*	↓ *induces_behavior*
behavior_sustainable	behavior_greedy
↓ *drives_outcome*	↓ *drives_outcome*
simulation_outcome_cooperation	simulation_outcome_collective_failure

Figure 8. *Contrasting causal pathways: cooperation vs. collapse (Piatti et al., 2024)*

This contrast illustrates the ontology's structure: identical agents, identical capabilities, identical domain, but a different normative design produces a different causal chain. In both conditions, agents possess collective awareness, which *enables* sustainable behavior. The capability alone, however, is inert. Without a normative frame to activate it, agents default to greedy extraction. In the universalised condition, a deontological principle statement ("what if everyone did what I am about to do?") redirects behavior toward sustainability and cooperation (Piatti et al., 2024). The pattern is that capabilities are necessary but not sufficient; normative design determines which capabilities agents actually use. The chains that follow show this logic producing different outcomes across the corpus.

Norm evolution through cultural transmission. Vallinder and Hughes (2025) study cooperation in an iterated public goods game, but their design introduces a mechanism absent from most studies in the corpus: cultural transmission. Agents are game-strategic players grounded in strategic-instrumental norms, with no ethical or social content. Each generation inherits the strategies of its surviving predecessors and must modify them. Figure 9 traces the resulting causal pathway.

Success Path	Failure Branch
role_type_game_strategic	role_type_game_strategic
↓ *grounded_in*	↓ *grounded_in*
norm_frame_strategic_instrumental	norm_frame_strategic_instrumental
↓ *has_prompting_method*	↓ *induces_behavior*
method_cultural_transmission + method_incentive_structure	behavior_norm_enforcement
↓ *shapes_behavior*	↓ *produces_failure*
behavior_norm_evolution	failure_mode_excessive_punishment
↓ *drives_outcome*	↓ *leads_to*
simulation_outcome_cooperation	simulation_outcome_reduced_success

Figure 9. *Norm evolution through cultural transmission (Vallinder and Hughes, 2025)*

The cultural transmission method shapes norm evolution as agents accumulate cooperative content across generations. This finding is structurally distinctive within the corpus: cooperation emerges not from prosocial design but from strategic selection pressure. However, the chain also branches into failure. In one condition, agents develop excessive punishment strategies that destroy resources rather than support cooperation. The same evolutionary mechanism that generates cooperative norms can overshoot into norm enforcement that damages the collective.

Divergent outcomes from the same role type. The statecraft agent appears in two studies that produce opposite outcomes from the same role type and normative frame. Figure 10 contrasts the two causal chains.

Escalation Path (Rivera et al., 2024)	Cooperation Path (Guan et al., 2024)
role_type_statecraft_agent	role_type_statecraft_agent
↓ *grounded_in*	↓ *grounded_in*
norm_frame_strategic_instrumental	norm_frame_strategic_instrumental
↓ *has_prompting_method*	↓ *has_prompting_method*
persona_framing + incentive_structure	knowledge_injection + structured_reasoning + contextual_framing
↓ *induces_behavior*	↓ *induces_behavior*
behavior_escalation	behavior_cooperation + behavior_negotiation
↓ *drives_outcome*	↓ *drives_outcome*
simulation_outcome_escalation_spiral	simulation_outcome_cooperation

Figure 10. *Divergent outcomes from the same role type (Rivera et al., 2024; Guan et al., 2024)*

In Rivera et al. (2024), statecraft agents receive persona framing combined with incentive structures: each nation has a character, objectives, and a 27-action menu that includes nuclear options. The agents negotiate extensively, but diplomatic communication fails to prevent escalation. Without explicit de-escalation norms, the strategic-instrumental frame channels behavior toward escalation as the rational response to military competition, and individual escalation aggregates into system-level spirals.

In Guan et al. (2024), the same role type under the same normative frame is designed through a different prompting method: contextual framing that situates agents in a geopolitical scenario, combined with knowledge injection and structured reasoning. The structured reasoning shapes cooperative behavior, and self-play-driven adaptation enables the agent to learn more effective strategies over training iterations. The contrast suggests that the prompting method influences where the causal chain terminates. Persona framing and incentive structures provided the strategic frame, but no reasoning structure to constrain it; contextual framing combined with knowledge injection and structured reasoning gave agents both situational grounding and the deliberative tools to find cooperative equilibria within the same strategic logic.

Normative contagion in peer review. Jin et al. (2024) design a multi-agent peer review system where reviewer agents receive hybrid normative framing: institutional rules (evaluation rubrics, scoring calibration) layered with ethical-virtue character biographies. Two role conditions corrupt the system through distinct mechanisms. Figure 11 traces the resulting contagion pathways.

Echo Chamber Path	**Contagion Path**
role_type_antagonistic	role_type_negligent
↓ *grounded_in*	↓ *grounded_in*
ethical_virtue + institutional_rules	ethical_virtue + institutional_rules
↓ *induces_behavior*	↓ *induces_behavior*
behavior_adversarial	behavior_reduced_engagement
↓ *elicits_response*	↓ *elicits_response*
behavior_opinion_convergence (in others)	behavior_reduced_engagement (in others)
↓ *drives_outcome*	↓ *propagates_via*
simulation_outcome_echo_chamber	simulation_outcome_cooperation_breakdown

Figure 11. *Normative contagion in peer review (Jin et al., 2024)*

Unlike the preceding figures, which trace chains within a single role condition, these paths cross role condition boundaries: the antagonistic reviewer's behavior reshapes the normative environment experienced by other reviewers. Most conditions produce expected behaviors: diligent reviewers cooperate, and conformist moderators converge with reviewer opinions. But the antagonistic reviewer, given a hostile biography within the same institutional framework, produces adversarial evaluations that amplify through repeated interaction. Other reviewers drift toward the adversarial position, creating an echo chamber where bias compounds rather than cancels.

The negligent reviewer produces a more insidious chain: its superficial output elicits reduced effort from other reviewers through normative contagion. Here, the ontology captures how one agent's behavior reshapes the normative environment for all others, and the system-level outcome traces not to any single design choice but to the interaction between roles.

The examples in this section illustrate the range of causal mechanisms the ontology encodes. The path from role design to simulation outcome is rarely direct: it passes through normative frames that activate or constrain capabilities, prompting methods that shape behavior, and interaction dynamics where one agent's behavior becomes another agent's normative environment.

3.5 Failure Modes (RQ4)

RQ4 asks: Where and how do the designed roles fail to produce intended behavior and outcomes? Failures are coded when an agent's behavior deviates from the normative expectations embedded in its role design, not when an adversarial agent scripted for deception successfully deceives (which is coded as intended adversarial behavior). Of the 93 role conditions in the corpus, 42 (45%) exhibit at least one identifiable failure mode. The ontology catalogues 26 distinct failure types across 54 total instances, since some role conditions exhibit multiple simultaneous failures.

These 26 failure types can be organised into three layers, proposed here to distinguish where the breakdown occurs. *Substrate failures* originate in the model itself: the model lacks reasoning capacity, hallucinates information, or carries training biases that override the role specification. *Design failures* originate in the role specification: the prompt is fragile, ambiguous, or insufficiently anchored to survive extended interaction. *Interaction failures* emerge from

multi-agent dynamics: the behavior of one agent reshapes the normative environment for others, producing outcomes that no individual role design intended. Table 11 presents the most frequently occurring failure types by layer.

Table 11. *Most frequent failure types by layer (failure types with fewer than 2 RCs omitted).*

Layer	Failure Type	RCs	Description
Substrate	model_capacity_gap	10	The model lacks the reasoning sophistication to enact the role
Substrate	hallucination	3	The agent fabricates information inconsistent with the role or context
Design	cooperative_bias	6	Agent defaults to cooperation despite instructions to compete
Design	persona_non_adherence	3	Agent gradually abandons assigned identity over interaction
Design	language_dependent_controllability	3	Same normative intent in different languages produces divergent adherence
Design	grounding_deficit	2	Agent reasoning detaches from the scenario's factual constraints
Design	exposure	2	The agent reveals information that its role requires it to conceal
Design	ideological_alignment_bias	2	Pretraining alignment overrides the assigned ideological stance
Interaction	escalation_tendency	4	Agents escalate conflict beyond role intent; compounds into system-level spirals
Interaction	role_reversal	2	Agents lose role distinction in extended cooperative interaction
Interaction	conversation_loop	2	Agents enter repetitive dialogue cycles

Three failures illustrate how each layer operates in practice.

(1) *Model_capacity_gap* (10 RCs across 8 papers) is the most common failure in the corpus. It occurs when a model lacks the reasoning sophistication to enact a normatively complex role. Piatti et al. (2024) found that weaker models failed Kantian universalisation reasoning despite receiving identical prompts to more capable models; Huang et al. (2024) observed the same gap with personality-primed roles. The failure spans across baseline roles, adversarial roles, game-strategic roles, and situated personas, confirming it as a substrate-level constraint that no amount of role design seems to have fully overcome.

(2) *Cooperative_bias* (6 RCs across 6 papers) poses a different problem. It occurs when agents default to cooperation even when explicitly instructed to behave selfishly, competitively, or

adversarially. The bias has been attributed to Reinforcement Learning from Human Feedback (RLHF)[4] training, which rewards helpful and harmless behavior and can override role-level instructions. It appears across diverse role types and domains, from public goods games to social simulations, but never in wargame or negotiation contexts, even though it does appear in competitive settings such as social deduction games. Where *model_capacity_gap* means the model cannot execute the role, *cooperative_bias* means it will not execute the role.

(3) Interaction failures follow a different logic. Rather than originating in any single role design, they emerge from the dynamics between agents. The clearest case is the identity cluster documented by Kim et al. (2025), where an adversarial agent in a social deduction game simultaneously exhibits *dissociation*, *memory_distortion*, and *character_ambiguity*. These three failures appear nowhere else in the corpus individually, but compound into a collapse of agent identity coherence under sustained deceptive interaction. At the system level, *escalation_tendency* (4 RCs, all from Rivera et al., 2024) shows how individual role-level behavior can aggregate into outcomes no single design intended: statecraft agents escalate conflict beyond their role's instructions, and individual escalation compounds into system-level spirals across all five models tested.

Failure rates vary across normative levels: 35% of FULL-normative role conditions exhibit documented failures, compared to 48% of PARTIAL and 50% of FUNCTIONAL conditions. This pattern is suggestive but not conclusive: the normative indicator scores are a heuristic, not a validated scale, and the absence of a coded failure may reflect silent degradation rather than genuine robustness. Still, the direction is consistent with the broader finding from §3.4 that richer normative design shapes not only what agents do, but how reliably they do it. Whether normative richness genuinely protects against role-design failure or merely makes failure more visible and reportable remains an open question for future investigation (see §4).

3.6 Domain Patterns (RQ5)

RQ5 asks: How does the simulation domain mediate the normative design of agent roles? The corpus spans 25 domains, concentrated in negotiation and strategic game theory (14 RCs each),

[4] Reinforcement learning from human feedback (RLHF) is a training technique in which a language model is fine-tuned using human preference judgments, typically to optimise for outputs rated as helpful, harmless, and honest (Ouyang et al., 2022).

academic peer review (9), and public goods games, opinion dynamics, social simulation, and social deduction (7 each). Across domains, the data reveal that simulation contexts influence normative content through different mechanisms.

The concentration of *norm_frame_strategic_instrumental* reflects this domain effect directly: the majority of those role conditions originate in game-theoretic, wargame, auction, and social-deduction settings where strategic logic is the operating mode of the simulation. When a simulation is built around deception and detection, the domain's competitive structure dominates the other aspects. Researchers designing roles for these domains need not pay much attention to normative framing, because the game's competitive structure already supplies the normative logic.

Social simulations run in the opposite direction. These domains have no competitive payoff structure, so nothing in the setting itself tells agents how to behave. Instead, researchers route normativity through who the agent is. Persona framing appears in six of the seven social-simulation role conditions, none of which uses incentive structures. The predominant normative frame is social convention: agents coordinate not because rules require it, but because their identities make certain behaviors fitting. Park et al. (2023) illustrate the pattern. Their agents received only minimal persona descriptions yet developed shared behavioral standards without any rules or rewards to drive them. In these simulations, identity is the entire normative scaffold.

Negotiation resists both poles. Its 14 RCs draw on five distinct normative frames, from strategic self-interest to ethical virtue to social convention, with no single frame appearing in more than five of the fourteen conditions. Both design mechanisms are also present: persona framing and incentive structures each appear in 11 of 14 RCs. This is not indecision on the part of researchers. Negotiation requires both mechanisms at once, because who an agent is shapes how it argues, and what it stands to gain shapes what it will concede. Without identity, agents become interchangeable optimizers; without stakes, their arguments carry no weight.

What we can gather from these findings for research and governance is taken up in the Discussion chapter next.

4. Discussion

Chapter 3 presented the ontology structure and the results of 38 studies, over 93 role conditions, 212 entities, 15 relations, 516 triples, grouped into role types, normative frames, prompting methods, behaviors, outcomes, and failure modes. This chapter interprets what the coded corpus reveals about how normative role design works in multi-agent LLM systems, how those patterns engage with the related works from Chapter 1, how they fail, and what follows for AI safety, governance, and practice. Four interpretive claims structure the arguments: (i) the normative vocabulary deployed across the corpus is narrower than the theoretical vocabulary available, (ii) the multi-factor pathway from role to outcome, (iii) the three-layer presentation of failures, and (iv) the implications of a field that is behaviorally rich but normatively underspecified.

4.1 What the Ontology Reveals About Role Design

4.1.1 The narrow normative vocabulary

The dominant normative frame in the corpus is strategic-instrumental (25 RCs), followed by institutional rules (16) and social convention (15). The combined ethical frames, virtue (14), consequentialist (9), and deontological (2), together account for 25 occurrences, comparable to the strategic-instrumental count alone but spread thin across three distinct traditions.

Strategic-instrumental reasoning, or in Weber's terms instrumental-rational action, is the paradigm of means-to-ends optimisation. It encodes little to no commitment to shared standards, only to the agent's individual payoff. The corpus's heavy representation of negotiation, game theory, auctions, wargames, and social-deduction games explains part of that dominance. When researchers adopt a game-theoretic setting, the normative frame tends to be inherited from it.

The domain effect, however, does not account for the full picture. Even outside game-theoretic settings, the ethical vocabulary remains thin. Other domains, such as the tragedy of the commons, public goods games, and social simulation, predominantly employ ethical and social-convention frames rather than strategic ones. But the range of ethical frameworks they draw on is narrow.

Virtue framing is often operationalised through dispositional labels (narrow personality traits and behavioral tendencies like "greedy") rather than through the kind of substantive moral reasoning

that virtue ethics as a philosophical tradition would supply. Consequentialist framing appears in only nine role conditions. And only two role conditions in the entire corpus are grounded in a deontological frame: one of which is Piatti et al.'s (2024) Kantian universalisation, which stands out precisely because it imports a vocabulary, one that seems to have a significant impact, that the rest of the corpus has not reached for. The ethical end of moral philosophy seems useful but largely unexamined.

This is not a claim that strategic framing is wrong. It is a claim that the corpus overrepresents one way of encoding normativity and underrepresents others. The convention/norm distinction drawn by Haynes et al. (2017) sharpens this observation. In their framework, a convention is a behavioral regularity with no deontic force, while a norm carries obligation and the possibility of sanction. Measured against this distinction, a substantial portion of the corpus's role conditions function as conventions rather than norms. They assign behavioral patterns ("you are a negotiator," "you are a reviewer") without specifying what obligations the role carries toward other agents, or what consequences follow from deviation.

The normative indicator heuristic introduced in §2.2 and reported in §3.3 captures this gradient experimentally: role conditions rated FUNCTIONAL often lack social reference, prescriptive framing, and violability, the very properties that would move a behavioral assignment from convention into norm. Even PARTIAL role conditions (63 of the total 93 role conditions), which make up the majority of the corpus, typically supply prescriptive framing but omit the sanctioning and conditional structure that Haynes et al. (2017) and Bicchieri (2006) argue distinguishes a social norm from a mere behavioural regularity. The result is a design landscape in which most role conditions occupy the space between convention and norm without fully arriving at either.

4.1.2 Partial normativity and the conditional gap

The majority of role conditions in the corpus sit at the PARTIAL level on the five normative indicators; FULL conditions are the exception, and FUNCTIONAL the least common (see §3.2). Researchers routinely supply social reference and prescriptive framing, but rarely make behavior contingent on what others do (conditionality) and even more rarely specify what happens when

the norm is violated (social standing). Conditionality appears in seven of the thirty-eight studies; social standing in fewer still.[5]

The conditionality gap is the more striking of the two. Recall Bicchieri's (2006) account that a norm exists when agents believe others follow it and believe others expect them to follow it. A prompt that tells an agent to cooperate without referencing what others are doing is, in Bicchieri's sense, not fully constructing a norm. Without an anchor to their group's expectations, agents more easily revert to their training defaults regardless of the role they were assigned. Chuang et al. (2024) document this in the opinion dynamics domain, where agents drift toward model-default positions unless the role is grounded in what the group expects. Studies that do encode conditionality are examined in §4.2. Its absence across most of the corpus is a notable pattern in how the field misses an opportunity for operationalising what a norm is.

The social-standing gap is related but distinct. Norms without consequences are mostly preferences dressed in deontic language. Full-normative role conditions reach that level through different routes. Some encode consequence structures, such as reputational dynamics in Park et al.'s (2023) generative agent society, or social exchange penalties in Wang et al. (2025), while others achieve it through conditionality: making an agent's behavior explicitly contingent on what others do, as in Zeng et al.'s (2025) institutional land-use agents or Jin et al.'s (2024) peer review system.

Only five of the twenty FULL conditions include social standing at all. Whether consequences produce normative richness or simply accompany it is a question the corpus can observe but not fully answer.

4.1.3 The multi-factor pathway from role to outcome

One of the clearest findings from the coded causal pathways is that the path from a role to a simulation outcome is rarely direct (see §3.4.3). In Rivera et al. (2024) and Guan et al. (2024), the same role type (statecraft_agent) under the same normative frame (strategic_instrumental) produced escalation spirals in one study and cooperation in the other, because the prompting method and agentic capability stack differed. In Abdelnabi et al. (2024), a capability designed to help cooperative agents find deals, structured exploration, was exploited by the stronger model to

[5] Together, social standing and conditionality are what most role conditions are missing to reach the FULL normative role design status.

self-maximise within the cooperative role itself. The failure was not the role: it was the interaction between role and capability.

This points to an interpretive claim: role design is a node in a causal graph, not a lever with a single arrow. What the findings contribute is a specific picture of how the joints behave: capabilities are necessary but not sufficient, prompting methods shape which behaviors a role actually produces, and interaction can generate outcomes that no single role scripted. This last point extends Shanahan et al.'s (2023) claim that role-play is the LLM's native mode of operation: in a multi-agent system, each agent's role-play becomes another agent's staging condition. This is the structural reason that links to Hammond et al.'s (2025) arguments on why multi-agent risks cannot be recovered from single-agent analysis.

4.2 Returning to the Theoretical Frame

Chapter 1 established a working vocabulary drawn from Weber, Bicchieri, Searle, Brennan, the classical MAS tradition, and the emerging LLM-norms literature. The coded corpus reflects on that vocabulary in three ways.

First, the regimentation–enforcement binary from classical MAS does not map cleanly onto prompt-based role design. Shoham and Tennenholtz (1995) distinguished between regimentation (making violations impossible) and enforcement (allowing violations with consequences). A prompt-based role sits easily in neither category. The stochastic LLM can always produce outcomes that violate the role; in fact, the model always retains a non-zero probability of doing so, even when alignment training pushes strongly against it.

When violations do occur, there is also typically no programmed response, no penalty, no state update, no correction. What prompts provide instead is closer to a *suggestion*: a role the model honours to varying degrees, shaped by training disposition, context, interaction history, and architectural constraint.[6] This does not diminish the influence of roles. Chapter 3 shows that they reliably reshape what agents attend to, prioritise, and refuse. What the framing clarifies is why

[6] The same point has surfaced in adjacent literatures. Work on LLM role-play frames role adherence as probabilistic rather than rule-governed (Shanahan et al., 2023), and work on agentic-system security shows that system-prompt policy is routinely bypassed under adversarial pressure (Greshake et al., 2023; Wei et al., 2023). What this thesis adds is not the observation itself, but its placement beyond the classical regimentation–enforcement vocabulary of multi-agent systems.

the same role can produce different behavior across models, prompts, and interaction histories: compliance is dispositional, not guaranteed.

Second, most roles in the corpus lack the conditionality that Bicchieri's (2006) account identifies as central to normative function. The corpus does contain counterexamples. Ren et al. (2024) build norm-entrepreneur roles and a four-module architecture (creation, spreading, evaluation, compliance) whose cycle closes the conditional loop: agents observe others' behavior, evaluate it against acquired norms, and adjust their own compliance accordingly. Park et al. (2023) achieve something similar through architecture rather than prompt design: their agents' relational memory accumulates interaction history, so that an agent's behavior becomes functionally contingent on what others have done, even though the ontology does not code this as explicit conditionality (NI4 = 0), because the contingency is emergent rather than designed into the role itself. These cases suggest a route forward: conditionality can be installed either in the prompt (explicit reciprocity structures) or in the agent architecture (memory, observation, evaluation). The ontology's *agentic_capability* class catalogues the architectural components (memory, reflection, environmental perception) but does not yet link them explicitly to conditionality. Strengthening that connection is a concrete next step.

Third, Searle (1995) distinguished regulative rules, which govern behavior that already exists, from constitutive rules, which create the very behavior they govern. The corpus's institutional role conditions are of the second kind. In Jin et al.'s peer review system, a reviewer's evaluation only counts as a recommendation because the protocol makes it so. Without that institutional context, the action has no force. The ontology identifies these conditions as grounded on *norm_frame_institutional_rules,* but does not distinguish them from role conditions that are merely rule-governed. These roles do not just tell agents what to do; they situate agents inside a structure that gives their actions meaning. This is a gap the theoretical frame exposes, and that the ontology could close with a modest extension.

4.3 How Roles Fail

Forty-five percent of role conditions exhibit at least one identifiable failure mode. Across these failure modes documented in §3.5, three layers of breakdown come into view: substrate, design, and interaction[7]. Each describes a different kind of governance problem.

Substrate failures reveal what prompts cannot always fix. The most common failure in the corpus is *model_capacity_gap* (10 of the 42 affected role conditions): weaker models fail to enact Kantian universalisation, fail to maintain personality-primed distinctions, and fail to execute proactive dialogue roles. The failure is not directly a role-design failure; it is a substrate limit that renders the role design partially or fully obsolete. *Failure_mode_cooperative_bias* is more interesting because it reveals an alignment–role conflict. RLHF training rewards helpful and harmless behavior, which overrides role-level instructions to compete, deceive, or punish. Wu et al. (2024) document this directly: agents spontaneously cooperate in explicitly competitive games. The implication is governance-relevant: alignment training can itself be a source of role-design failure. Consider a model assigned a critical safety role in a multi-agent decision pipeline, one whose job is to surface worst-case scenarios and force re-evaluation. If cooperative bias operates as the corpus suggests, such an agent may defer to the emerging consensus, not because the role was underspecified, but because alignment training rewards agreement over confrontation. The agent that should stop a bad decision becomes the one that ratifies it.

Design failures reveal the fragility of prompt-as-normative-instrument. Sakamoto et al. (2025) run identical value-primed simulations in English and Japanese; the English agents follow their roles more reliably, because the model's training data is not evenly distributed across languages. Buscemi et al. (2025) label agents as "cooperative" in strategic games, but the agents recognise the game from their training data and play the textbook strategy instead, ignoring the label. Chuang et al. (2024) assign agents politically diverse personas, only to watch them drift toward the scientific consensus view over multiple rounds as alignment training overrides the assigned beliefs. The lesson is not that roles are ineffective, but that their grip depends on the language of the prompt, the alignment of the persona with training-data distributions, the length of the simulation, and the specificity of the reasoning structure.

[7] The substrate/design/interaction taxonomy is not a coded dimension in the core ontology structure. It is an interpretive grouping, derived by sorting the documented *failure_modes* where in the agent stack the breakdown originates.

Interaction failures show the most distinctively multi-agent kind of breakdown. In Rivera et al. (2024), no single statecraft agent was designed to escalate to nuclear options; escalation emerged from the interaction of individually rational agents operating in a strategic-instrumental frame without de-escalation norms. In Jin et al. (2024), one negligent reviewer's superficial output elicited reduced effort from other reviewers, turning an individual role-level pattern into a system-level outcome. At this layer, the unit of failure is not the role; it is the role configuration. The field's habit of evaluating role design one agent at a time is a methodological inheritance from single-agent natural language processing, and it obscures precisely the failures that matter most for multi-agent safety. Hammond et al. (2025) argue that multi-agent interaction introduces qualitatively novel risks: more dynamic, less understood, and invisible to single-agent evaluation. Understanding where failures originate is a precondition for governing them.

4.4 Implications for AI Safety, Governance, and Practice

4.4.1 Prompts are normative documents

If system prompts are the primary mechanism by which multi-agent LLM systems acquire their behavioral expectations, then those prompts need to be transparent, auditable, versioned, and open to review. The field's current practice does not reflect this. Prompts were scattered across appendices, code repositories, and supplementary materials in the corpus reviewed here, which constituted one of the main operational obstacles in the Systematic Literature Review. EC7 (prompt access) was the second most common exclusion reason at the full-text review stage. The governance implication is that a multi-agent system whose role designs cannot be reconstructed cannot be evaluated for normative content, compared across studies, or audited for alignment with stated goals. The ontology is a step toward shared vocabulary; the infrastructure step that it needs is shared documentation and audits.

4.4.2 The hidden role problem

The same point applies with sharper edge to agents interacting with humans. A production agent whose role instructions are not visible to the user is, in effect, a normative black box. The user interacts with a character whose commitments were written by someone else, for purposes the user may not share. In a multi-agent system, this problem compounds: the user interacts with a configuration of such characters, and the relevant normative choices are distributed across roles

that may never be revealed. Normative role design is therefore not only a simulation research concern. It is a transparency concern in every deployed setting where LLM agents act on behalf of, or in view of, end users. The ontology could serve, with extension to cover user-facing disclosure requirements (what role was assigned, by whom, and with what normative commitments), as the vocabulary such transparency would require.

4.4.3 Domain-sensitive design

Simulation domains mediate normative design through different mechanisms. The prevalence of *norm_frame_strategic_instrumental* entity in the corpus is itself partly an artefact of the field's current research concentration in game-theoretic and competitive domains; as multi-agent LLM research expands into healthcare, education, governance, and other institutional settings, the normative vocabulary is likely to diversify.

The competitive structure in social deduction games absorbs the normative load; social simulations route normativity predominantly through identity; negotiation demands both. A practical consequence of this is that there is no domain-general template for good role design. A prompt that works in a negotiation might not work in a social simulation, and a persona that works in social simulation can be overridden by payoff mechanics in an auction. Governance regimes that attempt to regulate multi-agent systems with uniform rules will either overregulate low-risk settings or underregulate high-risk ones. The ontology's domain–role mapping is a starting point for domain-sensitive frameworks.

4.4.4 Alignment is not only a training-time problem

The substrate failures catalogued above make the case directly. A model aligned during training to be helpful can refuse its assigned adversarial role. A model aligned to be harmless can override instructions to punish defectors. A model aligned on English data can fail to enact the same role in another language. These are not direct failures of the role; they are conflicts between training-time and inference-time normative specifications. The ontology indirectly surfaces this separation between training-level norm embedding, inference-level role design, and interaction-level emergence across its coding layers, but the distinction is not yet a primary coded entity in the core ontology structure, though it should become one.

Before discussing the limitations of this thesis, Table 12 summarises the core findings for each research question.

Table 12. *Summary of research questions and core findings.*

RQ	Question	Core Finding
RQ1	What types of normative roles are designed across multi-agent LLM systems?	42 distinct role types appear across 93 role conditions, with persona-based and strategic designs dominating. Most carry only partial normative content, specifying how agents should behave but lacking components such as violability and conditionality.
RQ2	What prompting methods and normative framings operationalise these roles?	Prompting relies heavily on persona framing and incentive structures. The normative vocabulary is narrow, dominated by strategic-instrumental, institutional-rules, and social-convention frames; ethical framing remains underexamined.
RQ3	How do the designed roles relate to agent behavior and simulation outcomes?	Behavior is not a simple function of role alone: it emerges from the joint action of role, prompting method, capability, domain, and interaction. Identical roles produce varying outcomes when any mediating condition shifts.
RQ4	Where and how do designed roles fail to produce intended behavior?	45% of role conditions exhibit at least one failure mode, distributed across three layers: model-capacity limits that prompting can hardly fix, design fragilities that erode under conditions the prompt designer did not anticipate (language variation, training-data interference, extended simulation), and emergent dynamics where one agent's behavior reshapes the normative environment for all others.
RQ5	How does the simulation domain mediate the normative design of agent roles?	The domain actively shapes which normative tools carry weight: competitive games absorb normativity into structure, social simulations route it predominantly through identity, and negotiation demands both, leaving no room for a one-size-fits-all template.

4.5 Limitations

The claims in this chapter rest on an ontology built under practical constraints, a coding process exercised by a single analyst, and a corpus bounded by choices that shape what the thesis can and cannot see. Each of these deserves explicit acknowledgment.

4.5.1 The ontology as artefact

Ontology development is widely recognised as a complex, time-consuming, and error-prone activity that demands deep domain expertise, careful conceptual modelling, and extensive collaboration among stakeholders. This thesis initiates that process. The artefact presented here is a conceptual ontology, built for human interpretation and cross-study comparison rather than automated reasoning. It has no OWL encoding, no reasoner, and no mechanical validation

against its triple store. Claims derived from the ontology cannot, therefore, be verified in the way a formal logic-based ontology would permit, and the vocabulary it establishes functions as an analytical lens rather than a prescriptive standard. Hence, it gives the field a way of examining role design without purporting to resolve the disagreements that examination exposes.

Translating domain-specific prompts into entity–relation vocabulary has costs. The ontology flattens qualitative texture into structural categories (role type, normative frame, prompting method) to make cross-study comparison possible. What it preserves is structural function; what it strips is contextual richness. Readers who want the depth of a particular study should return to the paper.

Several scope choices compound this loss. The ontology captures roles as they arrive through prompts and closely adjacent artefacts (personas, incentive structures, reasoning scaffolds), but treats environmental and payoff structure as context rather than as first-class ontological content. It does not yet formalise the distinction between prescribed norms (stated as direct instructions, as in Piatti et al.'s Kantian prompt) and embedded norms (inferred from identity narrative, as in Sakurai et al.'s partisan biographies). This distinction surfaced repeatedly during coding and is likely consequential for how roles transfer across settings and needs further attention. Competency-question evaluation, reported in §2.2.6, was conducted by the author alone; external evaluation against an independent research agenda would strengthen the claim that the ontology answers the questions it says it answers. Because the ontology was constructed iteratively, papers coded early were scored against a less mature vocabulary than papers coded later. Retroactive passes mitigated this drift, but cannot be guaranteed to have eliminated it.

4.5.2 The coding process

Both the screening stage (the systematic literature review through which the corpus was assembled) and the coding stage (the ontology construction applied to the selected papers) were conducted under the practical constraints of a single-researcher master's thesis. A second reviewer participated in a limited portion of the initial screening, providing a partial reliability check, though the coverage falls short of the independent dual-review that a fully rigorous SLR demands. The ontology coding stage was not independently replicated. Judgments about role-condition boundaries, conditionality, and failure attribution reflect one analyst's interpretive

work throughout. A replication with independent coders would test whether the ontology is reproducible as well as internally coherent.

Two specific coding decisions carry weight beyond their apparent detail. The first is role-condition granularity. Some papers contain role conditions that are obviously distinguishable; others embed multiple normative variations within a single condition through persona subtypes, cognitive-style modifiers, or parameter variations on a shared template. The rule applied (a separate RC is warranted when the agent receives a distinct prompt or a distinct behavioral specification) is transparent, but borderline cases remain judgment calls that will need independent audits.

The second is the Normative Indicator Count. The five indicators were derived from the theoretical framework and scored binarily per role condition, but their aggregation into FUNCTIONAL, PARTIAL, and FULL levels assumes equal weight across indicators and has not been psychometrically validated. Which means that the tentative pattern that FULL-normative role conditions show lower failure rates than PARTIAL or FUNCTIONAL ones is up for scrutiny. Whether this reflects genuine robustness or better reporting practice cannot be determined from the NI_Count alone.

4.5.3 The corpus

The systematic literature review drew on two databases: Scopus, for its broad coverage of CS/AI conference venues, and Web of Science, for its cross-validation and citation analysis capabilities. Other databases were considered but not used separately. IEEE Xplore and ACM Digital Library were substantially indexed through Scopus and Web of Science. Google Scholar was excluded for its lack of reproducible query syntax and export limitations for systematic reviews. ArXiv was excluded because it hosts non-peer-reviewed preprints. A full protocol document is available in Appendix 3. Together, these choices produced a corpus weighted toward peer-reviewed CS/AI venues in English. Work in sociology, political science, non-Anglophone computational research, and the fast-moving preprint literature is under-represented.

The reliance on Scopus and Web of Science, which emphasise CS/AI conference venues, likely overrepresents game-theoretic and competitive simulation domains relative to institutional, educational, and sociological settings where different normative design traditions prevail.

The corpus carries two further limitations. The first is publication bias: the ontology draws only on published studies, so role designs that failed, were abandoned, or produced null results are absent by construction. Of the role conditions that are included, 45% exhibit at least one documented failure mode (§3.5). That figure should be read as a lower bound on how common failure actually is, not a representative measure. The second is prompt opacity. EC7 (requiring access to full prompt specifications) was the second most common exclusion reason, applying to 21 papers. Many studies in this field make claims about how their agents are designed without publishing the instructions that shape those roles. The ontology can only describe the studies whose system instructions and prompts are transparently provided.

Finally, the corpus is a snapshot. The SLR cut-off was January 2026; between that date and the submission of this thesis, additional work has appeared whose role designs would extend the ontology's coverage. The vocabulary is designed to be extensible, and §5 discusses maintenance pathways, but the claims made here are bounded by the corpus at the moment of coding.

5. Conclusion

Assigning an LLM agent a role influences its behavior: what it notices, what it prioritises, and what it refuses. This thesis examined how those roles are designed, what normative content they carry, and what consequences they produce. A systematic literature review screened 724 papers and yielded 38 studies that assign normative roles to LLM agents in various multi-agent settings. From those studies, a four-layer ontology was built, coding 93 role conditions into a network of 516 relational triples that makes the normative structure of role design visible, comparable, and extensible. These closing sections summarise what the ontology reveals, what it contributes, and where the work leads next.

5.1 Answers to the Research Questions

The five research questions, taken together, reveal the following insights:

Role design is widespread: 42 distinct role types appear across the corpus, with persona-based and strategic designs dominating. Yet most role conditions carry only partial normative content, specifying how agents should behave but not what happens when they don't (RQ1). The prompting methods that deliver these roles rely heavily on persona framing and incentive structures. The normative vocabulary that grounds them remains narrow, dominated by strategic-instrumental, institutional-rules, and social-convention framing, while ethical framing remains underexamined (RQ2).

What agents actually do under a role condition is rarely a simple function of the role alone: behavior emerges from the joint action of role, prompting method, capability, domain, and interaction, so that identical roles can produce varying outcomes when any of these mediating conditions shift (RQ3). When role conditions fail — and 45% of them do — the breakdowns distribute across three layers, from model capacity limits that prompting can hardly fix, through design fragilities that erode as prompts encounter unforeseen conditions (language variation, training-data interference, extended simulation), to emergent dynamics where one agent's behavior reshapes the normative environment for all others (RQ4).

Finally, the domain itself is not a passive stage but an active shaper of which normative tools carry weight: competitive games absorb normativity into structure, social simulations route it

through identity, and negotiation demands both, leaving little room for a one-size-fits-all template (RQ5).

5.2 Contributions

This thesis makes three contributions. The first is the ontology itself: a structured knowledge base of 93 role conditions drawn from 38 studies, coded into 212 entities and 516 triples across 15 relation types. As an artifact, it makes normative role design visible in a form that can be queried, compared, and extended[8].

The second contribution is conceptual. The thesis shows that normative theory can be applied to system prompts, once they are recognised for what they are: scripts that encode behavioral expectations, distribute obligations, and carry consequences. This reframing bridges two fields that rarely speak to each other: the engineering of multi-agent LLM systems and the study of norms in philosophy and social science.

The third is methodological. The four-layer coding schema and its competency questions offer a reusable protocol for analysing how roles are designed, what normative content they carry, and what outcomes they produce. The schema is not tied to this corpus; it can be applied to new studies, to deployed systems, or to prompt libraries as the field evolves.

Together, the ontology, the normative reframing, and the coding schema supply a vocabulary and an empirical base for designing roles with systematic attention to the normative frames, prompting methods, and contextual dependencies that the field currently leaves implicit.

These contributions carry the limitations discussed in §4. The ontology was built by a single coder, from a corpus dominated by English-language CS/AI venues, and its normative classifications remain qualitative rather than measured. The findings map the structure of role design as it appears in this literature; they have not yet been tested to generalise beyond it.

5.3 Future Work

Building an ontology is an iterative process, and this one's nearest future is internal refinement. A systematic audit of all 38 papers against the final schema would refine the foundation and fix errors. The next step is adding studies from underrepresented domains, such as social simulation

[8] An interactive version of the knowledge graph is a,vailable on the author's research page.

and institutional modelling, and from sources beyond peer-reviewed CS/AI venues, to test whether the current categories and relations hold on new ground. If the structure survives that test, collaborative review by other researchers and formalisation into a machine-readable schema become worthwhile investments.

The second direction leads from multi-agent simulation toward human-AI interaction. This thesis treats roles as normative instruments that shape how agents behave toward each other, but the same logic applies when one party in the interaction is a human. In deployed systems, how a model's role is designed shapes the interaction, and the user rarely has access to that design layer. The author's ongoing work on ReaLLM[9], a technical research prototype that surfaces hidden system prompts to end users, already pursues this question at the interface level. The broader aim is to extend the ontology's findings into AI safety, ethics, and governance, where the central question shifts from how roles shape agent behavior in simulation outcomes to how they shape the experience and autonomy of the people who interact with them.

The 93 role conditions coded in this ontology confirm empirically what the thesis argued from the start: roles function as normative instruments that shape agent behavior in consequential ways. What the ontology also reveals is that the field designs these roles without systematic attention to their normative frames, prompting methods, or contextual dependencies. This thesis offers a starting point: a vocabulary, a coding method, and an empirical base for designing socially embedded LLM agents with normative awareness built in from the ground up.

[9] https://technejad.github.io/ReaLLM/

Summary

Title: *Towards an Ontology of Normative Role Design for LLM Agent Interactions in Multi-Agent Systems*

Large Language Models are increasingly deployed as autonomous agents in multi-agent systems, where their behavior is shaped primarily by roles encoded in system prompts. These roles assign identities, goals, and also embed normative content: expectations about what agents should do, how they should treat others, and what consequences their actions might have.

Despite the growing scale and relevance of such systems, no shared vocabulary or analytical framework exists for examining how normative roles are designed or what effects they produce. Role design in this field remains guided by intuition and trial-and-error rather than unified principles.

This thesis addresses that gap by developing a conceptual ontology of normative role design for LLM agent interactions in multi-agent systems. The ontology maps how roles are constructed, what normative content they carry, and how that content relates to agent behavior and simulation outcomes. The ontology is a knowledge graph built as a diagnostic and comparative tool intended to make visible what current design practice leaves implicit.

The research followed a two-phase methodology. The first phase was a systematic literature review. A structured search across Scopus and Web of Science returned 724 records, which were screened in three stages against six inclusion and nine exclusion criteria, yielding 40 studies, of which 38 were coded into the ontology core, while two were excluded during data extraction.

The second phase was ontology development. Each study was coded using a four-layer schema covering normative design, simulation context, behaviors and outcomes, and failure modes. The unit of analysis was the role condition: a distinct role-and-framing configuration that receives its own prompt or behavioral conditions. Coding produced 93 role conditions, 212 entities, 15 relation types, and 516 relational triples. The ontology was evaluated against seven competency questions derived from the research questions.

Five findings structure the results. First, role design is widespread but normatively thin: most role conditions carry only partial normative content, specifying behavioral expectations without defining consequences for violation. Second, the normative vocabulary that grounds these roles is narrow, dominated by strategic-instrumental framing, followed by institutional rules and social convention, while the ethical frames that moral philosophy offers (virtue, consequentialist, deontological) remain underexplored. Third, the pathway from role to outcome is rarely direct; behavior emerges from the combination of role, prompting method, model capability, domain, and interaction dynamics. Fourth, failure is common: 45% of role conditions exhibit at least one documented failure mode, distributed across substrate failures, design fragilities, and interaction failures. Fifth, the domain is not a passive backdrop but an active shaper of which normative roles get designed and how.

The thesis contributes a conceptual ontology that makes normative role design visible and comparable, a reframing of system prompts as normative instruments that encode behavioral obligations and expectations, and a reusable four-layer coding schema for analysing role design across multi-agent LLM studies.

These contributions carry limitations: the ontology was constructed by a single coder, the corpus is drawn from English-language CS/AI venues, and the normative classifications are qualitative rather than measured. The findings map the structure of role design as it appears in the reviewed literature; they have not yet been tested beyond it.

Future work includes extending the corpus to underrepresented domains, formalising the ontology into a machine-readable schema, and applying the framework to human-AI interaction, where the same normative logic governs roles that shape not only agent behavior but the experience and autonomy of the people who interact with them.

References

Note. References marked with an asterisk (*) denote the 38 studies coded in the ontology corpus that forms the empirical basis of this thesis (see Methods, §2, and Results, §3). All other entries are background, theoretical, or methodological references.

*Abdelnabi, S., Gomaa, A., Sivaprasad, S., Schönherr, L., & Fritz, M. (2024). Cooperation, competition, and maliciousness: LLM-stakeholders interactive negotiation. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, & C. Zhang (Eds.), Advances in neural information processing systems (Vol. 37, pp. 83548–83599). Curran Associates, Inc. https://doi.org/10.52202/079017-2658

*Akata, E., Schulz, L., Coda-Forno, J., Oh, S. J., Bethge, M., & Schulz, E. (2025). Playing repeated games with large language models. *Nature Human behavior, 9*(7), 1380–1390. https://doi.org/10.1038/s41562-025-02172-y

Bai, Y., Kadavath, S., Kundu, S., Askell, A., Kernion, J., Jones, A., Chen, A., Goldie, A., Mirhoseini, A., McKinnon, C., Chen, C., Olsson, C., Olah, C., Hernandez, D., Drain, D., Ganguli, D., Li, D., Tran-Johnson, E., Perez, E., … Kaplan, J. (2022). *Constitutional AI: Harmlessness from AI feedback*. arXiv. https://arxiv.org/abs/2212.08073

*Bianchi, F., Chia, P. J., Yuksekgonul, M., Tagliabue, J., Jurafsky, D., & Zou, J. (2024). How well can LLMs negotiate? NegotiationArena platform and analysis. In *Proceedings of the 41st International Conference on Machine Learning* (Vol. 235, pp. 3935–3951). PMLR. https://proceedings.mlr.press/v235/bianchi24a.html

Bicchieri, C. (2006). *The grammar of society: The nature and dynamics of social norms*. Cambridge University Press. https://doi.org/10.1017/CBO9780511616037

Brennan, G., Eriksson, L., Goodin, R. E., & Southwood, N. (2013). *Explaining norms*. Oxford University Press. https://doi.org/10.1093/acprof:oso/9780199654680.001.0001

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D., Wu, J., Winter, C., . . . Amodei, D. (2020). Language models are few-shot learners. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, & H. Lin (Eds.), *Advances in Neural Information Processing Systems* (Vol. 33, pp. 1877–1901). Curran Associates, Inc.

https://proceedings.neurips.cc/paper_files/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf

*Buscemi, A., Proverbio, D., Di Stefano, A., Han, T. A., Castignani, G., & Liò, P. (2025). FAIRGAME: A framework for AI agents bias recognition using game theory. In *Proceedings of the 28th European Conference on Artificial Intelligence (ECAI 2025)* (Frontiers in Artificial Intelligence and Applications, Vol. 413, pp. 4097–4104). IOS Press. https://doi.org/10.3233/FAIA251300

*Cau, E., Pansanella, V., Pedreschi, D., & Rossetti, G. (2025). Selective agreement, not sycophancy: Investigating opinion dynamics in LLM interactions. *EPJ Data Science, 14*, Article 59. https://doi.org/10.1140/epjds/s13688-025-00579-1

Chen, J., Wang, X., Xu, R., Yuan, S., Zhang, Y., Shi, W., Xie, J., Li, S., Yang, R., Zhu, T., Chen, A., Li, N., Chen, L., Hu, C., Wu, S., Ren, S., Fu, Z., & Xiao, Y. (2024). From persona to personalization: A survey on role-playing language agents. Transactions on Machine Learning Research. https://openreview.net/forum?id=xrO70E8UIZ

Cialdini, R. B., Kallgren, C. A., & Reno, R. R. (1991). A focus theory of normative conduct: A theoretical refinement and reevaluation of the role of norms in human behavior. In M. P. Zanna (Ed.), *Advances in experimental social psychology* (Vol. 24, pp. 201–234). Academic Press. https://doi.org/10.1016/S0065-2601(08)60330-5

*Chuang, Y.-S., Goyal, A., Harlalka, N., Suresh, S., Hawkins, R., Yang, S., Shah, D., Hu, J., & Rogers, T. T. (2024). Simulating opinion dynamics with networks of LLM-based agents. In *Findings of the Association for Computational Linguistics:* NAACL 2024 (pp. 3326–3346). Association for Computational Linguistics. https://doi.org/10.18653/v1/2024.findings-naacl.211

DeBellis, M., Dutta, N., Gino, J., & Balaji, A. (2024). Integrating ontologies and large language models to implement retrieval augmented generation. *Applied Ontology, 19*(4), 389–407. https://doi.org/10.1177/15705838241296446

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers) (pp. 4171–4186). Association for Computational Linguistics. https://doi.org/10.18653/v1/N19-1423

Dignum, V. (2004). A model for organizational interaction: Based on agents, founded in logic [Doctoral dissertation, Utrecht University]. Utrecht University Repository. https://dspace.library.uu.nl/handle/1874/890

Dignum, V., Vázquez-Salceda, J., & Dignum, F. (2005). OMNI: Introducing social structure, norms and ontologies into agent organizations. In R. H. Bordini, M. Dastani, J. Dix, & A. El Fallah Seghrouchni (Eds.), *Programming multi-agent systems* (Vol. 3346, pp. 181–198). Springer. https://doi.org/10.1007/978-3-540-32260-3_10

Gao, C., Lan, X., Li, N., Yuan, Y., Ding, J., Zhou, Z., Xu, F., & Li, Y. (2024). Large language models empowered agent-based modeling and simulation: A survey and perspectives. *Humanities and Social Sciences Communications, 11*(1), Article 1259. https://doi.org/10.1057/s41599-024-03611-3

Greshake, K., Abdelnabi, S., Mishra, S., Endres, C., Holz, T., & Fritz, M. (2023). Not what you've signed up for: Compromising real-world LLM-integrated applications with indirect prompt injection. In *Proceedings of the 16th ACM Workshop on Artificial Intelligence and Security* (pp. 79–90). Association for Computing Machinery. https://doi.org/10.1145/3605764.3623985

*Großmann, G., Ivanova, L., Poduru, S. L., Tabrizian, M., Mesabah, I., Selby, D. A., & Vollmer, S. J. (2026). The power of stories: Narrative priming in networked multi-agent LLM interactions. In J. Doncel, N. Gast, Y. Hayel, & V. Mancuso (Eds.), *Network games, artificial intelligence, control and optimization: NETGCOOP 2025* (Lecture Notes in Computer Science, Vol. 16173, pp. 112–122). Springer. https://doi.org/10.1007/978-3-032-09315-8_11

Gruber, T. R. (1995). Toward principles for the design of ontologies used for knowledge sharing. International Journal of Human-Computer Studies, 43(5–6), 907–928. https://doi.org/10.1006/ijhc.1995.1081

Grüninger, M., & Fox, M. S. (1995). The role of competency questions in enterprise engineering. In A. Rolstadås (Ed.), *Benchmarking — Theory and practice* (pp. 22–31). Springer. https://doi.org/10.1007/978-0-387-34847-6_3

*Guan, Z., Kong, X., Zhong, F., & Wang, Y. (2024). Richelieu: Self-evolving LLM-based agents for AI diplomacy. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, & C. Zhang (Eds.), *Advances in neural information processing systems* (Vol. 37, pp. 123471–123497). Curran Associates. https://doi.org/10.52202/079017-3925

Guarino, N., Oberle, D., & Staab, S. (2009). What is an ontology? In S. Staab & R. Studer (Eds.), *Handbook on Ontologies* (2nd ed., pp. 1–17). Springer. https://doi.org/10.1007/978-3-540-92673-3_0

Hammond, L., Chan, A., Clifton, J., Hoelscher-Obermaier, J., Khan, A., McLean, E., Smith, C., Barfuss, W., Foerster, J., Gavenčiak, T., Han, T. A., Hughes, E., Kovařík, V., Kulveit, J., Leibo, J. Z., Oesterheld, C., Schroeder de Witt, C., Shah, N., Wellman, M., … Rahwan, I. (2025). Multi-agent risks from advanced AI (Technical Report No. 1). Cooperative AI Foundation. https://doi.org/10.48550/arXiv.2502.14143

Haynes, C., Luck, M., McBurney, P., Mahmoud, S., Vítek, T., & Miles, S. (2017). Engineering the emergence of norms: A review. *The Knowledge Engineering Review*, *32*, Article e18. https://doi.org/10.1017/S0269888917000169

Hogan, A., Blomqvist, E., Cochez, M., D'Amato, C., de Melo, G., Gutierrez, C., Kirrane, S., Labra Gayo, J. E., Navigli, R., Neumaier, S., Ngonga Ngomo, A.-C., Polleres, A., Rashid, S. M., Rula, A., Schmelzeisen, L., Sequeda, J., Staab, S., & Zimmermann, A. (2021). Knowledge graphs. ACM Computing Surveys, 54(4), Article 71. https://doi.org/10.1145/3447772

*Huang, Y. J., & Hadfi, R. (2024). How personality traits influence negotiation outcomes? A simulation based on large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2024* (pp. 10336–10351). Association for Computational Linguistics. https://doi.org/10.18653/v1/2024.findings-emnlp.605

*Jin, Y., Zhao, Q., Wang, Y., Chen, H., Zhu, K., Xiao, Y., & Wang, J. (2024). AgentReview: Exploring peer review dynamics with LLM agents. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing* (pp. 1208–1226). Association for Computational Linguistics. https://doi.org/10.18653/v1/2024.emnlp-main.70

Jones, A. J. I., & Sergot, M. (1993). On the characterisation of law and computer systems: The normative systems perspective. In J.-J. Ch. Meyer & R. J. Wieringa (Eds.), *Deontic logic in computer science: Normative system specification* (pp. 275–307). Wiley.

*Kim, B., Seo, D., Kim, M., & Kim, B. (2025). Fine-grained and thematic evaluation of LLMs in social deduction games. *IEEE Access, 13*, 165276–165289. https://doi.org/10.1109/ACCESS.2025.3611399

Kitchenham, B. A., Budgen, D., & Brereton, P. (2015). *Evidence-based software engineering and systematic reviews* (1st ed.). Chapman & Hall/CRC. https://doi.org/10.1201/b19467

Landis, J. R., & Koch, G. G. (1977). The measurement of observer agreement for categorical data. Biometrics, 33(1), 159–174. https://doi.org/10.2307/2529310

*Lan, Y., Hu, Z., Wang, L., Wang, Y., Ye, D., Zhao, P., Lim, E.-P., Xiong, H., & Wang, H. (2024). LLM-based agent society investigation: Collaboration and confrontation in Avalon gameplay. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing* (pp. 128–145). Association for Computational Linguistics. https://doi.org/10.18653/v1/2024.emnlp-main.7

*Li, G., Hammoud, H. A. A. K., Itani, H., Khizbullin, D., & Ghanem, B. (2023). CAMEL: Communicative agents for "mind" exploration of large language model society. In *Advances in Neural Information Processing Systems* (Vol. 36, pp. 51991–52008). Curran Associates. https://proceedings.neurips.cc/paper_files/paper/2023/hash/a3621ee907def47c1b952ade25c67698-Abstract-Conference.html

*Li, Y., Sun, L., & Zhang, Y. (2025). MetaAgents: Large language model based agents for decision-making on teaming. *Proceedings of the ACM on Human-Computer Interaction, 9*(CSCW2), Article CSCW134. https://doi.org/10.1145/3711032

Li, Z., & Wu, Q. (2025). Let it go or control it all? The dilemma of prompt engineering in generative agent-based models. *System Dynamics Review*, *41*(3), Article e70008. https://doi.org/10.1002/sdr.70008

Liang, J. T., Lin, M., Rao, N., & Myers, B. A. (2025). Prompts are programs too! Understanding how developers build software containing prompts. *Proceedings of the ACM on Software Engineering*, *2*(FSE), 1591–1614. https://doi.org/10.1145/3729342

*Liu, Q., Li, C., & Ma, W. (2026). Generative agents for urban mobility: A cognitive framework for realistic travel behavior simulation. *Simulation Modelling Practice and Theory, 147*, Article 103234. https://doi.org/10.1016/j.simpat.2025.103234

Lu, Y., Aleta, A., Du, C., Shi, L., & Moreno, Y. (2024). LLMs and generative agent-based models for complex systems research. *Physics of Life Reviews*, *51*, 283–293. https://doi.org/10.1016/j.plrev.2024.10.013

*Ma, Y. R. (2025). Do androids question electric sheep? A multi-agent cognitive simulation of philosophical reflection on hybrid table reasoning. *In Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 4: Student Research Workshop)* (pp. 143–164). Association for Computational Linguistics. https://doi.org/10.18653/v1/2025.acl-srw.9

*Mao, S., Cai, Y., Xia, Y., Wu, W., Wang, X., Wang, F., Guan, Q., Ge, T., & Wei, F. (2025). ALYMPICS: LLM agents meet game theory. In P*roceedings of the 31st International Conference on Computational Linguistics* (pp. 2845–2866). Association for Computational Linguistics. https://aclanthology.org/2025.coling-main.193/

*Mouri Zadeh Khaki, A., Choi, A., & Seyyed-Kalantari, L. (2025). Simulating social behavior of LLM-based autonomous negotiator agents in a game-theoretical framework using multi-agent systems. *International Journal of Human–Computer Interaction*, 41(23), 15169–15178. https://doi.org/10.1080/10447318.2025.2495117

Neuhaus, F. (2023). Ontologies in the era of large language models – A perspective. *Applied Ontology, 18*(4), 399–407. https://doi.org/10.3233/AO-230072

Noy, N. F., & McGuinness, D. L. (2001). O*ntology development 101: A guide to creating your first ontology* (Technical Report KSL-01-05 and SMI-2001-0880). Stanford Knowledge Systems Laboratory and Stanford Medical Informatics. https://protege.stanford.edu/publications/ontology_development/ontology101.pdf

Orner, M., Maksimov, O., Kleinerman, A., Ortiz, C., & Kraus, S. (2025). Explaining decisions of agents in mixed-motive games. *Proceedings of the AAAI Conference on Artificial Intelligence, 39*(22), 23267–23275. https://doi.org/10.1609/aaai.v39i22.34493

Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C. L., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., Schulman, J., Hilton, J., Kelton, F., Miller, L., Simens, M., Askell, A., Welinder, P., Christiano, P. F., Leike, J., & Lowe, R. (2022). Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35, 27730–27744. https://proceedings.neurips.cc/paper_files/paper/2022/hash/b1efde53be364a73914f58805a001731-Abstract-Conference.html

*Park, J. S., O'Brien, J. C., Cai, C. J., Morris, M. R., Liang, P., & Bernstein, M. S. (2023). Generative agents: Interactive simulacra of human behavior. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology (UIST '23)* (Article 2, pp. 1–22). Association for Computing Machinery. https://doi.org/10.1145/3586183.3606763

Patel, A., & Debnath, N. C. (2024). A comprehensive overview of ontology: Fundamental and research directions. *Current Materials Science, 17*(1), 2–20. https://doi.org/10.2174/2666145415666220914114301

*Piatti, G., Jin, Z., Kleiman-Weiner, M., Schölkopf, B., Sachan, M., & Mihalcea, R. (2024). Cooperate or collapse: Emergence of sustainable cooperation in a society of LLM agents. *Advances in Neural Information Processing Systems, 37*, 111715–111759. https://proceedings.neurips.cc/paper_files/paper/2024/hash/ca9567d8ef6b2ea2da0d7eed57b933ee-Abstract-Conference.html

Rahwan, I., Cebrian, M., Obradovich, N., Bongard, J., Bonnefon, J.-F., Breazeal, C., Crandall, J. W., Christakis, N. A., Couzin, I. D., Jackson, M. O., Jennings, N. R., Kamar, E., Kloumann, I. M., Larochelle, H., Lazer, D., McElreath, R., Mislove, A., Parkes, D. C., Pentland, A. S., … Wellman, M. (2019). Machine behavior. *Nature*, *568*(7753), 477–486. https://doi.org/10.1038/s41586-019-1138-y

*Ren, S., Cui, Z., Song, R., Wang, Z., & Hu, S. (2024). Emergence of social norms in generative agent societies: Principles and architecture. In K. Larson (Ed.), *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence* (pp. 7895–7903). International Joint Conferences on Artificial Intelligence Organization. https://doi.org/10.24963/ijcai.2024/874

*Rivera, J.-P., Mukobi, G., Reuel, A., Lamparth, M., Smith, C., & Schneider, J. (2024). Escalation risks from language models in military and diplomatic decision-making. *In Proceedings of the 2024 ACM Conference on Fairness, Accountability, and Transparency* (pp. 836–898). Association for Computing Machinery. https://doi.org/10.1145/3630106.3658942

Russell, S. J., & Norvig, P. (2021). *Artificial intelligence: A modern approach* (4th ed.). Pearson.

Saeedizade, M. J., & Blomqvist, E. (2024). Navigating ontology development with large language models. In A. Meroño Peñuela, A. Dimou, R. Troncy, O. Hartig, M. Acosta, M. Alam, H. Paulheim, & P. Lisena (Eds.), *The semantic web – ESWC 2024* (pp. 143–161). Springer. https://doi.org/10.1007/978-3-031-60626-7_8

*Sakamoto, Y., Uchida, T., & Ishiguro, H. (2025). Value-based large language model agent simulation for mutual evaluation of trust and interpersonal closeness. *Scientific Reports*, *15*, Article 41653. https://doi.org/10.1038/s41598-025-25531-1

*Sakurai, M., Ueta, K., & Hashimoto, Y. (2025). Exploring the limits of LLMs in simulating partisan polarization with confirmation bias prompts. *Engineering Proceedings*, *107*(1), Article 2. https://doi.org/10.3390/engproc2025107002

*Sarkar, B., Xia, W., Liu, C. K., & Sadigh, D. (2025). Training language models for social deduction with multi-agent reinforcement learning. In *Proceedings of the 24th International Conference on*

Autonomous Agents and Multiagent Systems (AAMAS 2025) (pp. 1830–1839). International Foundation for Autonomous Agents and Multiagent Systems. https://www.ifaamas.org/Proceedings/aamas2025/pdfs/p1830.pdf

Schick, T., Dwivedi-Yu, J., Dessì, R., Raileanu, R., Lomeli, M., Hambro, E., Zettlemoyer, L., Cancedda, N., & Scialom, T. (2023). Toolformer: Language models can teach themselves to use tools. *Advances in Neural Information Processing Systems, 36*, 68539–68551. https://doi.org/10.52202/075280-2997

Searle, J. R. (1995). *The construction of social reality*. Free Press.

Shanahan, M., McDonell, K., & Reynolds, L. (2023). Role play with large language models. *Nature, 623*(7987), 493–498. https://doi.org/10.1038/s41586-023-06647-8

Shoham, Y., & Tennenholtz, M. (1995). On social laws for artificial agent societies: *Off-line design. Artificial Intelligence*, 73(1–2), 231–252. https://doi.org/10.1016/0004-3702(94)00007-N

*Spangher, A., Lu, M., Kalyan, S., Cho, H. J., Huang, T., Shi, W., & May, J. (2025). NewsInterview: A dataset and a playground to evaluate LLMs' grounding gap via informational interviews. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (pp. 32895–32925). Association for Computational Linguistics. https://doi.org/10.18653/v1/2025.acl-long.1580

*Sreedhar, K., Cai, A., Ma, J., Nickerson, J. V., & Chilton, L. B. (2025). Simulating cooperative prosocial behavior with multi-agent LLMs: Evidence and mechanisms for AI agents to inform policy decisions. In *Proceedings of the 30th International Conference on Intelligent User Interfaces (IUI '25)* (pp. 1272–1286). Association for Computing Machinery. https://doi.org/10.1145/3708359.3712149

Tranfield, D., Denyer, D., & Smart, P. (2003). Towards a methodology for developing evidence-informed management knowledge by means of systematic review. *British Journal of Management*, 14(3), 207–222. https://doi.org/10.1111/1467-8551.00375

Tseng, Y.-M., Huang, Y.-C., Hsiao, T.-Y., Chen, W.-L., Huang, C.-W., Meng, Y., & Chen, Y.-N. (2024). Two tales of persona in LLMs: A survey of role-playing and personalization. In *Findings of the Association for Computational Linguistics: EMNLP 2024* (pp. 16612–16631). Association for Computational Linguistics. https://doi.org/10.18653/v1/2024.findings-emnlp.969

Uschold, M., & Grüninger, M. (1996). Ontologies: Principles, methods and applications. *The Knowledge Engineering Review, 11*(2), 93–136. https://doi.org/10.1017/S0269888900007797

*Vallinder, A., & Hughes, E. (2025). Cultural evolution of cooperation among LLM agents: Extended abstract. In Y. Vorobeychik, S. Das, & A. Nowe (Eds.), Proceedings of the 24th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2025) (pp. 2771–2773). International Foundation for Autonomous Agents and Multiagent Systems. https://www.ifaamas.org/Proceedings/aamas2025/pdfs/p2771.pdf

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. In *Advances in Neural Information Processing Systems* (Vol. 30, pp. 5998–6008). https://papers.nips.cc/paper/7181-attention-is-all-you-need

Wang, L., Ma, C., Feng, X., Zhang, Z., Yang, H., Zhang, J., Chen, Z., Tang, J., Chen, X., Lin, Y., Zhao, W. X., Wei, Z., & Wen, J.-R. (2024). A survey on large language model based autonomous agents. *Frontiers of Computer Science, 18*(6), Article 186345. https://doi.org/10.1007/s11704-024-40231-1

*Wang, L., Zhang, Z., & Chen, X. (2025). Investigating and extending Homans' social exchange theory with large language model based agents. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (pp. 9762–9777). Association for Computational Linguistics. https://doi.org/10.18653/v1/2025.acl-long.481

*Wang, L., Zhang, J., Yang, H., Chen, Z.-Y., Tang, J., Zhang, Z., Chen, X., Lin, Y., Sun, H., Song, R., Zhao, X., Xu, J., Dou, Z., Wang, J., & Wen, J.-R. (2025). User behavior simulation with large language model-based agents. *ACM Transactions on Information Systems, 43*(2), Article 55. https://doi.org/10.1145/3708985

Weber, M. (1978). *Economy and society: An outline of interpretive sociology* (G. Roth & C. Wittich, Eds.). University of California Press. (Original work published 1922)

Wei, A., Haghtalab, N., & Steinhardt, J. (2023). Jailbroken: How does LLM safety training fail? *Advances in Neural Information Processing Systems, 36*, 80079–80110. https://papers.nips.cc/paper_files/paper/2023/hash/fd6613131889a4b656206c50a8bd7790-Abstract-Conference.html

Wei, J., Wang, X., Schuurmans, D., Bosma, M., Ichter, B., Xia, F., Chi, E. H., Le, Q. V., & Zhou, D. (2022). Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems, 35*, 24824–24837. https://proceedings.neurips.cc/paper_files/paper/2022/hash/9d5609613524ecf4f15af0f7b31abca4-Abstract-Conference.html

Woodgate, J., & Ajmeri, N. (2024). Macroethics principles for responsible AI systems: Taxonomy and directions. *ACM Computing Surveys*, *56*(11), Article 289. https://doi.org/10.1145/3672394

*Wu, Z., Peng, R., Zheng, S., Liu, Q., Han, X., Kwon, B. I., Onizuka, M., Tang, S., & Xiao, C. (2024). Shall we team up: Exploring spontaneous cooperation of competing LLM agents. In *Findings of the Association for Computational Linguistics: EMNLP 2024* (pp. 5163–5186). Association for Computational Linguistics.https://doi.org/10.18653/v1/2024.findings-emnlp.297

Xi, Z., Chen, W., Guo, X., He, W., Ding, Y., Hong, B., Zhang, M., Wang, J., Jin, S., Zhou, E., Zheng, R., Fan, X., Wang, X., Xiong, L., Zhou, Y., Wang, W., Jiang, C., Zou, Y., Liu, X., … Gui, T. (2025). The rise and potential of large language model based agents: *A survey. Science China Information Sciences, 68*(2), Article 121101. https://doi.org/10.1007/s11432-024-4222-0

*Xie, C., Chen, C., Jia, F., Ye, Z., Lai, S., Shu, K., Gu, J., Bibi, A., Hu, Z., Jurgens, D., Evans, J., Torr, P., Ghanem, B., & Li, G. (2024). Can large language model agents simulate human trust behavior? In *Advances in Neural Information Processing Systems* (Vol. 37). https://proceedings.neurips.cc/paper_files/paper/2024/hash/1cb57fcf7ff3f6d37eebae5becc9ea6d-Abstract-Conference.html

*Xu, B., Zhao, S., Wu, R., Huang, Z., Wang, J., Hu, Z., Wang, K., Liu, H., Lv, T., Li, L., Fan, C., Tong, X., & Han, J. (2025). Empowering economic simulation for massively multiplayer online games through generative agent-based modeling. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining* (Vol. 2, pp. 3366–3377). Association for Computing Machinery. https://doi.org/10.1145/3711896.3736929

*Xu, S., & Zhong, F. (2025). CoMet: Metaphor-driven covert communication for multi-agent language games. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics* (Volume 1: Long Papers) (pp. 7892–7917). Association for Computational Linguistics. https://doi.org/10.18653/v1/2025.acl-long.389

*Xu, Z., Wang, J., Hu, B., Wang, L., & Zhang, M. (2025). MeKB-Sim: Personal knowledge base-powered multi-agent simulation. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (System Demonstrations)* (pp. 393–403). Association for Computational Linguistics. https://doi.org/10.18653/v1/2025.naacl-demo.33

*Yang, S., Ceferino, L., Zhang, Y., Gu, C., Guo, T., & Kondo, G. (2026). When agents learn to think: Large language model-enhanced agent-based modeling for crowd evacuation in disaster

scenarios. *Reliability Engineering & System Safety, 269*, Article 112056. https://doi.org/10.1016/j.ress.2025.112056

*Zeng, Y., Brown, C., Byari, M., Raymond, J., Schmitt, T., & Rounsevell, M. (2025). InsNet-CRAFTY v1.0: Integrating institutional network dynamics powered by large language models with land use change simulation. *Geoscientific Model Development*, *18*(15), 4983–5013. https://doi.org/10.5194/gmd-18-4983-2025

*Zhou, X., Su, Z., Feng, S., Zhou, J., Huang, J.-T., Kao, H.-T., Lynch, S., Volkova, S., Wu, T., Woolley, A., Zhu, H., & Sap, M. (2025). SOTOPIA-S4: A user-friendly system for flexible, customizable, and large-scale social simulation. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (System Demonstrations)* (pp. 350–360). Association for Computational Linguistics. https://doi.org/10.18653/v1/2025.naacl-demo.30

Appendices

Appendix 1. Overview of the 38 articles coded for the ontology

Table A1.1 lists the 38 studies that passed the screening and quality assessment stages of the systematic literature review and that were coded into the ontology described in Chapter 3. For each study, the table records the reference, the article title and its publication venue, the primary simulation domain(s) under which the study was coded, and the number of distinct role conditions (RCs) extracted from it. Domain labels follow the B1.Domain field of the ontology. A single study can contribute multiple role conditions when it contrasts several agent specifications within the same experimental design.

Table A1.1. Overview of the 38 articles coded for the ontology. Reference, title with publication venue, primary simulation domain, and number of role conditions.

ID	Reference	Title [Publication venue]	Primary domain (RCs)
1	Piatti et al. (2024)	Cooperate or collapse: Emergence of sustainable cooperation in a society of LLM agents *[NeurIPS 2024]*	Tragedy of the commons (4 RCs)
2	Abdelnabi et al. (2024)	Cooperation, competition, and maliciousness: LLM-stakeholders interactive negotiation *[NeurIPS 2024]*	Negotiation (non-zero-sum) (3 RCs)
3	Yang et al. (2026)	When agents learn to think: Large language model-enhanced agent-based modeling for crowd evacuation in disaster scenarios *[Reliability Engineering & System Safety]*	Disaster evacuation (1 RC)
4	Liu et al. (2026)	Generative agents for urban mobility: A cognitive framework for realistic travel behavior simulation *[Simulation Modelling Practice and Theory]*	Urban mobility (1 RC)
5	Großmann et al. (2026)	The power of stories: Narrative priming in networked multi-agent LLM interactions *[NETGCOOP 2025, LNCS 16173 (Springer)]*	Public goods game (5 RCs)
6	Sakamoto et al. (2025)	Value-based large language model agent simulation for mutual evaluation of trust and interpersonal closeness *[Scientific Reports]*	Interpersonal trust (1 RC)

ID	Reference	Title [Publication venue]	Primary domain (RCs)
7	Cau et al. (2025)	Selective agreement, not sycophancy: Investigating opinion dynamics in LLM interactions *[EPJ Data Science]*	Opinion dynamics (1 RC)
8	Buscemi et al. (2025)	FAIRGAME: A framework for AI agents bias recognition using game theory *[ECAI 2025]*	Strategic game theory (2 RCs)
9	Zeng et al. (2025)	InsNet-CRAFTY v1.0: Integrating institutional network dynamics powered by large language models with land use change simulation *[Geoscientific Model Development]*	Institutional land-use policy (4 RCs)
10	Xu et al. (2025)	Empowering economic simulation for massively multiplayer online games through generative agent-based modeling *[KDD 2025]*	MMO game economy (1 RC)
11	Akata et al. (2025)	Playing repeated games with large language models *[Nature Human Behavior]*	Strategic game theory (1 RC)
12	Li et al. (2025)	MetaAgents: Large language model based agents for decision-making on teaming *[PACM HCI (CSCW 2025)]*	Team assembly (2 RCs)
13	Sreedhar et al. (2025)	Simulating cooperative prosocial behavior with multi-agent LLMs: Evidence and mechanisms for AI agents to inform policy decisions *[IUI 2025]*	Public goods game / social simulation (4 RCs)
14	Wang et al. (2025)	User behavior simulation with large language model-based agents *[ACM Transactions on Information Systems]*	Recommender social simulation (1 RC)
15	Mao et al. (2025)	ALYMPICS: LLM agents meet game theory. *[COLING 2025]*	Auction game (2 RCs)
16	Sakurai et al. (2025)	Exploring the limits of LLMs in simulating partisan polarization with confirmation bias prompts *[Engineering Proceedings (MDPI)]*	Opinion dynamics (2 RCs)
17	Wang et al. (2025)	Investigating and extending Homans' social exchange theory with large language model based agents *[ACL 2025]*	Social exchange (2 RCs)

ID	Reference	Title [Publication venue]	Primary domain (RCs)
18	Xu et al. (2025)	CoMet: Metaphor-driven covert communication for multi-agent language games *[ACL 2025]*	Social deduction / adversarial language game (3 RCs)
19	Spangher et al. (2025)	NewsInterview: A dataset and a playground to evaluate LLMs' grounding gap via informational interviews *[ACL 2025]*	Strategic dialogue (2 RCs)
20	Ma et al. (2025)	Do androids question electric sheep? A multi-agent cognitive simulation of philosophical reflection on hybrid table reasoning *[ACL 2025 SRW]*	Hybrid table reasoning (2 RCs)
21	Kim et al. (2025)	Fine-grained and thematic evaluation of LLMs in social deduction game *[IEEE Access]*	Social deduction game (2 RCs)
22	Vallinder & Hughes (2025)	Cultural evolution of cooperation among LLM agents *[AAMAS 2025 (Extended Abstract); arXiv:2412.10270]*	Strategic game theory (2 RCs)
23	Sarkar et al. (2025)	Training language models for social deduction with multi-agent reinforcement learning *[AAMAS 2025 (Extended Abstract); arXiv:2502.06060]*	Social deduction game (2 RCs)
24	Mouri Zadeh Khaki et al. (2025)	Simulating social behavior of LLM-based autonomous negotiator agents in a game-theoretical framework using multi-agent systems *[International Journal of Human–Computer Interaction]*	Negotiation (5 RCs)
25	Rivera et al. (2024)	Escalation risks from language models in military and diplomatic decision-making *[ACM FAccT 2024]*	Wargame simulation (4 RCs)
26	Lan et al. (2024)	LLM-based agent society investigation: Collaboration and confrontation in Avalon gameplay *[EMNLP 2024]*	Social deduction game (2 RCs)
27	Huang & Hadfi (2024)	How personality traits influence negotiation outcomes? A simulation based on large language models *[Findings of EMNLP 2024]*	Negotiation (2 RCs)
28	Jin et al. (2024)	AgentReview: Exploring peer review dynamics with LLM agents *[EMNLP 2024]*	Academic peer review (9 RCs)
29	Wu et al. (2024)	Shall we team up: Exploring spontaneous cooperation of competing LLM agents *[Findings of EMNLP 2024]*	Strategic game theory / competitive

ID	Reference	Title [Publication venue]	Primary domain (RCs)
			market/evacuation (3 RCs)
30	Ren et al. (2024)	Emergence of social norms in generative agent societies: Principles and architecture *[IJCAI 2024]*	Social simulation (2 RCs)
31	Bianchi et al. (2024)	How well can LLMs negotiate? NegotiationArena platform and analysis *[ICML 2024]*	Negotiation / strategic game theory (3 RCs)
32	Chuang et al. (2024)	Simulating opinion dynamics with networks of LLM-based agents *[Findings of NAACL 2024]*	Opinion dynamics (4 RCs)
33	Xie et al. (2024)	Can large language model agents simulate human trust behavior? *[NeurIPS 2024]*	Interpersonal trust / game theory (3 RCs)
34	Guan et al. (2024)	Richelieu: Self-evolving LLM-based agents for AI diplomacy *[NeurIPS 2024]*	Strategic diplomacy (1 RC)
35	Park et al. (2023)	Generative agents: Interactive simulacra of human behavior *[UIST 2023]*	Social simulation (1 RC)
36	Zhou et al. (2025)	SOTOPIA-S4: A user-friendly system for flexible, customizable, and large-scale social simulation *[NAACL 2025 (System Demonstrations)]*	Social simulation / negotiation (1 RC)
37	Xu et al. (2025)	MeKB-Sim: Personal knowledge base-powered multi-agent simulation *[NAACL 2025 (System Demonstrations)]*	Social simulation (1 RC)
38	Li et al. (2023)	CAMEL: Communicative agents for 'mind' exploration of large language model society *[NeurIPS 2023]*	Cooperative task solving (2 RCs)

Appendix 2. Supplementary Materials

The following online resources were developed as part of this research and are referenced throughout the thesis. They complement the static content of the printed document with live, queryable, or interactive versions of the research data. All resources were accessible as of April 2026. Readers are advised to consult the Zenodo archive (Resource D) for stable, permanently citable versions of the underlying data.

Resource A: SLR Screening Sheet
URL:
https://docs.google.com/spreadsheets/d/1xN1NOSkuGSy0l6WRPOl13dCZt1p4ZPEpVQTFqE9yzm0/edit

Accessed: April 2026

This spreadsheet served as the primary screening instrument for the systematic literature review described in Chapter 2 and Appendix 3. The workbook contains five dedicated tabs: (1) Process Map, documenting the overall screening workflow; (2) Stage 1 Screening, recording title and abstract decisions and scores for all 724 deduplicated records; (3) Stage 2 Screening, recording full-text eligibility decisions and exclusion reasons for the 210 papers forwarded from Stage 1; (4) Quality Assessment, scoring the 41 papers that passed full-text screening against the five QA criteria; and (5) reviewer notes. All inclusion and exclusion decisions, scores, and rationales are recorded here, including the inter-rater reliability calculations reported in the SLR protocol Appendix 3.

Resource B: Ontology Data Sheet

URL: https://docs.google.com/spreadsheets/d/1e1-g2dR2RaFHiTtfeL3oAHJY7fFG1X5d/edit

Accessed: April 2026

The Ontology Data Sheet is the primary empirical output of this thesis. It contains the full coding of the 38 papers in the final corpus (two papers were excluded during coding for insufficient evidence), organised around the four-layer coding schema described in Chapter 2. The workbook includes: individual paper tabs coded as N_author_year (e.g., 1_piatti2024); a

MASTER_SYNTHESIS sheet aggregating coded values across all 38 studies, with columns covering role type (A1–A4), normative level, domain (B1), outcomes (C1, C3), and failure modes (D1); and a TRIPLES sheet containing all 516 ontology triples extracted from the corpus. This document is the basis for all analyses reported in Chapter 3 and the findings interpreted in Chapter 4. Readers wishing to inspect specific coding decisions or cross-check claims against raw data should consult this sheet directly.

Resource C: Research Companion Page and Interactive Knowledge Graph

URL: https://technejad.github.io/MA-thesis-ongoing/

Accessed: April 2026

This publicly accessible web page provides supplementary materials for the thesis, including an interactive visualization of the ontology knowledge graph. The interactive graph allows readers to explore the ontology's entity classes and relation types, filter by simulation domain, and navigate the structural map of normative role design patterns identified in the corpus. The visualization complements the static figures presented in Chapter 3 by enabling direct, queryable engagement with the full ontology structure. Additional materials and future updates may be available at this address beyond those included in the static thesis document.

Resource D: Zenodo Archive

DOI: 10.5281/zenodo.19844930

Accessed: 28 April 2026

This deposit provides permanent, citable archiving of the supplementary materials for this thesis. It contains a static .xlsx export of the Ontology Data Sheet (Resource B) and a static .xlsx export of the SLR Screening Sheet (Resource A). The concept DOI above resolves to the latest version of the deposit.

Appendix 3. Systematic Literature Review Protocol

Note: All screening decisions and reviewer notes are recorded in the SLR Screening Sheet (see Appendix 2, Resource A).

1. Introduction and Objective

This document specifies the protocol for the systematic literature review (SLR) conducted as the primary data collection method for the master's thesis. The SLR follows established guidelines for systematic reviews in software engineering and information systems (Kitchenham et al., 2015; Tranfield et al., 2003), adapted for a rapidly evolving and interdisciplinary field.

The purpose of this SLR is not to produce a standalone review article, but to systematically identify, select, and assess the corpus of multi-agent LLM simulation studies from which an ontology of normative role design will be constructed. The review is therefore instrumental: it serves the ontology construction by ensuring that the studies forming its empirical basis are selected through a rigorous, transparent, and reproducible process.

1.1 Research Questions

The main research question is: **How are normative roles designed and operationalized across different applications of multi-agent LLM simulations?**

This is addressed through five sub-questions:

ID	Sub-Question
RQ1	What types of normative roles are designed across multi-agent LLM systems?
RQ2	What prompting methods and normative framings operationalize these roles?
RQ3	How do the designed roles relate to individual agent behavior and collective simulation outcomes?
RQ4	Where and how do the designed roles fail to produce intended behavior and outcomes?
RQ5	How does the simulation domain mediate the normative design of agent roles?

2. Conceptual Scope and Key Definitions

The SLR targets studies at the intersection of a specific conceptual chain: **LLM → System Prompt → Role-Play → LLM Agent → Multi-Agent System → Normative Roles → Ontology**. The following definitions clarify the scope and boundaries of the review.

2.1 Large Language Models and System Prompts

Large Language Models (LLMs) are autoregressive language models that generate text by predicting the next token in a sequence (Brown et al., 2020). System prompts are natural language instructions provided to an LLM at inference time that set the behavioral context. When placed in a turn-taking architecture with a system prompt, the LLM engages in what Shanahan et al. (2023) call role-play: continuing the pattern defined by the prompt and its training data. This role-playing characteristic is the native mode of LLM operation.

2.2 LLM Agents and Multi-Agent Systems

LLM agents combine foundational language models with an orchestrator: software that connects the LLM with tools, memory, and external APIs (Wang et al., 2024; Xi et al., 2025). Multi-Agent Systems (MAS) deploy multiple such agents in shared environments, enabling the study of emergent behavior, negotiation, cooperation, and collective dynamics (Bianchi et al., 2024; Piatti et al., 2024).

2.3 Normative Roles: Scope and Theoretical Grounding

A central challenge in this review is distinguishing *normative role design* from *individual behavioral modification*. Many LLM simulation studies assign agents personas or decision heuristics, but these shape individual behavior without involving social orientation or shared expectations between agents. This thesis uses “normative” rather than “ethical” to capture the broader sociological sense: not just moral principles, but strategic directives, contextual rules, and domain conventions that regulate conduct through mutual expectation (Brennan et al., 2013). Drawing on Weber (1922/1978), Bicchieri (2006), Searle (1995), and the OMNI framework (Dignum et al., 2005), an agent’s role qualifies as genuinely normative if it meets the following five criteria:

(1) Social reference to other agents or community standards; **(2) Prescriptive framing** using "should," "must," or "expected to"; **(3) Flexibility** allowing for meaningful non-compliance; **(4) Conditionality** on beliefs about others' behavior; and **(5) Social standing implications** for violation.

The full theoretical framework is detailed in the thesis literature review.

3. Search Strategy

3.1 Databases

Two electronic databases were selected for the systematic search:

Database	Rationale
Scopus	Scopus, for its comprehensive coverage of CS/AI venues, including major conferences (NeurIPS, ICML, AAAI, ICLR)
Web of Science	For its established role in systematic reviews and its cross-disciplinary citation coverage.

Other databases (IEEE Xplore, ACM Digital Library) were considered but not used separately; see the limitations at the end for more details.

3.2 Search Period

The search period was defined as **2017 – Present**. The starting year was selected to coincide with the introduction of the Transformer architecture (Vaswani et al., 2017), which serves as the foundational technology for modern LLMs. Literature published before 2017 on "normative agents" predominantly relies on symbolic AI or traditional reinforcement learning, which falls outside the scope of this review.

The vast majority of relevant publications cluster in 2024–2025, reflecting the recency of this research area. The final search was executed in January 2026.

3.3 Search Terminology

Search terms were derived from the conceptual chain, research questions, and iterative testing against known relevant studies. The following term clusters were used:

Category	Primary Terms	Alternatives
Core Technology	“large language model”, “LLM”	“language model”, “GPT”
Agent Framing	“agent”, “LLM agent”, “multi-agent”	“generative agent”, “autonomous agent”, “agency”
Role Design	“role”, “persona”	“profile”, “character”, “social behavior”
Domain / Setting	“simulation”, “game”, “negotiation”	“cooperation”, “collaboration”, “competition”
Normative Content	“normative”, “ethical”	“moral”, “norm”

Design decisions on excluded terms: “Role-play” appears in theoretical literature (Shanahan et al., 2023) but rarely in simulation papers and was therefore excluded. “System prompt” was excluded because papers use diverse variants (“dialogue prompt,” “prompt”), and the bare term “prompt” produces unmanageably broad results. “Normative” and “ethical” rarely appear in titles or abstracts of relevant simulation papers; normative content was therefore screened manually during title/abstract review rather than enforced at the query level.

3.4 Search Queries

Scopus:

TITLE-ABS-KEY ((LLM OR “language model*”) AND (agent*) AND (simulation* OR game*) AND (behaviour* OR behavior* OR social OR interaction* OR “agency”)) AND PUBYEAR > 2016 AND PUBYEAR < 2027 AND (LIMIT-TO (LANGUAGE, “English”)) AND (EXCLUDE (SRCTYPE, “b”))*

Results: **714 documents**

Web of Science:

TS=((“LLM” OR “language model*”) AND (“agent*”) AND (“simulation*” OR “game*”) AND (“behaviour*” OR “behavior*” OR “social” OR “interaction*” OR “agency”))*

Refined by: Publication Years 2017–2026.

Results: **342 documents**

Combined total: **1,056 records** imported into Zotero 7 for duplicate management.

4. Duplicate Removal

All records from both databases were exported to **Zotero 7** and organized into database-specific sub-collections (Scopus, Web of Science). Duplicate detection was performed using the **Zoplicate plugin** (community standard for Zotero 7), configured with: Action = Keep Old (preserves original record); Master Item = Earliest Added (ensures audit trail).

The duplicate view identified **654 items** forming approximately 327 duplicate pairs. After merging, the library was reduced from **1,056 to 724 unique records** (332 duplicates removed; slight variance from the 327 estimate is due to triplicate copies). The deduplicated library was exported as CSV (UTF-8 encoding) for screening.

5. Eligibility Criteria

The eligibility criteria were designed as a progressive filtering funnel, moving from basic publication requirements through technical scope to the normative and analytical core of the review. Criteria are divided into inclusion criteria (IC1–IC6) applied at the query and first screening stages, and exclusion criteria (EC1–EC9) applied at the first and second screening stages.

5.1 Inclusion Criteria

ID	Category	Description	Stage
IC1	Technology	Agents are architecturally driven by Large Language Models (LLMs).	Query + Stage 1
IC2	System Architecture	The system features multi-agent dynamics (2+ autonomous LLM agents) rather than single-agent or human–AI interaction.	Query + Stage 1

IC3	Interactive Context	Agents directly interact within a shared environment, game, or scenario (not isolated parallel tasks).	Query + Stage 1
IC4	Outcomes	The study analyses how LLM agents make decisions, interact, and produce collective outcomes (not technical performance benchmarks or model capability tests).	Query + Stage 1
IC5	Timeframe	Publication date from 2017 to the present. Rationale: coincides with the Transformer architecture (Vaswani et al., 2017).	Query filter
IC6	Language	Full text is available in English.	Query filter

5.2 Exclusion Criteria

ID	Category	Description	Stage
EC1	Publication Type	Surveys, review papers, editorials, workshop abstracts, posters, or entire conference proceedings volumes.	Stage 1 & 2
EC2	Conceptual vs. Experimental	Purely conceptual or theoretical studies that do not include a simulation experiment.	Stage 1 & 2
EC3	Agent Type	Non-LLM agents (rule-based models, symbolic AI, pure reinforcement learning).	Stage 1 & 2
EC4	MAS	Studies involving a single LLM agent or human–LLM interactions without multi-agent system dynamics.	Stage 1 & 2
EC5	Interaction Type	Agents interact only indirectly through intermediary mechanisms (market prices, mathematical functions, system aggregates, order books) with no direct communication or interaction.	Stage 1 & 2

EC6	Object of Study	The multi-agent system is merely a technical tool or instrument for generating synthetic data, benchmarking models, or optimisation.	Stage 1 & 2
EC7	System Prompt / Role Access	The study does not provide the full system prompt or an equivalent complete description (e.g., in an appendix or a public repository), preventing direct analysis of role design.	Stage 2
EC8	Normative Roles	Roles are purely functional or task-based with no reference to social expectations, obligations, or behavioral demands toward other agents. (See thesis Literature Review for full theoretical framework.)	Stage 2
EC9	Outcome Analysis	The study does not analyse how different roles relate to individual agent behavior and collective simulation outcomes.	Stage 2

5.3 Criteria Design Logic

EC1 restricts the corpus to primary research, excluding surveys, reviews, and editorials. EC2–EC4 build towards the topic: the review targets experimental studies deploying LLM agents in multi-agent simulations. EC5 requires direct interaction because normative roles orient agents toward one another; if agents interact only through market signals or system aggregates, there is no observable social behavior to analyze normative role design from.

EC6 distinguishes studies that treat multi-agent systems as objects of behavioral inquiry from those that use them as instruments for generating synthetic data, benchmarking models, or optimizing outputs. This review retains only the former, because the thesis examines how role design shapes what agents do, following Rahwan et al. (2019), who argue that AI systems can and should be studied as behavioral subjects in their own right.

EC7 requires access to the full system prompt or equivalent, because prompts are the primary unit of analysis for ontology construction; without them, role design cannot be examined directly. EC8 excludes studies whose roles are entirely functional, carrying no reference to social expectations, obligations, or behavioral orientation toward other agents. Studies that combine functional roles with normatively loaded ones were retained; the functional roles in such studies

were coded as baseline role types in the ontology (§3.2). EC9 ensures retained studies analyze how roles relate to agent behavior and simulation outcomes, mapping directly to RQ3.

6. Screening Process

6.1 Screening Procedure and Reviewer Roles

All screening was conducted by the primary reviewer (MHN), with the thesis supervisor independently scoring a sample at each stage to establish inter-rater reliability (see Section 6.4). The screening instrument was a purpose-built Excel workbook with dedicated sheets for each phase, recording all scores, exclusion reasons, and reviewer notes.

6.2 Stage 1: Title and Abstract Screening

Goal: Remove clearly irrelevant papers based on title, abstract, and metadata.

Input: 724 deduplicated records.

Scoring: Each paper was scored on a three-point scale: 0 = exclude (clearly out of scope), 0.5 = uncertain (possible relevance, requires full-text review), 1 = include (clearly relevant). All papers scoring 0.5 or 1.0 were forwarded to Stage 2.

Criteria applied: IC1–IC4 (checked against title and abstract); EC1–EC6 (applied where determinable from title/abstract).

Stage 1 Results:

Category	Count
Total screened	724
Excluded (score = 0)	514
Uncertain (score = 0.5)	65
Included (score = 1)	145
Forwarded to Stage 2	**210**

6.3 Stage 2: Full-Text Screening

Goal: Confirm eligibility by reviewing the full text. Screening follows a stop-at-first-violation logic: each paper is assessed against EC1–EC9 in sequence, and the first applicable exclusion criterion is recorded as the reason for exclusion.

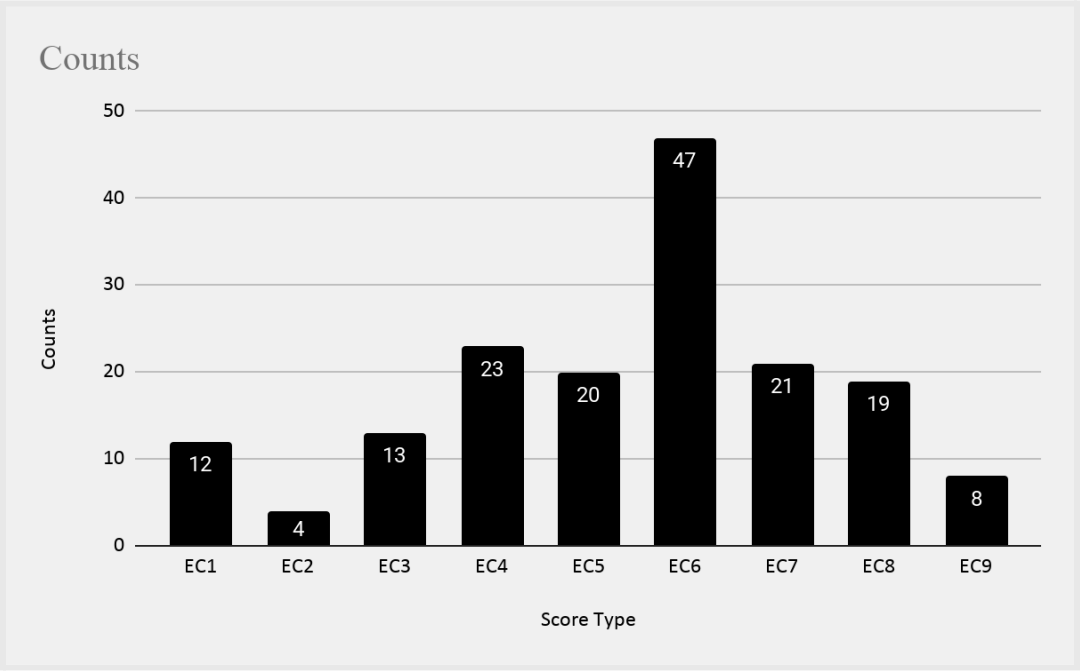
Input: 210 papers forwarded from Stage 1.

Criteria applied: Full set IC1–IC6 and EC1–EC9, with EC7–EC9 applied exclusively at this stage (requiring access to full text, system prompts, and repositories).

Stage 2 Results:

Category	Count
Total reviewed (full text)	210
Excluded	169
Included (forwarded to QA)	**41**

Exclusion reasons (Stage 2):



6.4 Inter-Rater Reliability

All screening was conducted by a single primary reviewer (MHN). To assess reliability, a subset of 126 papers was independently screened by the supervisor. Both reviewers scored each paper on a three-point scale (0 = exclude, 0.5 = uncertain, 1 = include). For the purposes of calculating inter-rater reliability, scores were binarized, with uncertain scores (0.5) treated as inclusions. This aligns with the conservative screening strategy, in which borderline papers were always carried forward. Inter-Rater Reliability (IRR) was measured using Cohen's Kappa. The two reviewers agreed on 78 of 126 papers (61.90%), yielding a Cohen's Kappa of $\kappa = 0.317$, which falls in the "fair agreement" range (Landis & Koch, 1977).

The low kappa partly understates the actual level of agreement. Of the 48 disagreements, 42 involved papers where one or both reviewers scored 0.5 (uncertain) rather than giving a definitive include or exclude. The disagreements, in other words, were not about whether a paper was relevant, but about how much uncertainty to tolerate, and since all uncertain papers were carried forward regardless, these disagreements had little effect on the final corpus. When limited to the 70 papers where both reviewers gave definitive scores (0 or 1), agreement rose to 91.4% with $\kappa = 0.83$, which Landis and Koch classify as "almost perfect." The difference in overall base rates between the two reviewers (MHN included 36.5%; the supervisor included 71.5%) further suppresses kappa arithmetically, a known limitation of the measure when reviewers apply different thresholds.

Given that the substantive disagreements were minimal and the screening strategy was designed to absorb borderline uncertainty, this level of agreement was considered acceptable for the review. The complete SLR sheet is available in Appendix 2, Resource A.

7. Quality Assessment

All 41 papers that passed full-text screening were subjected to a quality assessment (QA) to evaluate their internal and external validity. The QA used five questions, scored on a three-point scale: 0 (no), 0.5 (partial), 1 (yes). The threshold for inclusion was a total score of ≥3 out of 5, with QA2 (methodological description) as a mandatory pass.

ID	Question

QA1	Are the objectives and the context of the research clear and related to multi-agent LLM simulations?
QA2	Does the study clearly describe the simulation methodology, including agent setup, interaction setting, and evaluations? (Mandatory pass)
QA3	Does the study provide an explicit rationale or theoretical motivation (e.g., prior models, literature, or design assumptions) for the agent roles and simulation setup?
QA4	Are the analyses and interpretations coherent with the simulation design?
QA5	Are the study's contributions and limitations clearly stated?

Quality Assessment Results:

Category	**Count**
Total assessed	41
Passed (≥3/5)	40
Failed (< 3/5)	1
Final corpus for ontology construction	**38**

8. Summary of Data Extraction and Coding Schema for Ontology

The 40 studies that passed quality assessment entered the second phase of the methodology: ontology construction. During data extraction, two studies were excluded because their role conditions could not be studied from the published text, appendices, or repositories, reducing the coded corpus to 38 studies.

The unit of analysis is the role condition (RC), defined as any distinct agent type, persona, or normative framing that receives its own prompt or behavioral specification within a study. Across the 38 coded studies, 93 role conditions were identified and individually coded using a four-layer schema: normative role design (Layer A), simulation setup (Layer B), outcome analysis (Layer C), and failure modes (Layer D). System prompts are treated as the primary empirical evidence, separate from the paper's textual descriptions, because papers sometimes describe roles differently from what is actually implemented in the prompt.

The coded output was formalised into a conceptual ontology following a competency question-driven development approach (Grüninger & Fox, 1995; Uschold & Grüninger, 1996). The complete ontology development methodology is detailed in §2.2 of the main thesis body.

9. Review Flow Summary

The table below summarises the complete review flow from database search to final corpus. A visual representation of this flow appears as Figure 3 in the main thesis body.

Stage	Action	n	Notes
Identification	Database search (Scopus + WoS)	1,056	Scopus: 714; WoS: 342
Deduplication	Zotero 7 / Zotero plugin	724	332 duplicates removed
Stage 1 Screening	Title & abstract review	724 → 210	514 excluded (score = 0)
Stage 2 Screening	Full-text review	210 → 41	169 excluded (EC1–EC9 applied)
Quality Assessment	Five-question QA	41 → 40	1 failed QA threshold
Final corpus	**Ontology construction**	**38**	

10. Tools and Data Management

Reference Management: Zotero 7 with the Zoplicate plugin for duplicate detection. Records are organised in database-specific sub-collections (Scopus, WoS) within a master SLR library.

Screening Instrument: Custom spreadsheet with dedicated sheets for each phase: Process Map, Stage 1 Screening, Stage 2 Screening, Quality Assessment. All screening decisions, scores, exclusion reasons, and reviewer notes are recorded in this document (see Appendix 2, Resource A).

AI-Assisted Verification: Claude (Anthropic), ChatGPT (OpenAI), Gemini (Google), and Grok (xAI) were used during screening for cross-checking assistance after the author's own screening and assessment, helping identify possible oversights or inconsistencies in the application of predefined exclusion criteria (see the AI disclaimer at the beginning of the main thesis document for details).

11. Limitations

Search coverage: The search drew on two databases: Scopus for its broad coverage of CS/AI conference venues, and Web of Science for cross-validation and citation analysis. IEEE Xplore and ACM Digital Library were not searched separately because their relevant content is substantially indexed through these two sources. Google Scholar was excluded for its lack of reproducible query syntax and export limitations. ArXiv was excluded because it hosts non-peer-reviewed preprints. These choices produce a corpus weighted toward peer-reviewed CS/AI venues in English. Work in sociology, political science, non-Anglophone computational research, and the fast-moving preprint literature is under-represented. The terminological diversity of this interdisciplinary field means some relevant papers may use vocabulary not covered by the search strings; normative content was therefore screened manually during title/abstract review rather than enforced at the query level.

Single primary reviewer: Screening was conducted primarily by a single reviewer (MHN), with the thesis supervisor independently scoring a sample of 126 papers at Stage 1 to establish inter-rater reliability (see Section 6.4). This partial reliability check falls short of the independent dual-review standard of a fully rigorous SLR, though it is standard formaster 's-level research.

Publication bias and prompt opacity: The corpus draws only on published studies, so role designs that failed, were abandoned, or produced null results are absent by construction. Additionally, EC7 (requiring access to full prompt specifications) was the second most common exclusion reason at Stage 2, applying to 21 papers. Many studies in this field describe agent designs without publishing the system prompts that define those roles. The review can only analyse studies whose role conditions are transparently provided, which likely biases the corpus toward more methodologically transparent research groups.

Rapidly evolving field: Multi-agent LLM research is growing rapidly, with the majority of relevant publications appearing in 2024–2025. The final search was executed in January 2026. Between that date and thesis submission, additional work has appeared whose inclusion would extend the corpus.