HORAE: A Domain-Agnostic Language for Automated Service Regulation*

Yutao Sun[1], **Mingshuai Chen**[1(✉)], **Tiancheng Zhao**[2(✉)], **Kangjia Zhao**[1], **He Li**[1], **Jintao Chen**[1], **Zhongyi Wang**[1], **Liqiang Lu**[1], **Xinkui Zhao**[1], **Shuiguang Deng**[1] and **Jianwei Yin**[1(✉)]
[1]Zhejiang University, Hangzhou 310027, China
[2]Binjiang Institute of Zhejiang University, Hangzhou 310053, China
{m.chen, zjuyjw}@zju.edu.cn, tianchez@zju-bj.com

Abstract

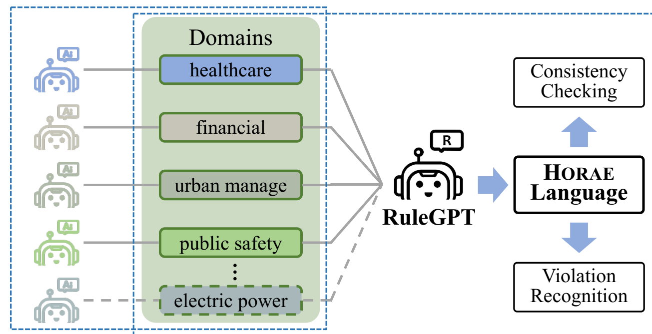
Artificial intelligence is rapidly encroaching on the field of service regulation. However, existing AI-based regulation techniques are often tailored to specific application domains and thus are difficult to generalize in an automated manner. This paper presents HORAE, a unified specification language for modeling (multimodal) regulation rules across a diverse set of domains. We showcase how HORAE facilitates an intelligent service regulation pipeline by further exploiting a fine-tuned large language model named RuleGPT that automates the HORAE modeling process, thereby yielding an end-to-end framework for fully automated intelligent service regulation. The feasibility and effectiveness of our framework are demonstrated over a benchmark of various real-world regulation domains. In particular, we show that our open-sourced, fine-tuned RuleGPT with 7B parameters suffices to outperform GPT-3.5 and perform on par with GPT-4o.

1 Introduction

Service regulation aims to determine whether services are delivered per established norms, rules, and/or standards within a specific context. The rapid advancements in the realm of artificial intelligence (AI) – particularly breakthroughs in deep neural networks and the swift rise of large language models (LLMs) – have triggered a recent surge of interest in *intelligent service regulation*. Employing AI in service regulation may substantially improve the degree of automation and accuracy, thereby yielding a significant cost reduction.

Current AI-based regulation methods predominantly adopt a *plug-and-play* approach: As illustrated in Fig. 1 (a), regulation industries encompass a wide spectrum of *scenarios* (aka *domains*, e.g., healthcare and financial services). A common practice is to train a distinct model that caters to a specific scenario, e.g., models for urban management [Kaginalkar *et al.*, 2021] and e-commerce [Raji *et al.*, 2024].

The plug-and-play method, however, suffers from two major issues: (i) *significant resource wastage*: the training and deployment of multiple large-scale AI models tailored for various scenarios necessarily incur a model proliferation and thereby substantial computing power consumption and carbon emissions [Luccioni *et al.*, 2023]; and (ii) *confined adaptability and efficiency*: the procedure of building and training models relies heavily on domain-specific knowledge (e.g., datasets, pre-trained models, and model architectures) of each scenario and is thus difficult to automate for general use.



(a) plug-and-play methods (b) HORAE architecture

Figure 1: Conventional plug-and-play methods are often confined to distinct models for specific domains, thus requiring extensive retraining and resource expenditure. In contrast, HORAE acts as a unified specification language to model regulation rules in a domain-agnostic fashion.

In response to these challenges, we propose HORAE – a unified specification language to model regulation rules in a *domain-agnostic* fashion. HORAE leverages the *zero-shot understanding* capability of LLMs [Wei *et al.*, 2022] to translate regulation rules from any scenario into a structured *intermediate representation* (IR); see Fig. 1 (b). This representation dissects complex behavior patterns across different domains into a set of fine-grained, readily-detectable events and actions. Consequently, the downstream recognition models and algorithms – being agnostic to specific domains – can utilize a unified rule interface to discharge the regulation tasks.

We show that HORAE facilitates an intelligent service regulation pipeline by further exploiting a fine-tuned LLM coined RuleGPT to automatically convert regulation rules written in natural languages to the intermediate representation of HORAE. A formal semantics is further developed for HORAE to enable rule-consistency checking and quantitative viola-

* HORAE (/ˈhɔːriː/) refers to – in Greek mythology – the goddesses of order who guarded the gates of Olympus (Homer, *The Iliad*). This paper extends the work-in-progress article [Sun *et al.*, 2024].

tion recognition (via, e.g., constraint-solving techniques), cf. Fig. 1 (b), thereby yielding an *effective end-to-end framework for fully automated intelligent service regulation*.

Contributions. Our main contributions are as follows:

- We present HORAE as a unified specification language to model cross-domain regulation rules. We show that, with a well-designed semantics, HORAE facilitates core regulation functionalities such as consistency checking and quantitative violation recognition.
- We collect a benchmark dataset named SRR-Eval covering a wide range of regulation domains, and thence create a fine-tuned LLM called RuleGPT to automate the modeling process in HORAE. Both SRR-Eval and RuleGPT are open-sourced to support practical applications in regulation modeling.
- We show that HORAE and RuleGPT admit multimodal rules and enable an end-to-end intelligent service regulation framework. The latter is, to the best of our knowledge, the first framework that admits *fully automated* service regulation with effective *domain unification*.

Experimental results demonstrate the feasibility and effectiveness of RuleGPT in automating the modeling process in HORAE across different real-world regulation domains. In particular, RuleGPT with the size of 7B parameters suffices to outperform GPT-3.5 and perform on par with GPT-4o.

2 General Workflow

Fig. 2 sketches an overview of our end-to-end framework of HORAE-steered intelligent service regulation. This framework consists of the following three major steps:

(I) *Rule Dataset Construction*: This initial step aligns (pre-processed) multimodal regulation rules – leveraging existing multimodal models – to the text modality such that rules of different formats can later be interpreted through a *unified medium*, i.e., HORAE rules.

(II) *Rule Modeling and Checking*: The textual rule dataset is then translated into HORAE utilizing our fine-tuned RuleGPT. As per the formal semantics of HORAE, we can check the qualitative and quantitative *consistency* of the rule library to detect potential conflicts before deploying it to downstream regulation tasks.

(III) *Violation Recognition*: The downstream recognition tasks are discharged by multimodal models and algorithms, which assess the *violation probabilities* of basic events in the rule library. These violation probabilities contribute to an overall likelihood of rule violation (computed by a probability calculation engine).

Our preliminary implementation of the framework indicates that the above steps suffice to produce highly accurate outcomes in a fully automated manner in various real-world domains. This paper focuses on the design principles behind HORAE and RuleGPT in Step (II). The details of aligning multimodal rules to the text modality are provided in [Sun *et al.*, 2025, Appx. A] whilst the integration with downstream recognition models and algorithms is subject to future work.

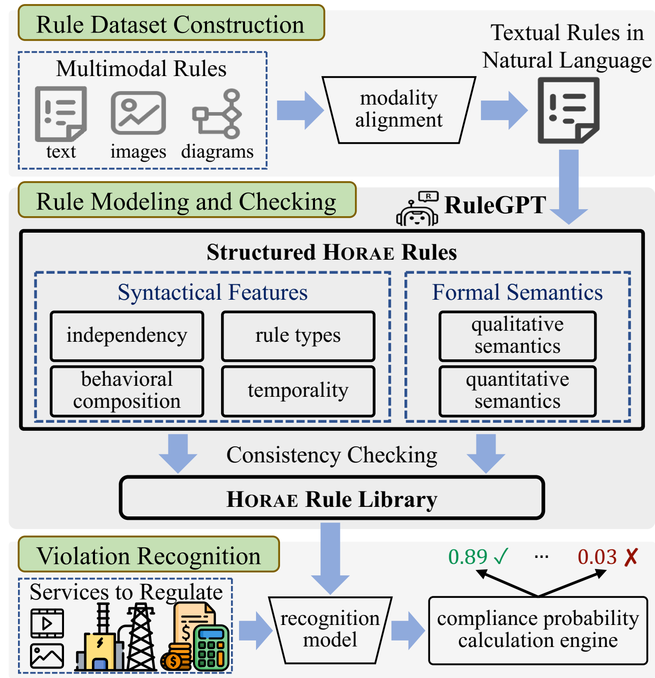


Figure 2: HORAE-steered intelligent service regulation.

3 Language Design

HORAE serves as the basis of intelligent service regulation by modeling a set of regulation rules in a structured, domain-agnostic fashion. We design the syntax and formal semantics as per several key principles, e.g., generality, structuration, automation, and quantification (as detailed in [Sun *et al.*, 2025, Appx. B]. These ingredients constitute the bases of HORAE parser (generated by ANTLR 4 [Parr *et al.*, 2014]); it compiles the text stream of a regulation rule into an abstract tree structure, thereby transforming flat, linear natural language into a structured language with hierarchical patterns.

3.1 Syntax

Our design of the HORAE syntax follows an *inductive reasoning paradigm*: We first collect a multilingual benchmark set of regulation rules across 50 domains (see details in [Sun *et al.*, 2025, Appx. D]), then conduct a syntactic analysis over this benchmark to extract key observations, and finally derive the core patterns and syntax from the body of observations.

Key observations extracted from our benchmark include

- *Independency*: Two textual sentences that are ostensibly disparate in grammatical structure (in terms of their host natural language) may well encode semantically similar regulation rules. For instance, consider the following three rules written in natural languages:

$$\underbrace{\text{Employees}}_{\text{subject}}\ \underbrace{\text{must}}_{\text{modal verb}}\ \underbrace{\text{wash hands}}_{\text{verb phrase}}\ \underbrace{\text{before returning to work}}_{\text{prepositional phrase}}.$$

$$\underbrace{\text{Hand washing before work resumption}}_{\text{subject}}\ \underbrace{\text{is}}_{\text{linking verb}}\ \underbrace{\text{mandatory}}_{\text{complement}}\ \underbrace{\text{for all employees}}_{\text{prepositional phrase}}.$$

在 返回 工作 前， 员工 必须 洗 手 。

$$(\underbrace{\text{when returning work before}}_{\text{temporal adverbial}},\ \underbrace{\text{employees}}_{\text{subject}}\ \underbrace{\text{must}}_{\text{modal verb}}\ \underbrace{\text{wash hands}}_{\text{verb phrase}}.)$$

These three rules (written in English, English, and Chinese, resp.) in fact represent analogous regulation intentions. Hence, the syntax of HORAE shall be independent of any specific natural language grammar and optimized towards the goal of admitting the most diverse set of intentions with as few grammatical categories as possible.

- *Rule Types*: A regulation rule is inherently well-typed, in the sense that, it typically describes certain behavior that is intended to be *enforced*, *recommended*, or *forbidden*:

  Employees must wear safety goggles at all times when on the factory floor. (enforced)

  It is advised that all participants review the safety manual before operating any machinery. (recommended)

  No smoking is allowed within 50 feet of the gas pumps. (forbidden)

  HORAE is thus expected to provide a simple mechanism to specify (a predefined set of) types for regulation rules.

- *Behavioral Composition*: The behavioral description of a regulation rule is highly *compositional*, namely, a regulated behavior often appears as a combination of several sub-behaviors via logical connectives, for instance,

  Company must conduct thorough testing and either obtain FDA approval or ensure compliance with international health regulations.

  $\Downarrow$ decomposition

  (Company conduct thorough testing) $\wedge$ ((Company obtain FDA approval) $\vee$ (Company ensure compliance with international health regulations)) .

  Such compositionality is crucial for service regulation as it facilitates the decomposition of a complex regulation problem into a set of sub-problems that can be more easily and accurately solved. HORAE support compositionality by maintaining an abstracted layer of *basic events*, which encode sub-behaviors of a regulated entity and can by logically assembled to describe the entire behavior.

- *Temporality*: Temporal properties are yet another important feature in service regulation; they are prominent especially for application domains where timing constraints are crucial, e.g., in financial services:

  Publicly traded companies have to disclose their quarterly financial results *within 45 days* by the end of the quarter; In case any significant financial events such as mergers or acquisitions occur within these 45 days, an additional prelim. report must be submitted *within 5 days* of the event.

  HORAE is consequently designed to support temporality by admitting *timestamped events* and *temporal constraints*, which further provide a natural means of modeling regulation rules that are (originally) specified in time-sensitive modalities, see [Sun *et al.*, 2025, Appx. A].

Based on these observations, we propose to model a *regulation rule* R in HORAE per the (abstracted snippet of) syntax:

$$\begin{array}{rcll}
R & ::= & \texttt{type}\ s & \text{(typed rule)} \\
\texttt{type} & ::= & \mathsf{shall} \mid \mathsf{should} \mid \mathsf{forbid} & \text{(predefined types)} \\
s & ::= & \neg s \mid s \wedge s \mid \langle \tau, e \rangle \mid e \mid \mathcal{C}(\boldsymbol{\tau}) & \text{(statement}^1\text{)} \\
\hline
e & ::= & \textit{object action} \mid & \text{(patterned event)} \\
 & & \textit{object action object} \mid \textit{object}.\textit{attribute} \diamond \textit{value} \mid & \\
 & & \textit{action object} \mid \textit{action}.\textit{attribute} \diamond \textit{value} \mid \cdots &
\end{array}$$

This abstract syntax consists of a *top-level grammar* and a *bottom-level grammar*, as indicated by the dashed line therein. The former combines (possibly timestamped) basic events via logical connectives into a regulation rule of certain type, whilst the latter assembles fine-grained sentence patterns and components into such basic events. Slicing basic events into smaller, detectable ingredients improves the precision of downstream recognition models and algorithms. Below, we provide details of the layered HORAE syntax.

Top-Level Grammar. This layer treats *basic events* as the smallest syntactic unit; they will later be interpreted as *propositions* in the formal semantics (see Sect. 3.2). The grammar allows for combining basic events e via logical connectives and specifying *types* (aka, *execution modes*) of the so-obtained regulation rule – shall, should, and forbid for *enforced*, *recommended*, and *forbidden* behaviors, respectively. For rules featuring temporal properties, the corresponding basic event can be associated with a *timestamp* τ signifying its time of occurrence; Moreover, *timing constraints* over timestamps $\boldsymbol{\tau} = \{\tau_1, \tau_2, \ldots\}$ are collected into $\mathcal{C}(\boldsymbol{\tau})$, which acts as a specific form of statement in the rule.

Bottom-Level Grammar. This layer describes core patterns of basic events extracted from our rule dataset. Key ingredients include (i) *action*: the behavior of the basic event; (ii) *object*: actor or recipient of the action – usually a detectable target; (iii) *attribute*: attributes of objects or actions (selected by the . operator), such as quantity, color, length, etc.; and (iv) *attribute* $\diamond$ *value*, with $\diamond \in \{<, >, \leq, \geq, =\}$: the *comparison* of some attribute against a given value (e.g., a threshold), which is commonly used in service regulation.

3.2 Formal Semantics

The formal semantics of HORAE aims to provide accounts of what a regulation rule adhering to the HORAE syntax *means* in an unambiguous manner. Such a semantics is essential to represent, interpret, and reason about a typically large set of regulation rules. In particular, it gives a mechanism to check the *consistency* of a rule library in order to detect potential conflicts before deploying it to downstream regulation tasks.

Qualitative Semantics

We start by formalizing the *qualitative semantics* of HORAE. Since rule types are fixed in HORAE, we interpret the (deno-

[1] $\vee$ and $\rightarrow$ are syntactic sugar expressible by $\neg$ and $\wedge$.

tational) semantics of a HORAE rule over its statement. Consider a library of type-free rules:

$$\mathit{RLib} \;=\; \{s_1, s_2, \ldots, s_n\} \;;$$

here, each rule statement s_k with $k = 1, \ldots, n$ is of the form:

$$s_k \;=\; \varphi_k(\boldsymbol{e}_k) \wedge \mathcal{C}_k(\boldsymbol{\tau}_k) \;,$$

where $\varphi_k(\boldsymbol{e}_k)$ is a *propositional* formula over the set of propositions, i.e., symbolic basic events $\boldsymbol{e}_k = \{e_{k1}, e_{k2}, \ldots\}$ in s_k; $\mathcal{C}_k(\boldsymbol{\tau}_k)$ is the corresponding quantifier-free linear constraints over timestamps[2] $\boldsymbol{\tau}_k = \{\tau_{k1}, \tau_{k2}, \ldots\}$. Without loss of generality, we assume that every rule statement s_k is in *conjunctive normal form* (CNF) over some quantifier-free arithmetic theory $\mathcal{T}$, i.e., a conjunction of disjunctions of (atomic) arithmetic predicates from $\mathcal{T}$, for example,

$$s_1 \;=\; (e_{11} \vee e_{12}) \wedge (\neg e_{13} \vee e_{14}) \wedge (\tau_{12} - \tau_{11} < \tau_{14}) \,. \quad (\star)$$

Let $\boldsymbol{e} \triangleq \bigcup_{k=1}^{n} \boldsymbol{e}_k$ and $\boldsymbol{\tau} \triangleq \bigcup_{k=1}^{n} \boldsymbol{\tau}_k$ be, respectively, the set of all basic events and timestamps in *RLib*. A *qualitative interpretation* of *RLib* is a (total) mapping:

$$I\colon\; \boldsymbol{e} \uplus \boldsymbol{\tau} \;\to\; \mathbb{B} \uplus \mathbb{R}_{\geq 0} \;,$$

where $\uplus$ denotes disjoint union; I thus interprets every basic event over the Boolean domain $\mathbb{B} \triangleq \{\mathsf{true}, \mathsf{false}\}$ and every timestamp over the set of non-negative real numbers $\mathbb{R}_{\geq 0}$. Let $\mathcal{I}$ be the set of all possible qualitative interpretations.

We define the *qualitative semantics* of *RLib* as

$$[\![\mathit{RLib}]\!]\colon\; \mathcal{I} \;\to\; \mathbb{B}, \qquad I \;\mapsto\; \bigwedge\nolimits_{k=1}^{n} s_k(I) \;,$$

where $s_k(I)$ denotes the substitution of interpretation I in s_k. The qualitative semantics of rule statement s_k, i.e., $[\![s_k]\!]$, is then a projection of $[\![\mathit{RLib}]\!]$ over $\boldsymbol{e}_k$ and $\boldsymbol{\tau}_k$. We say that the rule library *RLib* is *qualitatively consistent* if there exists an interpretation under which $[\![\mathit{RLib}]\!]$ evaluates to true, i.e.,

$$\exists I \in \mathcal{I}\colon\; [\![\mathit{RLib}]\!]\,(I) \;=\; \mathsf{true} \,. \qquad (\dagger)$$

The qualitative consistency of *RLib* as per (†) can be decided (over the quantifier-free mixed linear integer and real arithmetic [King *et al.*, 2014]) by various off-the-shelf satisfiability modulo theories (SMT) solvers, e.g., Z3 [de Moura and Bjørner, 2008] and CVC5 [Barbosa *et al.*, 2022].

Quantitative Semantics

The proposed qualitative semantics $[\![\mathit{RLib}]\!]$ does not address the *quantitative* aspects of rule satisfaction, i.e., the likelihood of it being satisfied. Such quantitative aspects are crucial for intelligent service regulation since the underlying recognition models and algorithms inherently produce imprecise results (measured by certain confidence factors). We thus extend the qualitative semantics to characterize quantitative satisfaction.

Let $\mathbb{P} \triangleq [0,1] \cap \mathbb{R}$ be the domain of probabilities. Given a rule library *RLib*, the *quantitative interpretation* of *RLib* is

$$I_\#\colon\; \boldsymbol{e} \uplus \boldsymbol{\tau} \;\to\; \mathbb{P} \uplus \mathbb{R}_{\geq 0} \;,$$

i.e., it interprets every basic event e_{ki} as the *probability* $p(e_{ki}) \in \mathbb{P}$ of it being true (cf. $\mathbb{B}$ for the qualitative case). Let $\mathcal{I}_\#$ be the set of all possible quantitative interpretations.

Similarly, we define the *quantitative semantics* of *RLib* as

$$[\![\mathit{RLib}]\!]_\#\colon\; \mathcal{I}_\# \;\to\; \mathbb{P}, \qquad I_\# \;\mapsto\; \prod\nolimits_{k=1}^{n} \mathit{Pr}(s_k(I_\#)) \;,$$

where $\mathit{Pr}(s_k(I_\#))$ denotes the probability that s_k is satisfied under $I_\#$, which can be computed recursively as

$$\mathit{Pr}(s_k(I_\#)) = \begin{cases} 1, & \text{if } s_k(I_\#) \text{ is logically equivalent to } \mathsf{true} \\ 0, & \text{if } s_k(I_\#) \text{ is logically equivalent to } \mathsf{false} \\ p(e_{ki}), & \text{if } s_k = e_{ki} \\ 1 - \mathit{Pr}(s(I_\#)), & \text{if } s_k = \neg s \\ \mathit{Pr}(s(I_\#)) \cdot \mathit{Pr}(s'(I_\#)), & \text{if } s_k = s \wedge s' \\ 1 - \mathit{Pr}(\neg s(I_\#)) \cdot \mathit{Pr}(\neg s'(I_\#)), & \text{if } s_k = s \vee s' \end{cases}$$

Analogously, the quantitative semantics of rule statement s_k, i.e., $[\![s_k]\!]_\#$, is then a projection of $[\![\mathit{RLib}]\!]_\#$ over $\boldsymbol{e}_k$ and $\boldsymbol{\tau}_k$. For instance, given the quantitative interpretation:

$$\begin{aligned} I_\#\colon\quad & e_{11} \mapsto 1, \quad e_{12} \mapsto 0, \quad e_{13} \mapsto 1/2, \quad e_{14} \mapsto 1/3, \\ & \tau_{11} \mapsto 3.5, \quad \tau_{12} \mapsto 6, \quad \tau_{13} \mapsto 11, \quad \tau_{14} \mapsto 3 \,, \end{aligned}$$

The quantitative semantics of the statement s_1 in ($\star$) is

$$[\![s_1]\!]_\#(I_\#) \;=\; (1 - 0 \cdot 1) \cdot (1 - 1/2 \cdot 2/3) \cdot 1 \;=\; 2/3 \,.$$

We say that the rule library *RLib* is *quantitatively consistent* if there exists a quantitative interpretation under which $[\![\mathit{RLib}]\!]_\#$ exhibits a positive satisfaction probability, i.e.,

$$\exists I_\# \in \mathcal{I}_\#\colon\; [\![\mathit{RLib}]\!]_\#(I_\#) \;>\; 0 \,. \qquad (\ddagger)$$

The quantitative consistency of *RLib* as per (‡) can be decided (over the non-linear real arithmetic [Tarski, 1951]) by dedicated SMT solvers, e.g., dReal [Gao *et al.*, 2013] and SMT-RAT [Corzilius *et al.*, 2015].

Remark. *Event correlation* remains as a challenge in consistency checking: Basic events $\boldsymbol{e}$ from the same rule library *RLib* may well be *semantically correlated* with each other, especially for events across different rule statements. We address this problem through *event abstraction*, i.e., abstracting these events written in natural languages into a set of symbolic propositions while preserving semantic correlations; see details in [Sun *et al.*, 2025, Appx. C]. ◁

4 Automation

This section presents the fine-tuning process of RuleGPT. As key to automation in intelligent service regulation, RuleGPT aims to automatically convert regulation rules written in natural languages to their unified, structured HORAE representations in the form of *token streams* as depicted in Fig. 3.

We note that off-the-shelf LLMs are not suitable for the above conversion task. The reasons are three-fold: (i) existing LLMs are *unaware* of HORAE since its knowledge is not part of the corpus used to pre-train these models; (ii) closed–source, proprietary models like GPT-4o are prone to security issues as many regulation tasks are *privacy-sensitive*; and (iii) general purpose LLMs, e.g., DeepSeek-R1 [Guo *et al.*, 2025] and GPT-4o, require *significant computational resources*. Moreover, they often exhibit low accuracies (see

[2] A timestamp τ_{ki} can be absent from $\boldsymbol{\tau}_k$ if e_{ki} is untimed. Assuming linearity of the constraints is necessary to attain decidability (for the qualitative setting) when discharging them via SMT solvers.

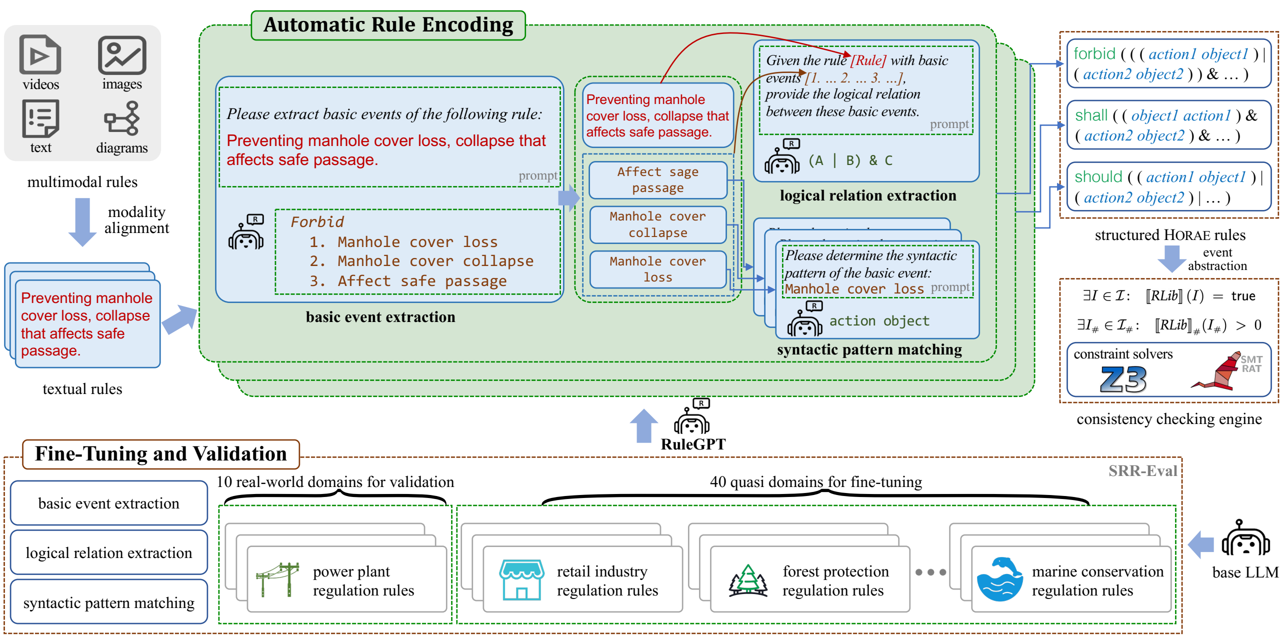


Figure 3: The overall process of automated transformation using the fined-tuned RuleGPT.

Sect. 5) when performing the transformation in a *monolithic* manner: Given a rule in natural language with a designed prompt, a general LLM cannot fully comprehend the basic events, logical relations, and syntactic patterns simultaneously and convert the rule into HORAE under zero- or few-shot conditions. To address these challenges, we propose to (i) *create* a benchmark dataset for service regulation rules (SRR-Eval, for short); (ii) *fine-tune* a pre-trained, open–sourced LLM using SRR-Eval to encode the HORAE knowledge; and (iii) *partition the fine-tuning process* into three cooperative phases, i.e., basic event extraction, logical relation extraction, and syntactic pattern matching.

Overview of SRR-Eval. SRR-Eval consists of 10 domains with real-world regulation rules (50 rules for each domain) and 40 domains with LLM-generated quasi rules (115 rules for each domain), amounting to *50 domains with 5,100 rules*. SRR-Eval is open-sourced at https://huggingface.co/datasets/Xfgll/SRR-Eval. See details in [Sun *et al.*, 2025, Appx. D].

Fine-Tuning Strategy. We use LoRA (*low-rank adaptation* [Hu *et al.*, 2022]) to fine-tune our base model M. Let $W \in \mathbb{R}^{d\times q}$ be the pre-trained weight matrix of M. In contrast to full fine-tuning where all model parameters are retrained by augmenting W with its accumulated gradient update $\Delta W \in \mathbb{R}^{d\times q}$, LoRA freezes M and injects low-rank decomposition matrices $A \in \mathbb{R}^{d\times r}$ and $B \in \mathbb{R}^{r\times q}$ with trainable parameters into each layer of the transformer architecture, i.e.,

$$W' \;=\; W + AB\,,$$

where $r \ll \min(d, q)$ is the rank of a LoRA module; $W' \in \mathbb{R}^{d\times q}$ is the adapted weight matrix. LoRA thus significantly reduces the number of trainable parameters. We denote by

$$M' \;=\; \text{LoRA}\,(M, D)$$

the process of fine-tuning M via LoRA into an adapted model M' which incorporates the knowledge encoded in dataset D.

4.1 Extracting Basic Events

Given a textual regulation rule R written in a natural language, the phase of *basic event extraction* aims to fine-tune a pre-trained base LLM M into a dedicated model M_{event} for extracting the set E of basic events from R, i.e.,

$$M_{\mathit{event}}\colon \quad R \;\mapsto\; \{e_1, e_2, \ldots, e_m\} \;\triangleq\; E\,,$$

where every basic event e_i is of a certain pattern adhering to the HORAE syntax. See [Sun *et al.*, 2025, Appx. D.1] for examples of the extraction. Note that recognizing the specific event patterns is the task of the syntactic pattern matching phase as discussed in Sect. 4.3.

We obtain M_{event} by fine-tuning M via LoRA, namely,

$$M_{\mathit{event}} \;=\; \text{LoRA}\,(M, D_{\mathit{event}})\;,$$

i.e., we feed LoRA with a dedicated training dataset D_{event} sourced from SRR-Eval, which is formatted as

$$D_{\mathit{event}} \;=\; \{(u_i, a_i)\}_{i=1}^{n}$$

with u_i being the *user prompt* and a_i the corresponding *assistant's extraction*. Specifically, every entry (u_i, a_i) in D_{event} is of the following query-response format:

$$u_i \;=\; \text{“Please extract basic events of the following rule: [original rule]”}$$

$$a_i \;=\; \text{“[basic events]”}$$

where [original rule] and [basic events] are raw ingredients of the *composite quasi rules* in SRR-Eval.

4.2 Extracting the Logical Relation

In the phase of *logical relation extraction*, we fine-tune a base LLM M into a tailored model $M_{\mathit{logic}}\colon (R, E) \;\mapsto\; L$ for extracting the logical relation L between basic events E of

R; e.g., the logical relation of rule (R3) in [Sun *et al.*, 2025, Appx. D.1] is $L = e_{11} \vee e_{12} \vee e_{13}$. Note that the quality of the HORAE transformation depends heavily on M_{logic} because logical relations are the key contributor in both the qualitative and quantitative semantics of HORAE as shown in Sect. 3.2.

Akin to the event extraction phase, M_{logic} is derived by $M_{logic} = \text{LoRA}(M, D_{logic})$. Here, the training dataset $D_{logic} = \{(u'_i, a'_i)\}_{i=1}^{n}$ consists of query-response pairs:

$$u'_i = \text{“Given the rule [original rule] with basic events [basic events], provide the logical relation between these basic events”}$$

$$a'_i = \text{“[logical relation]”}$$

where [original rule], [basic events], and [logical relation] are raw data of *composite quasi rules* in SRR-Eval.

4.3 Matching Syntactic Patterns

Let $T = \{t_1, t_2, \ldots, t_j\}$ be the fixed *finite* set of syntactic patterns as defined in the bottom-level grammar of HORAE in Sect. 3.1. The goal of *syntactic pattern matching* is to attach to every basic event in E a corresponding syntactic pattern in T via a fine-tuned model $M_{syntax}\colon E \to T$.

In analogous to M_{event} and M_{logic}, M_{syntax} is obtained by $M_{syntax} = \text{LoRA}(M, D_{syntax})$, where the dedicated training dataset $D_{syntax} = \{(u''_i, a''_i)\}_{i=1}^{n}$ is composed of

$$u''_i = \text{“Please determine the syntactic pattern of the basic event: [basic event]”}$$

$$a''_i = \text{“[syntactic pattern]”}$$

where [basic event] and [syntactic pattern] are raw ingredients of the *single-event quasi rules* in SRR-Eval ([Sun *et al.*, 2025, Appx. D.1]). These ingredients are utilized to train RuleGPT to classify basic events into right categories.

By combining the aforementioned fine-tuned models, we obtain RuleGPT (see the general pipeline in Fig. 3):

$$\text{RuleGPT} = \{M_{event}, M_{logic}, M_{syntax}\} .$$

5 Experimental Results

This section presents an empirical evaluation of RuleGPT's performance against several baselines. Our primary goal is to *demonstrate the feasibility and effectiveness of* RuleGPT *in automating the modeling process in* HORAE *across different real-world regulation domains*, which essentially enables our end-to-end framework for fully automated intelligent service regulation. RuleGPT is open-sourced via GitHub at https://github.com/FICTION-ZJU/RuleGPT.

Settings of Fine-Tuning. We implement RuleGPT by adapting – via the LoRA technique [Hu *et al.*, 2022] – Qwen2.5-7B-Ins [Yang *et al.*, 2024] as our common base model shared by the three fine-tuning phases. The fine-tuning procedure is conducted on a single NVIDIA A100-40GB GPU. We set the learning rate to 1×10^{-4} and employ gradient accumulation with 16 steps to effectively manage the computational load. The training spans 3 epochs, we use bf16 precision to assist in managing GPU memory efficiently and employ gradient checkpointing to further optimize the memory usage. The

Real-world dataset in SRR-Eval	Qwen2.5-7B-Ins			GPT-3.5			RuleGPT			GPT-4o		
	$\mathcal{P}$	$\mathcal{R}$	$\mathcal{F}_1$	$\mathcal{P}$	$\mathcal{R}$	$\mathcal{F}_1$	$\mathcal{P}$	$\mathcal{R}$	$\mathcal{F}_1$	$\mathcal{P}$	$\mathcal{R}$	$\mathcal{F}_1$
power plant	0.40	0.58	0.48	0.50	0.63	0.56	0.62	0.69	0.66	**0.71**	**0.78**	**0.74**
public place safety	0.40	0.65	0.50	0.72	0.80	0.76	**0.77**	0.76	0.76	0.76	**0.82**	**0.78**
tourism	0.34	0.62	0.44	0.71	**0.78**	0.75	**0.82**	0.76	**0.79**	0.69	**0.78**	0.73
energy regulation	0.62	0.59	0.60	0.73	0.55	0.63	**0.78**	0.51	0.62	0.76	**0.63**	**0.69**
urban management	0.53	0.74	0.62	0.64	0.77	0.70	**0.73**	0.79	**0.76**	0.63	**0.80**	0.70
forest products	0.35	0.48	0.40	**0.63**	0.47	0.54	0.57	0.52	0.54	0.55	**0.60**	**0.57**
tabacco	0.33	0.56	0.41	**0.72**	0.68	**0.70**	0.58	0.66	0.61	0.57	**0.75**	0.65
agricultural markets	0.34	0.50	0.40	0.59	0.43	0.50	**0.60**	0.54	0.57	0.58	**0.57**	**0.58**
food safety	0.33	0.54	0.41	0.53	0.54	0.54	**0.57**	**0.57**	**0.57**	0.51	0.56	0.54
forest degradation	0.36	0.52	0.42	**0.62**	0.46	0.53	0.43	0.45	0.44	0.59	**0.58**	**0.59**

Table 1: Experimental results w.r.t. basic event extraction ($\mathcal{P}$ for precision, $\mathcal{R}$ for recall, and $\mathcal{F}_1$ for F_1-score).

fine-tuning datasets are sourced from SRR-Eval as described in [Sun *et al.*, 2025, Appx. D]; a set of hyperparameters, e.g., weight decay (0.1), Adam optimizer's β_2 (0.95), warmup ratio (0.01), and cosine learning rate scheduler (enable) further contributes to the training stability and efficiency.

Baselines. We compare RuleGPT against three baselines: Qwen2.5-7B-Ins, GPT-3.5(-Turbo), and GPT-4o(-latest). The latter two, though being closed-source models, are chosen because (i) they are widely recognized for their capabilities in natural language understanding and generation; and (ii) models with 7B parameters may outperform GPT-3.5 in certain scenarios, as observed in [Bai *et al.*, 2023, Sect. 3.3].

In the rest of this section, we present detailed experimental results with respect to the three fine-tuning phases.

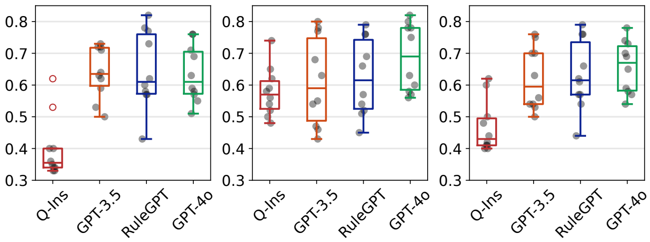
5.1 Basic Event Extraction

For the component of basic event extraction, we compare RuleGPT against the baselines in terms of three performance metrics: the *precision* $\mathcal{P}$, the *recall* $\mathcal{R}$, and the F_1*-score* $\mathcal{F}_1$ (i.e., the harmonic mean of $\mathcal{P}$ and $\mathcal{R}$). These metrics together provide a comprehensive assessment of the models' accuracy and adaptability in extracting basic events. The details of these metrics are presented in [Sun *et al.*, 2025, Appx. E].

We report our experimental results w.r.t. basic event extraction in Table 1, where we mark the **best** results and the second-best results among all the competitors. The scattered boxplots in Fig. 4 further visualize these numerical results separately for the three metrics. The following observations are drawn from these results: (i) RuleGPT significantly outperforms its base model Qwen2.5-7B-Ins in all three metrics, thus demonstrating the feasibility and effectiveness of our fine-tuning process and the quality of SRR-Eval. (ii) For the precision metric, RuleGPT is the winner amongst all the models – it achieves the best results over 6/10 benchmarks. (iii) For the recall metric, RuleGPT exhibits a comparable ability with GPT-3.5, but they both are slightly inferior to GPT-4o. (iv) For the F_1-score metric, RuleGPT *performs better than* GPT-3.5, *slightly inferior to* GPT-4o.

5.2 Logical Relation Extraction

Next, we compare RuleGPT against the baselines in terms of the *accuracy* in extracting logical relations between basic events. Since the formal semantics of a HORAE rule depends heavily on the underlying logical relation (see Sect. 3.2), an extraction is considered *correct* iff the extracted logical relation *semantically coincides with* the relation in SRR-Eval.



(a) The precision $\mathcal{P}$ (b) The recall $\mathcal{R}$ (c) The F_1-score $\mathcal{F}_1$

Figure 4: Visualization of data in Table 1 (Q-Ins abbreviates Qwen2.5-7B-Ins). Every scattered boxplot depicts the corresponding column of Table 1 with its five-number summary.

Real-world dataset in SRR-Eval	Logical relation extraction				Syntactic pattern matching			
	Q-Ins	GPT-3.5	GPT-4o	RuleGPT	Q-Ins	GPT-3.5	GPT-4o	RuleGPT
power plant	0.34	0.38	**0.70**	0.66	0.22	0.62	0.66	**0.72**
public place safety	0.39	0.57	0.78	**0.84**	0.08	0.13	**0.36**	0.23
tourism	0.24	0.40	0.74	**0.76**	0.14	0.17	0.16	**0.24**
energy regulation	0.11	0.24	0.4	**0.73**	0.06	0.23	**0.65**	0.39
urban management	0.22	0.38	**0.80**	0.60	0.11	0.17	**0.40**	0.26
forest products	0.10	0.34	**0.66**	0.46	0.08	0.19	**0.43**	0.39
tabacco	0.14	0.36	0.58	**0.66**	0.18	0.17	0.29	**0.47**
agricultural markets	0.10	0.34	**0.60**	0.52	0.36	0.08	0.24	**0.44**
food safety	0.18	0.48	0.62	**0.64**	0.31	0.15	**0.36**	0.27
forest degradation	0.10	0.16	**0.52**	0.44	0.23	0.17	0.26	**0.27**
Mean	0.19	0.37	0.64	0.63	0.18	0.21	0.38	0.37
Variance	0.01	0.11	0.01	0.01	0.01	0.14	0.02	0.02

Table 2: Accuracy of logical relation extraction and syntactic pattern matching (Q-Ins is shorthand for Qwen2.5-7B-Ins).

The evaluation results w.r.t. logical relation extraction are reported in (the left part of) Table 2. It shows that RuleGPT *exhibits the highest accuracy on par with* GPT-4o *consistently over all the ten benchmarks*. More concretely, we make the following observations: (i) As the underlying base model of RuleGPT, Qwen2.5-7B-Ins performs poorly in identifying logical relations. (ii) However, our fine-tuning procedure suffices to optimize this small model to perform better than the GPT-3.5, yielding a cost-effective and computationally efficient solution. (iii) The comparable performance of RuleGPT against GPT-4o indicates that, in our case, a small model fine–tuned with SRR-Eval can potentially replace larger proprietary models that are generally more resource-intensive.

5.3 Syntactic Pattern Matching

Finally, we compare RuleGPT against the baselines in terms of the *accuracy* in matching syntactic patterns of basic events. As it is essentially a classification task, the result is considered *correct* iff *the correct syntactic category is identified*.

The experimental results w.r.t. syntactic pattern matching are reported in (the right part of) Table 2. We observe that RuleGPT *achieves the highest accuracy over 5/10 benchmarks, which significantly outperforms* Qwen2.5-7B-Ins *and the proprietary model* GPT-3.5, and is on par with GPT-4o.

Overall Performance. Our experiments demonstrate the overall feasibility and effectiveness of RuleGPT in automating the modeling process in HORAE across different real-world regulation domains: (i) RuleGPT significantly outperforms GPT-3.5 in extracting logical relations and syntactic patterns, and performs on par with it in the task of basic event extraction. (ii) The substantial improvement of RuleGPT over Qwen2.5-7B-Ins underscores the effectiveness of our fine-tuning strategy, further demonstrating the high quality of SRR-Eval we have created. (iii) We show the feasibility of automating a complex task (i.e., HORAE modeling) by breaking it down into simpler components (i.e., the three fine-tuned models), each of which is optimized individually and contributes to a highly effective overall system (i.e., RuleGPT).

6 Related Work

Service regulation strives to represent regulatory compliance requirements with modeling languages for automation [zur Muehlen and Indulska, 2010]: The language SWRL [Horrocks *et al.*, 2004] enables complex reasoning in semantic web applications. BPMN-Q [Awad *et al.*, 2011] visually specifies compliance rules and explains violations in business processes using a pattern-based approach to link BPMN-Q graphs with formal temporal logic expressions. CRL [Elgammal *et al.*, 2016] offers a comprehensive framework for managing business process compliance, which introduces abstract pattern-based specifications while supporting compensations and non-monotonic requirements. DecSerFlow [van der Aalst and Pesic, 2006] is a declarative language for specifying, enacting, and monitoring service flows, grounded in temporal logic to address the autonomous nature of services. An orthogonal line of research aims to evaluate the expressiveness and complexity of rule languages by leveraging real-world examples and normative classification frameworks, addressing the challenge of representing complex constraints across multiple process perspectives [Zasada *et al.*, 2023].

Our work is closely related to the rule language CDSRL and the LLM-based converter RegGPT recently proposed in [Wang *et al.*, 2024] to model cross-domain regulatory requirements. The key differences are (i) HORAE supports *behavioral compositionality* by maintaining an abstracted layer of fine-grained basic events, thus admitting domain-agnostic downstream recognition models to discharge the regulation tasks. In contrast, CDSRL emphasizes holistic rule structuring without explicit behavioral decomposition; (ii) HORAE admits *formal semantics* that enable automated consistency checking and violation quantification through SMT solvers, whereas CDSRL lacks executable validation mechanisms beyond syntactic template matching; (iii) RuleGPT supports *fully autonomous rule conversion* through phased fine-tuning of open-sourced models while RegGPT's conversion pipeline depends critically on GPT-4 and prompt templates.

7 Conclusion

We presented the domain-agnostic modeling language HORAE. It enables an end-to-end intelligent regulation framework leveraging a fine-tuned LLM RuleGPT to automate the conversion of natural language regulation rules into a structured intermediate representation. HORAE is, to the best of our knowledge, the first modeling language that admits *fully automated* service regulation with effective domain-modality unification. Future work includes integrating HORAE and RuleGPT with downstream recognition models and algorithms to detect (quantitative) service-rule violations.

Acknowledgments

This work was partially supported by the National Key R&D Program of China (No. 2022YFF0902600), by the ZJNSF Major Program (No. LD24F020013 and LD24F020014), by the Fundamental Research Funds for the Central Universities of China (No. 226-2024-00140), by the Zhejiang Pioneer Project (No. 2023C01G1752957), and by the ZJU Education Foundation's Qizhen Talent program. The authors would like to thank Linyu Yang for the helpful discussion on the formal semantics of HORAE.

References

[Awad *et al.*, 2011] Ahmed Awad, Matthias Weidlich, and Mathias Weske. Visually specifying compliance rules and explaining their violations for business processes. *J. Vis. Lang. Comput.*, 22(1):30–55, 2011.

[Bai *et al.*, 2023] Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, et al. Qwen technical report. *CoRR*, abs/2309.16609, 2023.

[Barbosa *et al.*, 2022] Haniel Barbosa, Clark W. Barrett, Martin Brain, Gereon Kremer, Hanna Lachnitt, Makai Mann, et al. cvc5: A versatile and industrial-strength SMT solver. In *TACAS (I)*, volume 13243 of *LNCS*, pages 415–442. Springer, 2022.

[Corzilius *et al.*, 2015] Florian Corzilius, Gereon Kremer, Sebastian Junges, Stefan Schupp, and Erika Ábrahám. SMT-RAT: An open source C++ toolbox for strategic and parallel SMT solving. In *SAT*, volume 9340 of *LNCS*, pages 360–368. Springer, 2015.

[de Moura and Bjørner, 2008] Leonardo Mendonça de Moura and Nikolaj S. Bjørner. Z3: An efficient SMT solver. In *TACAS*, volume 4963 of *LNCS*, pages 337–340. Springer, 2008.

[Elgammal *et al.*, 2016] Amal Elgammal, Oktay Türetken, Willem-Jan van den Heuvel, and Mike P. Papazoglou. Formalizing and appling compliance patterns for business process compliance. *Softw. Syst. Model.*, 15(1):119–146, 2016.

[Gao *et al.*, 2013] Sicun Gao, Soonho Kong, and Edmund M. Clarke. dReal: An SMT solver for nonlinear theories over the reals. In *CADE*, volume 7898 of *LNCS*, pages 208–214. Springer, 2013.

[Guo *et al.*, 2025] Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, et al. DeepSeek-R1: Incentivizing reasoning capability in LLMs via reinforcement learning. *CoRR*, abs/2501.12948, 2025.

[Horrocks *et al.*, 2004] Ian Horrocks, Peter F. Patel-Schneider, Harold Boley, Said Tabet, Benjamin Grosof, and Mike Dean. SWRL: A semantic web rule language combining OWL and RuleML. *W3C Member submission*, 21(79):1–31, 2004.

[Hu *et al.*, 2022] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, et al. LoRA: Low-rank adaptation of large language models. In *ICLR*, 2022.

[Kaginalkar *et al.*, 2021] Akshara Kaginalkar, Shamita Kumar, Prashant Gargava, and Dev Niyogi. Review of urban computing in air quality management as smart city service. *Urban Climate*, 39:100972, 2021.

[King *et al.*, 2014] Tim King, Clark W. Barrett, and Cesare Tinelli. Leveraging linear and mixed integer programming for SMT. In *FMCAD*, pages 139–146. IEEE, 2014.

[Luccioni *et al.*, 2023] Alexandra Sasha Luccioni, Sylvain Viguier, and Anne-Laure Ligozat. Estimating the carbon footprint of BLOOM, a 176B parameter language model. *J. Mach. Learn. Res.*, 24:253:1–253:15, 2023.

[Parr *et al.*, 2014] Terence Parr, Sam Harwell, and Kathleen Fisher. Adaptive *LL(*)* parsing: The power of dynamic analysis. In *OOPSLA*, pages 579–598. ACM, 2014.

[Raji *et al.*, 2024] Mustafa Ayobami Raji, Hameedat Bukola Olodo, Timothy Tolulope Oke, Wilhelmina Afua Addy, Onyeka Chrisanctus Ofodile, and Adedoyin Tolulope Oyewole. E-commerce and consumer behavior: A review of AI-powered personalization and market trends. *GSC Advanced Research and Reviews*, 18(3):066–077, 2024.

[Sun *et al.*, 2024] Yutao Sun, Mingshuai Chen, Kangjia Zhao, and Jintao Chen. HORAE: A domain-agnostic modeling language for automating multimodal service regulation. In *ICWS*, pages 244–246. IEEE, 2024.

[Sun *et al.*, 2025] Yutao Sun, Mingshuai Chen, Tiancheng Zhao, Kangjia Zhao, He Li, Jintao Chen, et al. HORAE: A domain-agnostic modeling language for automating multimodal service regulation. *CoRR*, abs/2406.06600, 2025.

[Tarski, 1951] Alfred Tarski. *A decision method for elementary algebra and geometry*. University of California Press, Berkeley, 1951.

[van der Aalst and Pesic, 2006] Wil M. P. van der Aalst and Maja Pesic. Decserflow: Towards a truly declarative service flow language. In *WS-FM*, volume 4184 of *LNCS*, pages 1–23. Springer, 2006.

[Wang *et al.*, 2024] Zhaowen Wang, Qi Xie, Huan Zhang, Weihuan Min, Li Kuang, and Lingyan Zhang. RegGPT: A tool for cross-domain service regulation language conversion. In *ICWS*, pages 416–425. IEEE, 2024.

[Wei *et al.*, 2022] Jason Wei, Maarten Bosma, Vincent Y. Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, et al. Finetuned language models are zero-shot learners. In *ICLR*, 2022.

[Yang *et al.*, 2024] An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, et al. Qwen2.5 technical report. *CoRR*, abs/2412.15115, 2024.

[Zasada *et al.*, 2023] Andrea Zasada, Mustafa Hashmi, Michael Fellmann, and David Knuplesch. Evaluation of compliance rule languages for modelling regulatory compliance requirements. *Software*, 2(1):71–120, 2023.

[zur Muehlen and Indulska, 2010] Michael zur Muehlen and Marta Indulska. Modeling languages for business processes and business rules: A representational analysis. *Inf. Syst.*, 35(4):379–390, 2010.