

SAVANT: A Neuro-Symbolic Verification Framework for Adjuvant Design

Yi Chen^{1,2,3}, Yu Zhang^{4,5}, Jian Xu^{1,2}, Hua Yue^{4,5}, Xinming Wang^{1,2,3}
Zequan Lyu^{4,6}, Boran Wang³, Hongyi Liu⁴, Yan Wang⁴
Peiyuan Cao⁴, Xu-Yao Zhang^{1,2,3}, Wei Wei^{4,5}✉, Cheng-Lin Liu^{1,2,3}✉

¹ State Key Laboratory of Multimodal Artificial Intelligence Systems (MAIS),
Institute of Automation, Chinese Academy of Sciences (CAS), Beijing, China

² School of Artificial Intelligence, University of Chinese Academy of Sciences (UCAS)

³ Zhongguancun Academy, Beijing, China

⁴ State Key Laboratory of Biopharmaceutical Preparation and Delivery,
Institute of Process Engineering, CAS, Beijing, China

⁵ School of Chemical Engineering, UCAS, Beijing, China

⁶ School of Advanced Interdisciplinary Sciences, UCAS, Beijing, China

yi.chen@nlpr.ia.ac.cn

Abstract

Large language models (LLMs) can propose vaccine adjuvant designs with plausible mechanistic rationales, but such outputs often lack grounding in established immunological knowledge. We present SAVANT, a neuro-symbolic framework for LLM-based adjuvant design that combines two-stage constrained generation with literature-grounded verification. SAVANT first generates a natural-language design rationale, then renders it into an ontology-constrained domain-specific language (DSL) using logic masks and finite-state control. To reason under fragmented evidence and incomplete first-principles rules, SAVANT verifies generated designs against literature-derived symbolic evidence through minimal sub-theorem proof. In expert-aligned evaluation, SAVANT shows greater stability in stage-level rank alignment than the strongest LLM-judge baselines, while achieving the strongest alignment for immune-effect support and mechanistic reasoning. SAVANT provides graded, interpretable assessments that distinguish supported mechanisms, under-supported hypotheses, and knowledge gaps for hypothesis-driven experimentation. Data and code will be released at <https://github.com/banjiuyufen/SAVANT>.

1 Introduction

Vaccines are among the most effective tools for preventing infectious disease, and adjuvants are critical for shaping the strength, duration, and type of vaccine-induced immunity (Coffman et al., 2010; Reed et al., 2013; Lavelle and Mcentee, 2024; Zhao et al., 2023). Designing effective adjuvants, however, remains challenging. A formulation must account for material properties, de-

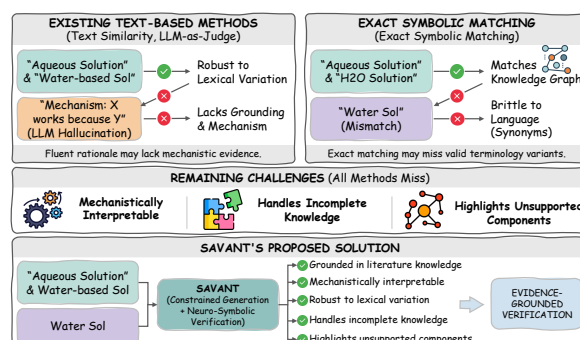


Figure 1: Motivation for neuro-symbolic verification in adjuvant design.

livery context, innate immune activation, antigen presentation, and downstream adaptive responses (O'Hagan and Fox, 2015; Pulendran et al., 2021; Turley and Lavelle, 2022). Although many adjuvant mechanisms have been studied, their effects remain context-dependent, and no complete first-principles theory can yet reliably predict how a formulation will shape immune outcomes. Evidence is also spread across a large and diverse immunology literature, making it difficult to judge whether a proposed design is grounded in prior knowledge.

Recent large language models (LLMs) have shown powerful potential for scientific design and discovery, including chemistry, molecular design, materials optimization, and scientific information extraction (Boiko et al., 2023; M. Bran et al., 2024; Reinhart and Statt, 2024; Dagdelen et al., 2024). Yet fluent scientific rationales are not necessarily evidence-grounded. LLMs may produce explanations that sound plausible but are unverified or incorrect, and hallucination remains a major concern in knowledge-intensive scientific settings (Farquhar et al., 2024; Manakul et al., 2023). This issue is especially important for adjuvant design,

where small changes in formulation, carrier material, particle size, or immune agonist can change both the mechanism and the expected immune outcome. We therefore need a framework that can use the flexibility of LLMs while grounding generated designs in structured immunological evidence.

As shown in Figure 1, existing verification strategies face a trade-off between flexibility and grounding. Text-based methods, including embedding similarity, retrieval-based citation support, and LLM-as-Judge evaluation, are robust to lexical variation and can compare generated rationales with retrieved evidence (Gao et al., 2023; Zheng et al., 2023a). However, they do not explicitly check whether each construction choice, immune effect, or mechanistic step is supported by immunological evidence.

Exact symbolic matching has the opposite limitation. It provides explicit structure through rules, logical forms, or knowledge bases, but is brittle to synonymy, naming variation, numerical variation, and missing rules. This brittleness is problematic in adjuvant design because the evidence base is incomplete: neither the literature nor current immunological theory provides a closed rule set from which every valid design can be derived. Prior work on fact verification and scientific claim verification has shown the value of evidence-grounded assessment (Thorne et al., 2018; Wadden et al., 2020), and neuro-symbolic reasoning has shown how neural representations can be combined with symbolic or differentiable proof procedures (Rocktäschel and Riedel, 2017; Manhaeve et al., 2018; Yang et al., 2017). However, these methods do not directly address open-ended adjuvant design, where the system must reason over structured formulations, immune effects, and mechanisms under incomplete evidence.

We introduce SAVANT, a neuro-symbolic framework for LLM-based adjuvant design. It combines two-stage LLM generation with literature-grounded verification. The LLM first generates a free-form design rationale from the user goal, and then renders it into an ontology-constrained domain-specific language (DSL) under logic masks and finite-state control. This preserves the exploratory capacity of LLMs while producing structured outputs that can be checked symbolically.

SAVANT grounds the DSL into unified predicates and verifies construction claims, immune-effect claims, and mechanistic steps against sym-

bolic evidence extracted from the literature. This setting is challenging because adjuvant first-principle mechanisms remain only partially understood, the literature is fragmented, and theorem libraries are necessarily incomplete. SAVANT therefore does not require a single complete proof chain for every mechanism. Instead, it decomposes mechanisms into minimal sub-theorems and aggregates local evidence support across construction, effect, and causal steps.

SAVANT is not intended to replace empirical vaccine evaluation. Its goal is to generate structurally valid proposals, estimate their support under the current evidence base, and make unsupported assumptions explicit before experimental follow-up. Our main contributions are as follows:

- To our knowledge, we present the first neuro-symbolic verifier for LLM-based adjuvant design, formulating proposal assessment as literature-grounded verification over structured immunological predicates.
- We develop SAVANT, a framework that decomposes generated adjuvant proposals into construction, immune-effect, and mechanism-level proof obligations, and verifies them against a theorem library through structure-aware matching and evidence aggregation.
- We enable interpretable, evidence-grounded assessment of open-ended scientific designs by reporting proof-level support traces and localized knowledge gaps, providing a structured alternative to global text-similarity scoring and holistic LLM-as-Judge evaluation.

2 Related Work

2.1 Scientific Design with LLMs

LLMs have increasingly been used for scientific design, including synthesis planning, tool-augmented molecular reasoning, MOF prediction and generation, and protein sequence design (Boiko et al., 2023; M. Bran et al., 2024; Zheng et al., 2023b; Kang and Kim, 2024; Zeinalipour et al., 2024). Agentic systems further connect LLMs with search, code execution, domain tools, and laboratory automation, making them useful assistants for complex scientific workflows (Boiko et al., 2023; M. Bran et al., 2024).

AI-driven vaccine adjuvant design remains comparatively limited. Existing work mainly focuses on screening, prediction, or recommenda-

tion over known candidates, such as machine-learning-based adjuvanticity and toxicity assessment in ADB (Natsume-Kitatani et al., 2025) and ontology-grounded disease–adjuvant recommendation in VaxjoGNN (Zheng and He, 2025). More recently, Chen et al. (2026) introduced an open-ended benchmark and formal framework for adjuvant research with multimodal LLMs. These studies provide useful resources for adjuvant modeling, but they do not directly address evidence-grounded verification of open-ended LLM-generated adjuvant designs.

2.2 Evidence-Grounded Neuro-Symbolic Verification

Evidence-grounded verification evaluates whether a claim is supported by retrieved evidence. FEVER formulates this task as deciding whether a textual claim is supported, refuted, or lacks sufficient evidence (Thorne et al., 2018), while SciFact extends the setting to scientific claims by retrieving evidence-containing abstracts and rationales from the literature (Wadden et al., 2020). These benchmarks establish useful verification paradigms, but they mainly operate on textual claims rather than structured design objects with construction constraints, immune effects, and mechanistic chains.

Neuro-symbolic reasoning provides a complementary route by combining neural similarity with symbolic rules or proof procedures. Neural Theorem Provers soften symbolic unification with differentiable matching (Rocktäschel and Riedel, 2017), Neural Logic Programming learns differentiable first-order rules for knowledge-base reasoning (Yang et al., 2017), and DeepProbLog integrates neural predicates into probabilistic logic programs (Manhaeve et al., 2018). These methods improve the flexibility of symbolic reasoning, but open-ended adjuvant design poses a different challenge: its mechanisms are supported by fragmented experimental evidence, and complete proof chains are rarely available.

3 Method

3.1 Formalizing Adjuvant Design

SAVANT first turns an adjuvant proposal into a formal object whose components correspond to distinct biological claims. Following the adjuvant formal framework (Chen et al., 2026), we decompose each generated proposal into

$$y = (\mathcal{C}, \mathcal{E}, \mathcal{M}), \quad (1)$$

where $\mathcal{C}$ contains construction claims about the formulation, $\mathcal{E}$ contains claimed immune-effect predicates, and $\mathcal{M}$ contains mechanistic rationales. This decomposition separates the composition of the design, the immune outcome it aims to induce, and the biological process used to justify that outcome.

To make these claims verifiable, SAVANT represents each elementary biological statement with a relation–role–value schema:

$$p = f(r_1 = v_1, r_2 = v_2, \dots, r_n = v_n), \quad (2)$$

where f denotes the predicate functor, i.e., a design or biological relation, r_i specifies the semantic role of an argument, and v_i is the corresponding entity, property, or value. For example, f may denote relations such as DELIVER, ACTIVATE, or ENHANCE, with roles such as *agent*, *target*, or *outcome*. This representation preserves how each term is used in the biological statement, such as a material component, immune target, measured quantity, or experimental condition.

A mechanistic rationale is represented as an ordered predicate chain using the DSL transition operator:

$$m = p_0 \gg p_1 \gg \dots \gg p_T. \quad (3)$$

Each $\gg$ denotes a literature-checkable transition between two biological states or effects. A proposal may contain multiple such chains:

$$\mathcal{M} = \{m^{(1)}, m^{(2)}, \dots\}. \quad (4)$$

This chain form allows a rationale to be evaluated both as a whole and as a set of local mechanistic steps.

Literature evidence is formalized in the same predicate space as a theorem library of evidence-grounded Horn clauses:

$$\mathcal{K} = \{b_j \Rightarrow p_j\}_{j=1}^N. \quad (5)$$

Each element $r_j \in \mathcal{K}$ denotes a single-premise rule with premise predicate b_j and conclusion predicate p_j , both of the form defined above. Here $\Rightarrow$ denotes a literature-supported causal or evidential relation, rather than a strict logical implication. By mapping generated proposals and literature-derived evidence into the same symbolic language, SAVANT turns adjuvant verification into a comparison between structured biological claims and structured evidence, rather than between free-form rationales and text passages. Appendix I provides end-to-end worked examples illustrating ontology-constrained DSL rendering

and the resulting evidence-grounded verification traces.

3.2 Overview

As shown in Figure 2, SAVANT combines two-stage LLM generation with neuro-symbolic verification. The LLM first generates a free-form design rationale and then renders it into an ontology-constrained DSL. The verifier grounds the DSL in a shared predicate space and checks construction claims, immune-effect claims, and mechanistic chains against literature-derived symbolic evidence using symmetric instantiation, minimal sub-theorem proof, and evidence aggregation.

3.3 Ontology-Constrained Generation

SAVANT formulates adjuvant generation as a two-stage procedure. The first stage produces an unconstrained natural-language rationale from the design goal, allowing the LLM to explore candidate formulations, immune-effect hypotheses, and mechanistic explanations. The second stage uses this rationale as context and translates it into the DSL representation introduced in Section 3.1.

Let π_1 and π_2 denote the prompts for rationale generation and DSL generation. Given a design goal x , the first stage samples

$$z \sim P_\theta(\cdot \mid \pi_1(x)), \quad (6)$$

where z is a natural-language rationale. The second stage generates a structured proposal:

$$y \sim P_\theta^\mathcal{O}(\cdot \mid \pi_2(x, z)), \quad (7)$$

where $y = (\mathcal{C}, \mathcal{E}, \mathcal{M})$ and $\mathcal{O}$ is the domain ontology.

We represent the ontology as

$$\mathcal{O} = (\mathcal{T}, \mathcal{R}, \mathcal{A}, \Gamma_\mathcal{O}), \quad (8)$$

where $\mathcal{T}$ defines the type system, $\mathcal{R}$ defines admissible design and biological relations, $\mathcal{A}$ defines argument roles and value types, and $\Gamma_\mathcal{O}$ denotes structural constraints over construction claims, immune-effect claims, and mechanism DSL chains.

During DSL generation, a finite-state controller maintains a generation state q_t . At each step, the model selects a grammar-level action a_t , such as filling a field, choosing a relation, or extending a mechanism chain. The controller derives a logic mask over ontology-compatible actions:

$$\mathcal{L}_t = \text{Mask}(q_t, \mathcal{O}). \quad (9)$$

Let $\eta_t = (a_{<t}, x, z)$ denote the generation context. The masked action distribution is

$$\tilde{P}_\theta(a_t \mid \eta_t) = P_\theta(a_t \mid \eta_t) \mathbf{1}[a_t \in \mathcal{L}_t], \quad (10)$$

where $\mathbf{1}[\cdot]$ is the indicator function. It is then renormalized over valid actions:

$$P_\theta^\mathcal{O}(a_t \mid \eta_t) = \frac{\tilde{P}_\theta(a_t \mid \eta_t)}{\sum_{a' \in \mathcal{L}_t} \tilde{P}_\theta(a' \mid \eta_t)}. \quad (11)$$

After each action, the controller updates the generation state:

$$q_{t+1} = \delta(q_t, a_t). \quad (12)$$

This design separates exploratory scientific reasoning from symbolic rendering: the first stage preserves LLM flexibility, while the second produces a domain-compatible DSL object for neuro-symbolic verification. Prompt templates and ontology details are provided in Appendices E.1 and F. We further conduct human validation of DSL rendering, the evaluation protocol and results are provided in Appendix B.2.

3.4 Neuro-Symbolic Proof under Incomplete Knowledge

Given a structured proposal $y = (\mathcal{C}, \mathcal{E}, \mathcal{M})$, SAVANT converts its components into proof obligations. We write each obligation as

$$\tau = b_\tau \Rightarrow h_\tau, \quad (13)$$

where b_τ is an optional premise predicate and h_τ is the target predicate to be supported. The theorem library $\mathcal{K}$ consists of single-premise Horn clauses:

$$\mathcal{K} = \{b_j \Rightarrow p_j\}_{j=1}^N, \quad (14)$$

where each rule has a premise predicate b_j , and a conclusion predicate p_j in the shared predicate space.

SAVANT constructs three types of proof obligations. Construction queries ask whether formulation predicates have precedent support in the literature:

$$\mathcal{Q}_c(y) = \{\emptyset \Rightarrow c \mid c \in \mathcal{C}\}. \quad (15)$$

Immune-effect queries ask whether the claimed immune-effect predicates are supported as effect-side statements:

$$\mathcal{Q}_e(y) = \{\emptyset \Rightarrow e \mid e \in \mathcal{E}\}. \quad (16)$$

For these two query types, SAVANT performs conclusion-side matching: it searches for rules in $\mathcal{K}$ whose conclusion predicate p_j matches c or e , while the premise b_j is not required to match the query. Thus, construction and immune-effect

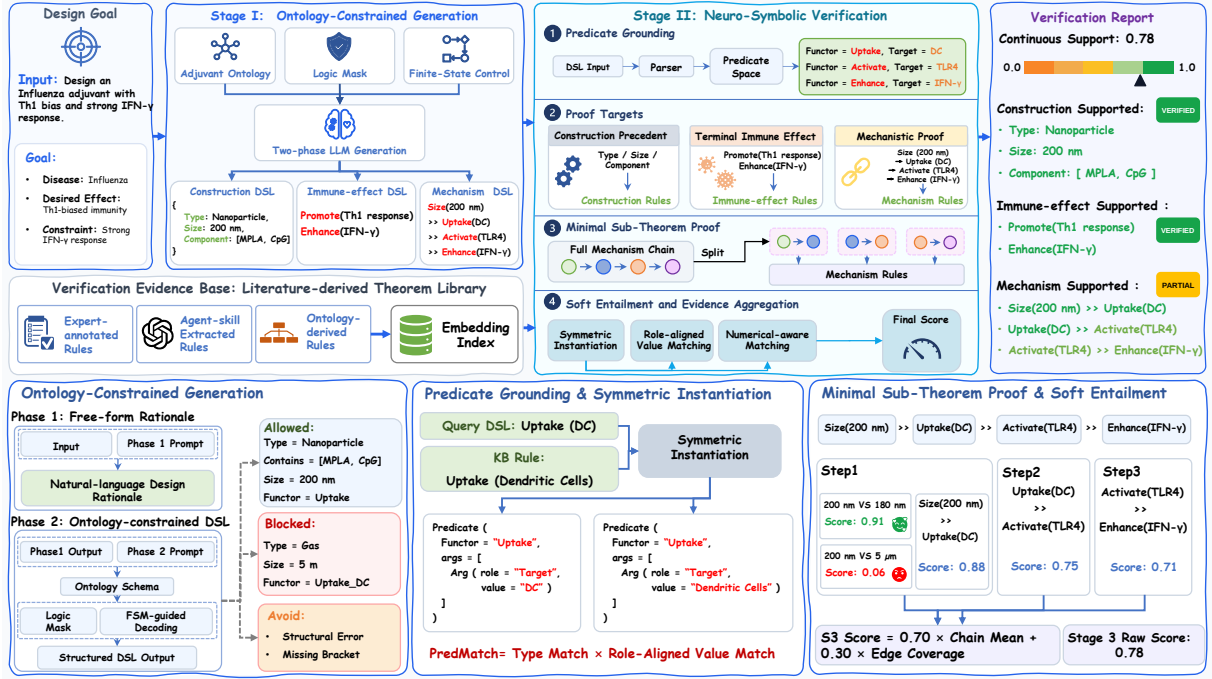


Figure 2: Overview of SAVANT: two-stage generation followed by neuro-symbolic verification.

queries evaluate whether a predicate itself has literature support, rather than whether a causal transition is proven.

Mechanistic queries are obtained by decomposing each chain in $\mathcal{M}$ into adjacent transitions. For a chain

$$m^{(k)} = p_0^{(k)} \gg p_1^{(k)} \gg \dots \gg p_{T_k}^{(k)}, \quad (17)$$

each adjacent pair yields a query asking whether the transition is supported by a rule in $\mathcal{K}$:

$$\mathcal{Q}_m(y) = \{ p_{t-1}^{(k)} \Rightarrow p_t^{(k)} \mid m^{(k)} \in \mathcal{M}, t = 1, \dots, T_k \}. \quad (18)$$

Unlike construction and immune-effect queries, mechanistic queries require transition-side matching: both the premise b_j and conclusion p_j of a rule must match the queried transition.

The full query set is therefore

$$\mathcal{Q}(y) = \mathcal{Q}_c(y) \cup \mathcal{Q}_e(y) \cup \mathcal{Q}_m(y). \quad (19)$$

Since adjuvant evidence is fragmented across empirical studies, verification must handle terminology variation and incomplete proof chains. SAVANT addresses this setting through two principles: minimal sub-theorem proof and symmetric instantiation.

Minimal sub-theorem proof. SAVANT does not require an entire generated design or mechanism to be derived as a single proof. This requirement is too strong for adjuvant design, where the theorem library is incomplete, and the field

lacks a closed first-principles rule system. Instead, SAVANT treats each construction claim, immune-effect claim, and adjacent mechanistic transition as a minimal sub-theorem.

For construction and immune-effect queries, the sub-theorem is a predicate-level obligation of the form $\emptyset \Rightarrow h_\tau$ and is evaluated by conclusion-side matching against the theorem library. For mechanism chains, SAVANT decomposes each chain $m^{(k)} \in \mathcal{M}$ into adjacent transitions:

$$\tau_t^{(k)} = p_{t-1}^{(k)} \Rightarrow p_t^{(k)}, \quad t = 1, \dots, T_k. \quad (20)$$

Each transition-level sub-theorem is evaluated by matching both its premise and conclusion against a single-premise rule in $\mathcal{K}$.

This proof strategy preserves local evidence even when a complete reasoning chain is not fully supported. It separates supported steps from unverified ones, rather than rejecting the entire design as a single unit. The resulting sub-theorem scores are aggregated into stage-level scores in Section 3.5.

Symmetric instantiation. SAVANT instantiates both generated proof obligations and literature-derived rules into the same predicate structure before matching. This is symmetric because the same grounding function Φ is applied to both sides. Matching is head-first: SAVANT first resolves the conclusion-side functor through exact matching, proxy functor mapping, and semantic fallback, and then compares argument values under aligned semantic roles. For transition

Algorithm 1: Symmetric Instantiation

Input: Proof obligations $\mathcal{Q}(y)$, theorem library $\mathcal{K}$
Output: Query-level scores $\{\rho(\tau)\}$

```

1 foreach  $\tau \in \mathcal{Q}(y)$  do
2    $\hat{\tau} \leftarrow \Phi(\tau)$   $\mathcal{R}_\tau \leftarrow \text{HeadCandidates}(\hat{\tau}.h.f, \mathcal{K})$ 
3    $\rho(\tau) \leftarrow 0$ 
4   foreach  $r \in \mathcal{R}_\tau$  do
5      $\hat{r} \leftarrow \Phi(r)$   $s_h \leftarrow \text{PredMatch}(\hat{\tau}.h, \hat{r}.p)$  if
6        $s_h < \lambda_{\text{head}}$  then
7         continue
8     if  $\hat{\tau}.b = \emptyset$  then
9        $s \leftarrow s_h$ 
10    else
11       $s_b \leftarrow \text{PredMatch}(\hat{\tau}.b, \hat{r}.b)$ 
12       $s \leftarrow \text{EdgeMatch}(s_b, s_h, \hat{\tau}, \hat{r})$ 
13     $\rho(\tau) \leftarrow \max(\rho(\tau), s)$ 
14 return  $\{\rho(\tau)\}$ 

```

queries, the same procedure is also applied to the premise side.

Here λ_{head} is the minimum conclusion-side match score for candidate rules. PredMatch first compares predicate functors and then role-aligned argument values. For transition queries, EdgeMatch combines premise-side and conclusion-side predicate matches with a functor-compatibility decay. The exact scoring form is provided in Appendix C.

3.5 Soft Entailment and Score Aggregation

SAVANT assigns each proof obligation a continuous support score rather than a binary decision. Predicate matching follows the functor-first procedure in Section 3.4. We write the predicate-level match score as

$$\text{PredMatch}(p, q) = s_{\text{type}}(p, q) \cdot s_{\text{value}}(p, q), \quad (21)$$

where s_{type} compares predicate functors and s_{value} measures semantic similarity between argument values under aligned roles. For non-numerical values, s_{value} is computed by semantic matching after normalization, and physical quantities are handled separately by numerical-aware matching.

Numerical-aware matching is used for physical quantities such as particle size, dose, concentration, and sequence length. After unit normalization, let u, u' be the normalized values and d, d' their physical dimensions. We define:

$$s_{\text{num}} = \begin{cases} \exp(-\kappa |\log_{10} x_q - \log_{10} x_r|), & d_q = d_r, \\ \epsilon_{\text{dim}}, & d_q \neq d_r. \end{cases} \quad (22)$$

where x_q, x_r are normalized numerical values, d_q, d_r are physical dimensions, and the configuration values are listed in Appendix C.

Let $\rho(\tau)$ denote the support score of a proof obligation. For construction and immune-effect queries, SAVANT aggregates conclusion-side support as

$$S_1 = G_{\text{guard}}(\text{Agg}_{c \in \mathcal{C}} \rho(\emptyset \Rightarrow c), \Pi_{\mathcal{O}}(y)), \quad (23)$$

$$S_2 = G_{\text{eff}}(\text{Mean}_{e \in \mathcal{E}} \rho(\emptyset \Rightarrow e), \text{Spec}(\mathcal{E})). \quad (24)$$

Here G_{guard} denotes conservative aggregation with ontology-derived caps, $\Pi_{\mathcal{O}}(y)$ denotes ontology-prior support, and G_{eff} combines effect support with effect specificity. The exact forms of G_{guard} and G_{eff} are provided in Appendix C.

For mechanism chains, SAVANT aggregates local edge scores with partial coverage. Let $\rho_t^{(k)}$ be the support score of the t -th edge in chain $m^{(k)}$:

$$\bar{\rho}_{\text{chain}} = \text{Mean}_k \text{Mean}_t \rho_t^{(k)}. \quad (25)$$

We also compute

$$C_{\text{edge}} = \frac{n_{\text{sup}} + 0.5 n_{\text{weak}}}{n_{\text{all}}}, \quad (26)$$

where $\rho_t^{(k)} \geq \theta_{\text{sup}}$ is supported and $0 < \rho_t^{(k)} < \theta_{\text{sup}}$ is weakly supported. The Stage 3 score is

$$S_3 = 0.70 \bar{\rho}_{\text{chain}} + 0.30 C_{\text{edge}}. \quad (27)$$

Stage scores are mapped to the 1–5 scale with stage-specific calibration, detailed in Appendix C.

$$\text{Disc}(S_i) = 1 + \sum_{\ell=1}^4 \mathbf{1}[S_i \geq \delta_{\ell}]. \quad (28)$$

Low-support proof obligations are reported as knowledge gaps. This report separates literature-supported components from assumptions that require further experimental validation.

4 Experiments

4.1 Theorem Library

We use the theorem library summarized in Table 1. Detailed source statistics, stage-specific rule usage, ontology-derived rule construction, and extraction skill objectives are provided in Appendix A.

Category	# Rules	Scope
Expert-annotated	3,365	Curated adjuvant evidence
Skill-extracted	1,433	Literature-derived mechanisms
Ontology-derived	1,684	Structured ontology fields
Total library	6,482	Deduplicated Horn clauses

Table 1: Theorem library sources.

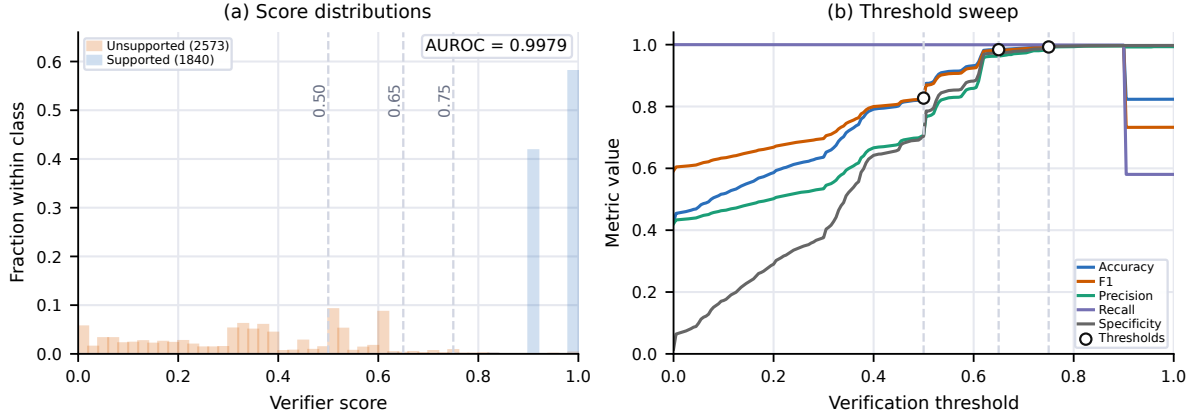


Figure 3: Closed-world verification results. Score distributions show separation between theorem-supported and perturbed instances, and threshold sweep shows classification metrics across operating thresholds.

4.2 Evaluation Data

We use the evaluation data summarized in Table 2. The closed-world verification evaluates verifier behavior under a fixed theorem library, the chain-ablation extension is used only for the minimal sub-theorem ablation, and the expert-aligned set evaluates agreement with human judgments. We additionally conduct a stability evaluation to assess whether stage-level expert alignment remains consistent under changes in both design questions and generators. Detailed dataset construction, perturbation types, annotation protocols, and the stability-evaluation setup are provided in Appendix B.

Dataset	Total	Pos.	Neg.
Closed-world verification	4,413	1,840	2,573
Chain-ablation extension	600	300	300
Expert-aligned set	51	—	—
Stability evaluation	60	—	—

Table 2: Evaluation datasets used in our experiments.

4.3 Implementation Details

For the expert-aligned evaluation, all methods are evaluated on the same design responses generated by GPT-4o (OpenAI), for the stability evaluation, design responses are generated by Qwen3-8B (Yang et al., 2025) and Kimi-k2.6 (Team et al., 2026), with the latter also serving as an LLM-as-Judge evaluator in the expert-aligned evaluation. SAVANT uses BioLORD-2023-M for non-numerical biomedical value matching (Remy et al., 2024). Baselines include BM25 (Robertson and Zaragoza, 2009), sentence-embedding similarity (Reimers and Gurevych, 2019), and LLM-as-Judge evaluators including GPT-5, DeepSeek-v4-

Flash (DeepSeek-AI, 2026), DeepSeek-R1 (Guo et al., 2025), Qwen3.5-Plus (Qwen Team, 2026), Gemini-3-Flash-Preview (Google AI for Developers, 2025), and GLM-5.1 (GLM-5-Team et al., 2026). Detailed RAG configurations are provided in Appendix D. In ablations, removed verification modules are replaced with direct embedding-style matching over serialized predicates. Prompts are provided in Appendix E.2.

4.4 Evaluation Metrics

For closed-world verification, we report AUROC, PR-AUC, and threshold metrics at $\tau \in \{0.50, 0.65, 0.75\}$, with per-type score and verdict diagnostics. For ablations, we report subset-specific metrics, including AUROC, positive-negative mean score gap, negative accuracy at $\tau = 0.65$, and mean score change. For expert-aligned and stability evaluations, we report stage-level Spearman’s ρ against human 1–5 scores for construction quality, immune-effect support, and mechanistic reasoning, the expert-aligned evaluation additionally reports MAE.

4.5 Closed-World Verification Results

We first evaluate SAVANT in a closed-world verification setting, where both positive instances and controlled perturbations are derived from the fixed theorem library. This experiment is an internal evaluation of the verifier and theorem library, testing whether supported proof obligations receive high scores and minimally corrupted variants are rejected. It does not measure open-domain generalization, which we evaluate through expert-aligned assessment in Section 4.7. Such controlled perturbation tests are commonly used in evidence

Mechanism	Ablation set	Metric	Full	Ablated	Δ
Minimal sub-theorem	One-bad-edge chains	AUROC $\uparrow$	0.9881	0.6698	-0.3184
	One-bad-edge chains	Pos-Neg gap $\uparrow$	0.1591	0.0406	-0.1184
Symmetric instantiation	Stage-2 endpoint cases	AUROC $\uparrow$	1.0000	0.7917	-0.2083
	Effect-direction substitutions	AUROC $\uparrow$	0.9999	0.6141	-0.3858
	Causal-direction reversals	AUROC $\uparrow$	0.9912	0.4637	-0.5275
Numerical-aware scoring	Numeric perturbations	NegAcc@0.65 $\uparrow$	1.0000	0.2133	-0.7867
	Numeric perturbations	Neg. mean $\downarrow$	0.1751	0.8128	+0.6377

Table 3: Targeted ablation results. Δ is Ablated minus Full. Pos-Neg gap is the mean positive-negative score difference, larger is better. Neg. mean is the mean score of negative instances, lower is better.

verification, behavioral testing, contrast sets, and knowledge-graph reasoning (Thorne et al., 2018; Ribeiro et al., 2020; Gardner et al., 2020; Bordes et al., 2013).

As shown in Figure 3, SAVANT clearly separates theorem-supported and perturbed instances, achieving 0.9979 AUROC and 0.9944 PR-AUC over 4,413 binary-labeled instances. At a stricter threshold of $\tau = 0.75$, it obtains 0.9925 accuracy and 0.9911 F1 while maintaining near-perfect recall. Full threshold and per-type diagnostics are provided in Appendix G.

4.6 Ablation Study

We evaluate the three core mechanisms of SAVANT using targeted ablation subsets. Each subset replaces the removed component with direct embedding-based matching over serialized predicates, with construction details provided in Appendix H. Table 3 summarizes the results.

Removing minimal sub-theorem proof weakens discrimination on chains with local errors, supporting local proof obligations rather than treating the entire chain as a single unit. Removing symmetric instantiation hurts endpoint matching, antonym substitutions, and causal-direction reversals, showing that direct embedding over serialized predicates fails to preserve predicate type, argument roles, and transition direction. Removing numerical-aware scoring causes numeric perturbations to receive much higher support, confirming that physical quantities require unit- and scale-aware comparison.

4.7 Expert-Aligned Evaluation Results

We further evaluate whether automatic scores align with expert judgments on open-ended adjuvant design responses. As shown in Table 4, SAVANT achieves the highest rank correlations across all three stages, with stronger alignment for immune-effect support and mechanistic reasoning than for

construction quality. The correlations indicate moderate absolute agreement, with expert scores serving as an external reference rather than a unique gold standard. Stage 1 is more sensitive to expert-specific formulation priors because multiple materials, delivery systems, preparation methods, and component combinations may be plausible. By contrast, immune-effect claims and mechanistic transitions are more directly tied to established evidence and correspond more closely to the proof obligations evaluated by SAVANT.

MAE measures proximity to the expert rating scale rather than rank agreement. Kimi-k2.6 has the lowest Stage 1 MAE but nearly zero rank correlation, while DeepSeek-R1 shows negative or near-zero rank correlations together with high MAE. These results indicate that LLM judges may use the same numeric scale differently even under a shared rubric.

Retrieval baselines in Table 5 show weak response-level alignment, suggesting that global text similarity is insufficient for mechanistic assessment. To control for differences in external evidence access, we further evaluate Paper-RAG and Evidence-RAG with GPT-5 and DeepSeek-v4-Flash as judges. Both use the same RAG setting and differ only in the indexed corpus. Paper-RAG indexes 20,577 passages from 53 domain papers and two textbooks, while Evidence-RAG additionally includes 2,529 deduplicated natural-language evidence records associated with the frozen runtime theorem library. The judges receive the underlying natural-language evidence but not method-specific symbolic representations or verification outputs, aligning evidence access without incorporating components of SAVANT into the baselines.

As shown in Table 6, SAVANT achieves the highest correlation in all three stages, although the Stage 3 margin is narrow. Additional rule-associated evidence improves some judge-stage

Method	S1 ρ / MAE	S2 ρ / MAE	S3 ρ / MAE
SAVANT (Ours)	0.24 / 1.27	0.39 / 0.98	0.33 / 0.65
GPT-5	0.18 / 1.06	0.32 / 0.71	0.29 / 0.65
DeepSeek-v4-Flash	0.17 / 1.39	0.28 / 0.82	0.23 / 0.76
Qwen3.5-Plus	0.18 / 1.22	0.08 / 1.02	0.25 / 0.73
Gemini-3-Flash	0.19 / 1.82	0.26 / 1.63	0.09 / 1.18
GLM-5.1	0.14 / 1.02	0.18 / 0.75	0.16 / 0.82
Kimi-k2.6	0.01 / 0.86	0.03 / 0.75	0.15 / 1.00
DeepSeek-R1	-0.12 / 2.02	-0.18 / 1.45	-0.01 / 1.27

Table 4: Stage-level expert alignment.

Retrieval scorer	ρ	MAE
BM25	0.08	0.92
Embedding similarity	-0.10	0.78

Table 5: Single-score retrieval diagnostics.

combinations but does not produce consistent gains, indicating that textual evidence access alone does not reproduce structured verification. This distinction is important for open-ended adjuvant design, where a new formulation often has no complete precedent and supporting evidence for construction, immune effects, and mechanistic transitions may come from different sources. SAVANT maps these claims and literature evidence into a shared predicate space and verifies local proof obligations, whereas the RAG baselines judge complete responses holistically from retrieved text. This decomposition also produces proof traces that localize supported obligations and low-support knowledge gaps, Appendix I provides representative examples.

4.8 Stability Evaluation

We further examine whether stage-level expert alignment remains stable when the design questions and generators change. Domain experts created 30 additional open-ended design questions, and Qwen3-8B and Kimi-k2.6 each generated one response per question, yielding 60 expert-scored responses. We use the same expert team, blinded procedure, and three-stage rating criteria as in the expert-aligned evaluation. The stability comparison includes SAVANT and the two strongest LLM-as-Judge baselines in the expert-aligned evaluation, GPT-5 and DeepSeek-v4-Flash.

As shown in Table 7, SAVANT exhibits the smallest variation in stage-level rank correlations and maintains the highest Stage 2 and Stage 3 correlations across all settings. This stability is consistent with the stage-wise verification design: responses from different generators are rendered into

Method	Stage 1 ρ	Stage 2 ρ	Stage 3 ρ
<i>Paper-RAG</i>			
GPT-5	0.14	0.31	0.26
DeepSeek-v4-Flash	0.14	0.13	0.12
<i>Evidence-RAG</i>			
GPT-5	-0.02	0.12	0.21
DeepSeek-v4-Flash	0.05	0.17	0.32
SAVANT (Ours)	0.24	0.39	0.33

Table 6: Stage-level expert alignment with RAG.

Method	Stage 1 ρ	Stage 2 ρ	Stage 3 ρ
<i>GPT-4o responses (n = 51)</i>			
SAVANT	0.24	0.39	0.33
GPT-5	0.18	0.32	0.29
DeepSeek-v4-Flash	0.17	0.28	0.23
<i>Qwen3-8B and Kimi-k2.6 responses (n = 60)</i>			
SAVANT	0.32	0.33	0.37
GPT-5	0.30	0.13	0.18
DeepSeek-v4-Flash	0.44	0.19	-0.04
<i>Combined responses (n = 111)</i>			
SAVANT	0.27	0.37	0.42
GPT-5	0.24	0.26	0.29
DeepSeek-v4-Flash	0.36	0.24	0.06

Table 7: Stage-level stability evaluation.

a shared DSL, while immune-effect claims and mechanism transitions are evaluated as local proof obligations against the same theorem library. This process reduces sensitivity to generator-specific wording and holistic scoring. The advantage is strongest in Stages 2 and 3 because these stages operate on explicit immune-effect predicates and directed mechanism edges, whereas construction assessment allows multiple plausible formulations.

5 Conclusion

We presented SAVANT, a neuro-symbolic verification framework for adjuvant design. SAVANT combines two-stage generation with literature-grounded verification, using minimal sub-theorem proof, symmetric instantiation, and numerical-aware matching to assess construction, immune-effect, and mechanistic claims under incomplete knowledge. Experiments show that SAVANT separates theorem-supported claims from controlled perturbations, exhibits the most stable stage-level expert alignment, and achieves the strongest alignment for immune-effect support and mechanistic reasoning.

Looking forward, SAVANT can serve as a reflection module for LLMs and provide structured reward for post-training toward evidence-grounded, mechanistically interpretable generation.

Acknowledgement

This work is sponsored by the National Key Research and Development Program of China (2023YFC2307700), Zhongguancun Academy Project No.02012501, CAS Project for Young Scientists in Basic Research (YSBR-083), National Natural Science Foundation of China (NO.62506357), Lingang Laboratory (Grant No. LGL-2616-04).

Limitations

SAVANT verifies literature support rather than empirical vaccine efficacy. This is a domain-level limitation of applying automated verification to adjuvant design. A high verification score indicates that a proposal is structurally well-formed and supported by the current symbolic evidence base. It does not imply safety, immunogenicity, manufacturability, or efficacy in wet-lab or clinical settings. Experimental validation remains necessary before drawing biological or translational conclusions.

SAVANT also depends on the coverage and quality of the theorem library. Although the library combines expert-annotated rules, skill-extracted rules, and ontology-derived rules, it remains incomplete and may reflect biases in the available literature. Plausible but under-studied mechanisms may therefore be marked as knowledge gaps. Low support should be interpreted as insufficient evidence in the current theorem library, not as evidence of biological invalidity.

Our closed-world verification evaluates the internal discriminative capability of the verifier and theorem library. Because positives and controlled perturbations are defined against the same evidence base, this setting does not measure open-domain generalization. We complement it with expert-aligned and stability evaluations across multiple generators, but these remain limited by sample size and the assigned expert-scoring protocol. The observed correlations indicate moderate absolute agreement rather than a unique gold standard for open-ended designs. Moreover, the stability evaluation changes both questions and generators, measuring robustness to a combined distribution shift rather than an isolated generator effect. Construction-quality alignment remains more sensitive to expert-specific formulation priors than the immune-effect and mechanism stages. Large-scale crowdsourced annotation is not appropriate because valid assessment requires specialized im-

munological expertise. Further evaluation should cover broader design contexts and generators and include more redundant expert scoring.

SAVANT relies on ontology-constrained DSL rendering. Although human validation of the audited subset shows high conceptual fidelity, residual errors may alter causal ordering, omit intermediate steps, or introduce unstated construction details. The structured DSL improves validity, reproducibility, and proof construction, but it may constrain designs to the current ontology, schema, and functor set. Extending the ontology, improving rule extraction, and using verification outputs for model reflection or post-training remain important directions for future work.

Ethical Considerations

SAVANT is a literature-grounded verification framework and does not establish vaccine safety, immunogenicity, manufacturability, or clinical efficacy. Its outputs should not be interpreted as experimental or medical recommendations. Any practical adjuvant proposal would require expert review, biosafety assessment, regulatory oversight, and wet-lab validation.

Human expert evaluation is used only as an external reference for reliability analysis and error diagnosis. The expert scores are not used to train SAVANT, tune verifier parameters, select model outputs, or serve as an absolute source of ground truth. Since each generated response is assigned to one domain expert under a shared rubric, the protocol is closer to a realistic expert-review setting than to a redundantly annotated agreement benchmark.

The data used in this study are derived from public literature, ontology resources, theorem-library construction, and model-generated design responses. They do not involve patient-level data, clinical samples, animal experiments, private health information, personally identifiable information, or sensitive user information. Human experts participate only in evaluating generated text responses.

A key ethical risk of LLM-based design is that fluent proposals may appear more reliable than their evidence base. SAVANT aims to mitigate this risk by exposing supported proof obligations and low-support knowledge gaps. These gaps indicate insufficient evidence in the current theorem library rather than definitive biological invalidity. The framework is intended to support expert re-

view and evidence-grounded reasoning, not to provide actionable experimental protocols.

References

- Daniil A Boiko, Robert MacKnight, Ben Kline, and Gabe Gomes. 2023. Autonomous chemical research with large language models. *Nature*, 624(7992):570–578.
- Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. *Advances in neural information processing systems*, 26.
- Yi Chen, Yu Zhang, Jian Xu, Hua Yue, Xinming Wang, Zequan Lyu, Xu-Yao Zhang, Wei Wei, and Cheng-Lin Liu. 2026. [An open-ended benchmark and formal framework for adjuvant research with MLLM](#). In *The Fourteenth International Conference on Learning Representations*.
- Robert L Coffman, Alan Sher, and Robert A Seder. 2010. Vaccine adjuvants: putting innate immunity to work. *Immunity*, 33(4):492–503.
- John Dagdelen, Alexander Dunn, Sanghoon Lee, Nicholas Walker, Andrew S Rosen, Gerbrand Ceder, Kristin A Persson, and Anubhav Jain. 2024. Structured information extraction from scientific text with large language models. *Nature communications*, 15(1):1418.
- DeepSeek-AI. 2026. Deepseek-v4: Towards highly efficient million-token context intelligence.
- Sebastian Farquhar, Jannik Kossen, Lorenz Kuhn, and Yarin Gal. 2024. Detecting hallucinations in large language models using semantic entropy. *Nature*, 630(8017):625–630.
- Tianyu Gao, Howard Yen, Jiatong Yu, and Danqi Chen. 2023. Enabling large language models to generate text with citations. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 6465–6488.
- Matt Gardner, Yoav Artzi, Victoria Basmov, Jonathan Berant, Ben Bogin, Sihao Chen, Pradeep Dasigi, Dheeru Dua, Yanai Elazar, and Ananth Gottumukkala. 2020. Evaluating models’ local decision boundaries via contrast sets. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1307–1323.
- GLM-5-Team, :, Aohan Zeng, Xin Lv, Zhenyu Hou, Zhengxiao Du, Qinkai Zheng, Bin Chen, Da Yin, Chendi Ge, Chenghua Huang, Chengxing Xie, Chenzheng Zhu, Congfeng Yin, Cunxiang Wang, Gengzheng Pan, Hao Zeng, Haoke Zhang, Hao-ran Wang, and 168 others. 2026. [Glm-5: from vibe coding to agentic engineering](#). *Preprint*, arXiv:2602.15763.
- Google AI for Developers. 2025. Gemini 3 Developer Guide. <https://ai.google.dev/gemini-api/docs/gemini-3>. Accessed: 2026-05-22.
- Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, and Xiao Bi. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.
- Yeonghun Kang and Jihan Kim. 2024. Chatmof: an artificial intelligence system for predicting and generating metal-organic frameworks using large language models. *Nature communications*, 15(1):4705.
- Ed C Lavelle and Craig P Mcentee. 2024. Vaccine adjuvants: Tailoring innate recognition to send the right message. *Immunity*, 57(4):772–789.
- Andres M. Bran, Sam Cox, Oliver Schilter, Carlo Baldassari, Andrew D White, and Philippe Schwaller. 2024. Augmenting large language models with chemistry tools. *Nature machine intelligence*, 6(5):525–535.
- Potsawee Manakul, Adian Liusie, and Mark Gales. 2023. Selfcheckgpt: Zero-resource black-box hallucination detection for generative large language models. In *Proceedings of the 2023 conference on empirical methods in natural language processing*, pages 9004–9017.
- Robin Manhaeve, Sebastijan Dumancic, Angelika Kimmig, Thomas Demeester, and Luc De Raedt. 2018. Deepproblog: Neural probabilistic logic programming. *Advances in neural information processing systems*, 31.
- Joshua Monickaraj, Hasin Rehana, Ani Bernardi, Yuping Zheng, Taiyu Lin, Le Liu, Amogh Madireddi, Leo Yeh, Jie Zheng, Junguk Hur, and et al. Vaxjo 2.0: A web-based vaccine adjuvant and adjuvant mechanism knowledgebase powered by ontology and large language model. *Frontiers in Cellular and Infection Microbiology*, 16:1763384.
- Yayoi Natsume-Kitatani, Kouji Kobiyama, Yoshinobu Igarashi, Taiki Aoshi, Noriyuki Nakatsu, Lokesh P Tripathi, Junichi Ito, Johan Nyström-Persson, Yuji Kosugi, Rodolfo S Allendes Osorio, and et al. 2025. An adjuvant database for preclinical evaluation of vaccines and immunotherapeutics. *Cell Chemical Biology*, 32(8):1075–1088.
- OpenAI. Openai. <https://openai.com/index>. Accessed: 2026-05-23.
- Derek T O’Hagan and Christopher B Fox. 2015. New generation adjuvants—from empiricism to rational design. *Vaccine*, 33:B14–B20.
- Bali Pulendran, Prabhu S. Arunachalam, and Derek T O’Hagan. 2021. Emerging concepts in the science of vaccine adjuvants. *Nature reviews Drug discovery*, 20(6):454–475.

- Qwen Team. 2026. [Qwen3.5: Towards native multi-modal agents](#).
- Steven G Reed, Mark T Orr, and Christopher B Fox. 2013. Key roles of adjuvants in modern vaccines. *Nature medicine*, 19(12):1597–1608.
- Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP)*, pages 3982–3992.
- Wesley F Reinhart and Antonia Statt. 2024. Large language models design sequence-defined macro-molecules via evolutionary optimization. *npj Computational Materials*, 10(1):262.
- François Remy, Kris Demuynck, and Thomas De-meester. 2024. Biolord-2023: semantic textual representations fusing large language models and clinical knowledge graph insights. *Journal of the American Medical Informatics Association*, 31(9):1844–1855.
- Marco Tulio Ribeiro, Tongshuang Wu, Carlos Guestrin, and Sameer Singh. 2020. Beyond accuracy: Behavioral testing of nlp models with checklist. In *Proceedings of the 58th annual meeting of the association for computational linguistics*, pages 4902–4912.
- Stephen Robertson and Hugo Zaragoza. 2009. *The probabilistic relevance framework: BM25 and beyond*, volume 4. Now Publishers Inc.
- Tim Rocktäschel and Sebastian Riedel. 2017. End-to-end differentiable proving. *Advances in neural information processing systems*, 30.
- Kimi Team, Tongtong Bai, Yifan Bai, Yiping Bao, S. H. Cai, Yuan Cao, Y. Charles, H. S. Che, Cheng Chen, Guanduo Chen, Huarong Chen, Jia Chen, Jiahao Chen, Jianlong Chen, Jun Chen, Kefan Chen, Liang Chen, Ruijue Chen, Xinhao Chen, and 307 others. 2026. [Kimi k2.5: Visual agentic intelligence](#). Preprint, arXiv:2602.02276.
- James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 2018. Fever: a large-scale dataset for fact extraction and verification. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 809–819.
- Joanna L Turley and Ed C Lavelle. 2022. Resolving adjuvant mode of action to enhance vaccine efficacy. *Current opinion in immunology*, 77:102229.
- David Wadden, Shanchuan Lin, Kyle Lo, Lucy Lu Wang, Madeleine van Zuylen, Arman Cohan, and Hannaneh Hajishirzi. 2020. Fact or fiction: Verifying scientific claims. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7534–7550.
- An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, and et al. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.
- Fan Yang, Zhilin Yang, and William W Cohen. 2017. Differentiable learning of logical rules for knowledge base reasoning. *Advances in neural information processing systems*, 30.
- Kamyar Zeinalipour, Neda Jamshidi, Monica Bianchini, Marco Maggini, and Marco Gori. 2024. Design proteins using large language models: enhancements and comparative analyses. In *Proceedings of the 1st Workshop on Language+ Molecules (L+ M 2024)*, pages 34–47.
- Tingmei Zhao, Yulong Cai, Yujie Jiang, Xuemei He, Yuquan Wei, Yifan Yu, and Xiaohe Tian. 2023. Vaccine adjuvants: mechanisms and platforms. *Signal transduction and targeted therapy*, 8(1):283.
- Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, and et al. 2023a. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in neural information processing systems*, 36:46595–46623.
- Yuping Zheng and Yongqun He. 2025. Vaxjognn: A graph neural network for ontology-grounded vaccine adjuvant recommendation. *bioRxiv*, pages 2025–11.
- Zhiling Zheng, Oufan Zhang, Christian Borgs, Jennifer T Chayes, and Omar M Yaghi. 2023b. Chatgpt chemistry assistant for text mining and the prediction of mof synthesis. *Journal of the American Chemical Society*, 145(32):18048–18062.

Appendix Overview

This appendix provides supplementary material for reproducibility and clarity. It is organized as follows:

- [Appendix A: Theorem Library](#)
- [Appendix B: Evaluation Data](#)
- [Appendix C: Scoring Details](#)
- [Appendix D: RAG Baseline Configuration](#)
- [Appendix E: LLM Prompts](#)
- [Appendix F: Ontology and FSM](#)
- [Appendix G: Closed-World Details](#)
- [Appendix H: Ablation Study Details](#)
- [Appendix I: Case Study and Verification Trace](#)

A Theorem Library Construction and Statistics

A.1 Overview

The theorem library is the evidence base used by SAVANT to verify generated adjuvant designs. It

contains 6,482 deduplicated single-premise Horn clauses, all represented in the shared predicate space defined in Section 3.1. Each rule has the form $b_j \Rightarrow p_j$, where both the premise b_j and conclusion p_j are relation–role–value predicates.

The library is constructed from three complementary sources, summarized in Table 8. Expert-annotated rules provide curated evidence from paper-level QA annotations and form the core support for construction and immune-effect verification. Skill-extracted rules are produced by a GPT-5.5 + Skills workflow to expand coverage of mechanistic transitions. Ontology-derived rules are converted from structured Vaccine Ontology / Vaxjo 2.0 templates (Monickaraj et al.) and provide additional standardized background knowledge. The following sections describe these sources, the formalization and normalization process, and the stage-specific usage of the resulting rules.

A.2 Expert-Annotated Rules

The expert-annotated rules are derived from paper-level adjuvant annotations produced by domain experts. For each annotated paper, experts directly write structured DSL annotations covering three components of adjuvant design: construction claims, immune-effect claims, and mechanistic rationales. These annotations provide the most curated source of evidence in the theorem library and align with the formal decomposition $y = (\mathcal{C}, \mathcal{E}, \mathcal{M})$ used by SAVANT.

We convert the expert-written DSL into the shared predicate space. Construction and immune-effect annotations are converted into predicate-level support rules, while mechanistic rationales are decomposed into local transition rules. After normalization and deduplication, this source contributes 3,365 single-premise Horn clauses. The expert-annotated rules form the core evidence base for Stage 1 and Stage 2, and also provide mechanism rules used in Stage 3.

A.3 Skill-Extracted Rules

The skill-extracted rules are used to expand the mechanism coverage of the theorem library, especially for Stage 3 verification. We use a Codex GPT-5.5 + Skills workflow with two extraction skills: `immunology-mechanism-extractor` for immune-core mechanism transitions, and `adjuvant-immune-bridge-extractor` for bridge rules connecting adjuvant materials, properties, or delivery features to immune mechanism

nodes. These rules are extracted from literature evidence only, expert scores and evaluation labels are never used to select or validate extracted rules.

The extraction workflow first produces complete source-grounded DSL chains, rather than directly writing isolated theorem edges. Adjacent steps in each accepted chain are then lowered into atomic single-premise Horn clauses of the form $b_j \Rightarrow p_j$. Non-adjacent multi-hop paths are stored only as derived sidecar indexes and are not treated as atomic theorem rules. This design preserves mechanism directionality while keeping the final theorem library compatible with the single-premise proof format used by SAVANT. Together, the two skill libraries contribute 1,433 accepted rules to the theorem library.

General Mechanism Extraction Skill

Implementation skill:

`immunology-mechanism-extractor`

Objective: Extract immune-mechanism event chains from literature evidence and lower adjacent mechanism steps into verifiable single-premise Horn clauses.

Scope: Immune-core pathway and mechanism transitions. This skill excludes construction and formulation-entry predicates.

Input evidence: Paper title, PMID, DOI, source path, abstract, section text, textbook or review evidence, open-access PDF text, PubMed/PMC or journal-page evidence, keyword-centered PDF snippets, and checked paper-level mechanism-chain records.

Intermediate output: A source-grounded DSL chain with ordered mechanism events, source metadata, evidence level, evidence text, confidence, confidence basis, check status, and candidate atomic edges.

Final output: Adjacent DSL edges lowered into single-premise Horn clauses with premise predicate, conclusion predicate, evidence text, source identifier, confidence, library scope, plugin identifier, and step index.

Adjuvant-Bridge Extraction Skill

Implementation skill:

`adjuvant-immune-bridge-extractor`

Objective: Extract bridge chains that connect adjuvant materials, properties, or delivery features to the first immune mechanism node.

Scope: Adjuvant-to-mechanism entry rules. This skill focuses on property, material, or delivery-feature bridges rather than raw composition facts.

Input evidence: The same source-grounded literature inputs used for mechanism extraction, including paper metadata, abstracts, sections, PDF snippets, journal pages, PubMed/PMC evidence, and curated mechanism-chain records.

Intermediate output: A source-grounded adjuvant-bridge DSL chain with evidence metadata, confidence basis, and atomic edge candidates.

Final output: Adjacent bridge steps lowered into single-premise Horn clauses with source traceability, evidence text, confidence, library scope, and step index.

Each accepted rule must contain source trace-

Source	# Rules	Source scale	Role in theorem library
Expert-annotated	3,365	51 paper-level QA annotations, 49 traceable paper IDs in the final rule library	Curated evidence for construction, immune-effect, and mechanism rules
Skill-extracted	1,433	GPT-5.5 + Skills extraction over 94 unique literature or section sources	Additional mechanistic transition rules for expanding Stage 3 coverage
Ontology-derived	1,684	Curated Vaccine Ontology / Vaxjo 2.0 structured templates (Monickaraj et al.)	Standardized background rules derived from ontology fields rather than free-text extraction
Total	6,482	–	Deduplicated single-premise Horn clauses in the shared predicate space

Table 8: Source composition of the SAVANT theorem library.

Extraction library	Input chains	Atomic edges	Duplicates removed	Accepted rules	Sources
General mechanism	374	1,222	53	1,169	69
Adjuvant bridge	57	277	13	264	28
Merged skill-extracted rules	431	1,499	66	1,433	94

Table 9: Statistics of skill-extracted rules. The two skill libraries together produce the 1,433 skill-extracted rules reported in the main theorem library. The merged source count is 94 because the two libraries share three overlapping sources.

ability, evidence text, confidence, confidence basis, and a valid relation–role–value predicate representation. External search is used only to clarify or corroborate a mechanism when local evidence is incomplete or ambiguous, it is not used to invent unverified rules. Model memory or general common sense cannot replace source-grounded evidence. If a rule depends on multiple evidence pieces, the supporting sources must be recorded in the confidence basis, and the rule is marked as context-limited when appropriate.

We apply automatic validation after extraction and lowering. The checks include JSON/JSONL schema validation, DSL parsing, single-premise validation, source traceability, required evidence text, confidence range checking, evidence-level checking, check-status validation, functor normalization with a controlled functor set, target-granularity checking, scope checking, adjacent-edge validation, derived-path validation, and duplicate removal. The final validated outputs contain 1,169 accepted general-mechanism rules and 264 accepted adjuvant-bridge rules.

A.4 Manual Spot Check

We further conduct a manual spot check to characterize the quality of skill-extracted rules after automatic validation, lowering, and duplicate removal. This check serves as an external audit of the validated skill-extracted rule pool and is not used for rule construction, theorem-library filter-

ing, or downstream experimental evaluation. As shown in Table 10, one vaccine-adjuvant expert reviewed a random sample of 112 skill-extracted theorem rules.

The expert found no systematic correctness issues in the sampled rules. Only one theorem rule was judged factually incorrect. The remaining comments mainly concerned rule specificity and predicate wording. Three rules were considered to have limited informational value because they expressed overly broad relations, such as generic statements that an adjuvant activates immunity without specifying an intermediate mechanism or terminal endpoint. Thirteen rules were judged to use less standard predicate wording, for example, using a generic causal functor where a more specific immunological functor would better match the intended relation. Overall, this spot check supports the reliability of the skill-extraction workflow for producing usable mechanistic rules after automatic validation, while indicating that residual issues primarily involve informativeness and surface formalization rather than broad biological correctness.

A.5 Ontology-Derived Rules

The ontology-derived rules are converted from the structured Vaccine Ontology / Vaxjo 2.0 template `vaccine_adjuvant.csv` (Monickaraj et al.). These rules use only curated ontology fields. Free-text Vaxjo-LLM mechanism summaries are

Spot-check item	Count	Share
Sampled rules for expert review	112	100.0%
Factually incorrect	1	0.9%
Limited informativeness	3	2.7%
Non-standard functor wording	13	11.6%

Table 10: Manual spot check of skill-extracted theorem rules after automatic validation and duplicate removal. The issue categories summarize expert comments over 112 randomly sampled rules and are not used for theorem-library filtering.

excluded from the formal theorem library.

Only three rule types are generated, as shown in Table 12. All final rules are single-premise Horn clauses and contain no co-conditions.

The ontology-derived rules are used only in the Stage 3 effective library, even when their `source_type` is `immune_effect`. They are not used for Stage 1 construction validation or Stage 2 immune-effect scoring.

We apply field filtering, normalization, deduplication, and traceability checks before rule construction. The main checks are summarized in Table 14. Here, source IDs refer to deduplicated `paper_id` values recorded in the final Horn rules, whereas processed ontology records refer to VO/Vaxjo entries.

A.6 Deduplication and Rule Normalization

Deduplication is performed at the structured rule level, rather than by embedding similarity or semantic clustering. The general principle is to first normalize each extracted or converted candidate into a canonical DSL or predicate rule, and then remove duplicates using deterministic canonical keys. This makes the process reproducible and avoids merging rules that are semantically related but differ in direction, condition, conclusion, role, or normalized value.

For skill-extracted rules, the deduplication unit is an atomic single-step Horn rule. Each extracted mechanism chain is first lowered into adjacent DSL edges, and each edge is normalized into a canonical form. The deduplication key is the canonical edge DSL:

$$\text{key}_{\text{skill}} = \text{CanonicalEdgeDSL}(b_j \Rightarrow p_j). \quad (29)$$

If two candidates normalize to the same edge, only one rule is retained. When duplicate candidates are supported by different evidence, the retained rule keeps the preferred evidence record, prior-

itizing higher-confidence support and preserving traceability where available. Rules that are only semantically similar are not merged unless they map to the same canonical edge.

For ontology-derived rules, the deduplication unit is the normalized structured relation produced after field conversion. Deduplication is not performed by ontology term ID or source ID. Instead, each rule candidate is normalized into one of three relation keys:

$$\begin{aligned} &(\text{component_profile}, c, u), \\ &(\text{component_receptor}, c, r), \\ &(\text{receptor_profile}, r, u), \end{aligned} \quad (30)$$

where c is the normalized component or adjuvant alias, r is the normalized receptor value, and u is the normalized immune-profile value. The corresponding deduplication key is:

$$\text{key}_{\text{onto}} = (\text{relation_type}, \text{condition_value}, \text{effect_value}). \quad (31)$$

Aliases are retained as separate matching entries when they produce distinct normalized component values. If repeated aliases or repeated structured fields yield the same normalized triple, only one theorem rule is retained.

Across sources, we do not perform embedding-level or manually judged semantic merging. Skill-extracted rules are deduplicated within their canonical DSL-edge space, while ontology-derived rules are deduplicated within their normalized relation-triple space. During final library assembly, rules that map to the same canonical structure are merged, rules that are merely semantically close but differ structurally are retained as separate evidence. This conservative deduplication strategy preserves directionality, role structure, and source traceability.

A.7 Stage-Specific Rule Usage

Stage-effective rule sets are usage-specific subsets or expansions of the theorem library, rather than a disjoint partition of all source-level rules. Table 17 summarizes which rule sources are used by each verification stage.

The expert-annotated source contains 3,365 rules in total. Of these, 445 are used in Stage 1 and 2,581 are used in Stage 2, the remaining 339 expert-annotated rules are retained in the theorem library but are outside the Stage 1/2 effective subsets.

Field	Used for rules	Usage
LABEL	Yes	Canonical adjuvant/component concept
alternative label	Yes	Alias expansion for component matching
definition	Yes	Stored as <code>evidence_text</code>
definition source	Yes	Stored as <code>paper_id</code> for traceability
target receptor	Yes	Generates component–receptor rules
induces immune profile	Yes	Generates component–profile rules
ID	Indirect	Retained as processed ontology entry ID
Parent, Comment	No theorem rule	Retained only in processed ontology metadata
derives from, equivalent axioms	No	Not promoted into theorem rules
Editor/tracker fields	No	Not used for theorem construction

Table 11: Structured Vaxjo/VO fields used for ontology-derived rule construction.

Rule type	Count	Predicate form	Source fields
Component–profile	1,018	CONTAINS $\Rightarrow$ ENHANCE	LABEL / alternative label + induces immune profile
Component–receptor	622	CONTAINS $\Rightarrow$ ACTIVATE	LABEL / alternative label + target receptor
Receptor–profile derived	44	ACTIVATE $\Rightarrow$ ENHANCE	Structured co-annotation of target receptor and induces immune profile
Total	1,684	–	–

Table 12: Ontology-derived rule types. The receptor–profile rules are derived from structured co-annotation, not from free-text inference.

Formal library	Total rules	Vaxjo/VO rules	Usage
Stage 1 effective	445	0	Construction validation
Stage 2 effective	2,581	0	immune-effect scoring
Stage 1/2 runtime	3,365	0	Expert-annotated construction and effect rules
Stage 3 effective	5,698	1,684	Expanded mechanism/effect reasoning

Table 13: Stage-specific usage of ontology-derived Vaxjo/VO rules.

Quality-control item	Result	Notes
Processed ontology records	645	Unique processed VO/Vaxjo ontology entry IDs
Records with target receptor	209	Eligible for component–receptor rules
Records with immune profile	437	Eligible for component–profile rules
Final ontology-derived rules	1,684	Converted structured-field Horn rules
Unique rule IDs	1,684	No duplicate final rule IDs
Single-premise rules	1,684 / 1,684	Exactly one condition per rule
Co-condition rules	0 / 1,684	No co-conditions in final rules
Unique source IDs	339	Deduplicated <code>paper_id</code> values in final rules, not rule counts

Table 14: Quality-control statistics for ontology-derived rules.

Extraction library	Atomic edges	Unique rules	Removed	Deduplication key
General mechanism	1,222	1,169	53	Canonical adjacent mechanism edge
Adjuvant bridge	277	264	13	Canonical adjacent bridge edge
Total	1,499	1,433	66	Exact canonical DSL edge

Table 15: Deduplication statistics for skill-extracted rules. Deduplication is exact over canonical DSL edges, not based on embedding similarity.

Ontology-derived rule type	# Rules	Deduplication key	Notes
Component–profile	1,018	(component_profile, c , u)	Component or alias mapped to immune profile
Component–receptor	622	(component_receptor, c , r)	Component or alias mapped to receptor
Receptor–profile derived	44	(receptor_profile, r , u)	Derived from structured co-annotation
Total	1,684	Normalized relation triple	339 unique source IDs retained for traceability

Table 16: Deduplication structure for ontology-derived rules. Source IDs are retained for traceability but are not used as deduplication keys.

Rule source	# Rules	Usage
Stage 1: Construction validation		
Expert-annotated construction rules	445	Conclusion-side support for construction predicates
Stage 2: immune-effect scoring		
Expert-annotated immune-effect rules	2,581	Conclusion-side support for immune-effect predicates
Stage 3: Mechanistic proof		
Expert-annotated immune-effect rules	2,581	Effect-side and mechanism-related expert evidence
Skill-extracted rules	1,433	Mechanism and bridge transitions extracted from literature evidence
Ontology-derived rules	1,684	Structured VO/Vaxjo-derived for mechanism/effect reasoning
Stage 3 effective total	5,698	Expanded rule set for mechanistic proof

Table 17: Stage-specific theorem-library usage. Stage 1 and Stage 2 use expert-annotated subsets, while Stage 3 expands coverage by combining expert-annotated, skill-extracted, and ontology-derived rules.

B Evaluation Data Construction

B.1 Overview

This appendix details the construction and quality control of the evaluation data used in our experiments. We use three evaluation datasets, conduct an additional stability evaluation, and perform a human audit of ontology-constrained DSL rendering. The closed-world verification evaluates whether the verifier can distinguish theorem-supported claims from controlled perturbations under a fixed theorem library. The chain-ablation extension is a separate diagnostic set used only for the minimal sub-theorem ablation. The expert-aligned set evaluates agreement with human and LLM-based judgments on open-ended design responses. The stability evaluation examines whether expert alignment remains consistent under changes in design questions and generators. The human audit evaluates whether the structured DSL outputs faithfully preserve the corresponding natural-language design responses.

The following sections describe how closed-world positives and negatives are constructed, how the chain-ablation extension is generated, how the expert-aligned set and stability-evaluation responses are constructed and annotated, how ontology-constrained DSL rendering is evaluated through human audit, and how quality-

control checks are applied to ensure label validity, parseability, and the absence of positive-negative conflicts.

B.2 Human Validation of DSL Rendering

Faithful ontology-constrained DSL rendering is a prerequisite for downstream verification because SAVANT verifies the structured DSL outputs rather than the free-form natural-language design responses. We therefore conduct a manual audit of the DSL outputs for 25 of the 51 GPT-4o-generated responses in the expert-aligned set. The adjudicating lead of the domain-expert team performed the audit.

Rather than assigning a single correctness label to each complete response, the audit evaluates atomic rendering units. These units consist of construction field-value pairs, terminal immune-effect claims, and directed mechanism edges. Each unit is assigned one of three fidelity labels: *consistent*, indicating that it faithfully matches the natural-language response, *conceptually consistent but could be more specific*, indicating that it preserves the intended concept but loses or generalizes some detail, or *inconsistent*, indicating that it introduces a clearly unsupported or contradictory interpretation.

As shown in Table 18, 1,089 of the 1,119 audited units (97.32%) faithfully match the source responses. A further 18 units (1.61%) are conceptually consistent but could be more specific. Together, these categories account for 1,107 units (98.93%) that are acceptable at the conceptual level. Only 12 units (1.07%) are inconsistent, and all 113 terminal immune-effect claims are fully consistent.

Most units that are not fully consistent occur in mechanism rendering, particularly when parallel relations are serialized as a causal sequence or an intermediate mechanistic step is omitted. The construction inconsistencies mainly involve preparation information that is not explicitly stated in the source response. Within the audited subset, these results indicate that the downstream verifier generally operates on semantically faithful structured inputs, while also identifying residual rendering errors that should not be ignored.

B.3 Closed-World Verification Construction

The closed-world verification is constructed from the fixed theorem library and contains 4,413 binary-labeled instances. Positive instances are de-

Unit type	Total	Consistent	Conceptually consistent	Inconsistent
Construction fields	764	752	5	7
Immune-effect claims	113	113	0	0
Mechanism edges	242	224	13	5
Total	1,119	1,089	18	12

Table 18: Human validation of DSL rendering.

rived from theorem-supported construction claims, immune-effect claims, atomic mechanism transitions, and true synonym variants. Negative instances are generated by controlled perturbations of theorem-derived claims. This setting evaluates whether the verifier can distinguish supported proof obligations from corrupted variants within the available evidence base.

The negative categories target different verifier failures. In particular, `n_invalid_value` tests rejection of out-of-distribution values, whereas `n_semantic` tests harder in-domain non-edges whose values are type-compatible but unsupported. The synonym positives are DSL/functor-level variants rather than natural-language paraphrases.

B.4 Chain-Ablation Extension

The chain-ablation extension is a separate diagnostic set used only for the minimal sub-theorem ablation. It contains 600 two-hop mechanism chains, with 300 positives and 300 negatives. We keep this extension separate from the closed-world verification because it is designed to test a specific failure mode: whether the verifier can detect a local mechanistic error inside an otherwise plausible chain.

The negative chains are not randomly generated invalid chains. Each negative consists of a valid first edge followed by an invalid second edge, where the corrupted value is sampled from a compatible value pool but does not form a supported Stage 3 edge. This construction directly tests whether a minimal sub-theorem proof can identify local failures instead of assigning support to the chain as a whole.

B.5 Expert-Aligned Evaluation and Stability Evaluation Set

The expert-aligned set contains 51 question-defined design contexts derived from expert-curated QA annotations over 51 papers. For each context, GPT-4o generated a natural-language adjuvant design response. Human experts scored this generated response, not the original paper-derived reference answer. The reference answer

was used only by retrieval baselines, where BM25 and embedding similarity compared the generated response against the corresponding human reference answer. For the stability evaluation, domain experts created 30 additional open-ended design questions, and Qwen3-8B and Kimi-k2.6 each generated one response per question.

Expert Scoring Protocol

Input. Each annotator receives a question-defined adjuvant design context and the corresponding natural-language design response.

Scored object. Annotators score the generated response. The paper-derived human reference answer is not shown as the target to be scored and is used only for retrieval-baseline similarity evaluation.

Scale. Each dimension is scored on a 1–5 scale, where 1 indicates severe inconsistency or unsupported reasoning, and 5 indicates strong consistency with domain knowledge and available evidence.

Stage 1: Construction quality. Annotators check whether the response clearly specifies the adjuvant type, major components, key physicochemical properties, and technically feasible preparation details. They also check whether the response contains implausible materials, hallucinated parameters, or construction details inconsistent with known vaccine-adjuvant practice.

Stage 2: Immune-effect support. Annotators check whether the response states concrete immune outcomes rather than vague immune enhancement. They assess whether the claimed effects are consistent with the design context, adjuvant type, and delivery setting, and whether the response avoids overclaiming unsupported effect strength or numerical outcomes.

Stage 3: Mechanistic reasoning. Annotators check whether the response provides a coherent causal chain from construction choices to intermediate immune mechanisms and terminal immune effects. They penalize unsupported jumps, causal reversals, missing intermediate steps, or mechanisms that contradict established immunological knowledge.

Output. Each annotation contains three numerical scores, one for each stage, together with optional notes.

Five annotators with vaccine-adjuvant expertise participated in the human evaluation as compensated expert evaluators, including two PhD-level and three master’s-level annotators. They were compensated for their time spent scoring the generated responses. Before formal scoring, all annotators received the same rubric and calibration instructions. We conducted a pre-annotation calibration exercise with redundant scoring on 10

Type	N	Label	Construction procedure
p_property	270	Pos.	Property bundles from Stage 1 construction or antigen-construction rules, including fields such as type, size, method, and components.
p_stage2	400	Pos.	immune-effect predicates from the conclusion side of Stage 2 rules.
p_mechanism	400	Pos.	Atomic Stage 3 transitions $b_j \Rightarrow p_j$, sampled from causal-chain and immune-effect rules.
p_synonym	770	Pos.	True functor-level synonym variants from the Stage 3 runtime synonym closure, with values unchanged.
n_property	540	Neg.	Stage 1 property bundles with one field value replaced by an invalid anchored value outside the allowed Stage 1 value set.
n_invalid_value	500	Neg.	Stage 3 samples with one condition-side or effect-side value replaced by an out-of-distribution value.
n_stage2	400	Neg.	immune-effect negatives with the effect functor preserved and the target value replaced by an invalid value.
n_semantic	500	Neg.	In-domain hard negatives sampled from the same source type, side, functor, and role value pool, but unsupported in the closed-world graph.
n_numeric	75	Neg.	Single-side numerical perturbations, including unit-scale changes and cross-dimensional replacements.
n_antonym	100	Neg.	Polarity negatives created by replacing the effect functor with an opposite-polarity functor.
n_causal	400	Neg.	Directionality negatives created by reversing a supported edge when the reversed edge is not in the positive closure.
n_pseudo_synonym	58	Neg.	Pseudo-synonym negatives where the effect functor is replaced by a surface-similar but unsupported non-synonym.
Total	4,413	–	1,840 positives and 2,573 negatives.

Table 19: Closed-world verification construction. Positive instances are theorem-supported claims or true synonym variants, negative instances are controlled perturbations.

Type	N	Label	Construction procedure
p_chain_all_edges	300	Pos.	Two-hop Stage 3 mechanism chains where both atomic edges are supported by the theorem library.
n_chain_one_bad_edge	300	Neg.	One-bad-edge chains where the first edge is supported and the second edge is corrupted by replacing the effect-side value with a typed in-domain non-edge.

Table 20: Chain-ablation extension used only for the minimal sub-theorem ablation.

held-out examples to assess whether the rubric supports consistent expert judgment. As shown in Table 21, calibration agreement was strong for Stage 1, moderate for Stage 2, and moderate-to-substantial for Stage 3. Although exact agreement was lower for the more judgment-intensive immune-effect and mechanistic-reasoning stages, within-one agreement remained very high across all stages, indicating that most disagreements were minor adjacent-score differences rather than large rating conflicts.

After calibration, the 51 expert-aligned and 60 stability evaluation samples were divided approximately evenly across annotators, and each sample was scored by one assigned expert. Among the five annotators, one senior annotator served as an adjudicating lead for protocol clarification and borderline scoring questions. This assignment-based protocol reflects a realistic expert-review setting,

where domain experts may differ in emphasis when assessing formulation plausibility, immune-effect support, and mechanistic completeness.

The expert scores were used only as an external alignment reference. They were not used to train SAVANT, tune verifier parameters, select model outputs, or serve as an absolute source of ground truth. Because the expert-aligned and stability-evaluation samples were intentionally assigned rather than redundantly annotated, we report the redundant pre-annotation calibration agreement instead of inter-annotator agreement on the final assigned set. LLM-as-Judge baselines use the same scoring dimensions as the human protocol. The full LLM-as-Judge prompt and score parsing procedure are provided in Appendix E.2.

Stage	Ordinal α	Interval α	Mean pairwise ρ	Exact	Within-1
Stage 1: Construction quality	0.874	0.870	0.917	0.922	1.000
Stage 2: Immune-effect support	0.586	0.541	0.709	0.578	0.984
Stage 3: Mechanistic reasoning	0.688	0.623	0.722	0.563	1.000

Table 21: Pre-annotation expert calibration results on 10 held-out examples. We report ordinal and interval Krippendorff’s α , mean pairwise Spearman’s ρ , exact agreement, and within-one agreement on the 1–5 scoring scale. Within-one agreement counts score pairs that differ by at most one point.

B.6 Data Quality Control

We apply quality-control checks to the closed-world verification and the chain-ablation extension before evaluation. For the closed-world verification, all samples are binary-labeled, have unique sample identifiers, and contain evaluable DSL or property fields. Table 22 summarizes the main checks.

Deduplication during verification construction is performed at the candidate level using the sample type, source type, and DSL or property representation. This prevents exactly repeated candidates within the same construction process. We do not, however, enforce a strict global canonical deduplication over all samples, because the same terminal effect or property bundle may be supported by multiple source rules. When grouping only by stage and DSL/properties, the final verification contains 66 duplicate evaluation keys. All of them are positive duplicates, and none creates a positive–negative label conflict.

Negative construction also checks that a perturbed instance does not fall back into the corresponding positive closure. Stage 1 negatives are checked against the positive property set, Stage 2 negatives against the terminal-effect positive set, and Stage 3 negatives against the positive DSL-edge set. Synonym positives use the true synonym closure, and negative samples are not allowed to fall back into that closure.

The chain-ablation extension is validated separately because it is used only for the minimal sub-theorem ablation. Table 24 shows that the extension is balanced, parseable, and free of duplicate evaluation keys or label conflicts.

Check	Result	Notes
Total samples	4,413	Closed-world verification
Binary labels	Yes	All labels are 0/1
Positive / negative counts	1,840 / 2,573	Matches the verification composition
Missing labels	0	No unlabeled samples
Non-binary labels	0	No invalid label values
Duplicate sample IDs	0	No repeated sample identifiers
Positive–negative conflicts	0	No canonical key has conflicting labels
Property parse failures	0	All Stage 1 property samples are parseable
DSL parse failures	0	All DSL-based samples are parseable
Empty evaluation samples	0	All samples contain evaluable DSL or properties

Table 22: Quality-control checks for the closed-world verification.

Duplicate source	Count	Interpretation
p_stage2 terminal-effect duplicates	65	Source-distinct positives sharing the same canonical terminal claim
p_property property-bundle duplicates	1	Source-distinct positives sharing the same canonical property bundle
Positive–negative conflicts	0	No duplicate key has conflicting labels

Table 23: Source-distinct duplicate evaluation keys retained in the closed-world verification.

Check	Result	Notes
Total samples	600	Chain-ablation extension
Pos. / neg.	300 / 300	Balanced diagnostic set
Duplicate IDs	0	No repeated sample IDs
Duplicate eval. keys	0	No repeated chain keys
Label conflicts	0	No pos.–neg. conflicts
DSL parse failures	0	All chains parseable

Table 24: Quality-control checks for the chain-ablation extension.

C Scoring and Aggregation Details

C.1 Notation and Operators

This section describes the scoring rules used in SAVANT. The weights and thresholds are verifier configuration settings rather than parameters tuned to expert scores. We use symbolic coefficients in the equations and report their values in Table 25.

Let $\tau = b_\tau \Rightarrow h_\tau$ denote a proof obligation, where b_τ is an optional premise predicate and h_τ is the target predicate. Let $r = b_r \Rightarrow p_r$ denote a theorem-library rule, where b_r is the rule premise and p_r is the rule conclusion. We use $\mathcal{T}_1$, $\mathcal{T}_2$, and $\mathcal{T}_3$ to denote the rule sets used by Stage 1, Stage 2, and Stage 3, respectively.

For a set of values X , $\text{Mean}(X)$ denotes the arithmetic mean. The operator $\text{clip}(x, 1, 5)$ truncates x to the interval $[1, 5]$. The indicator function $\mathbf{1}[\cdot]$ returns 1 when the condition is true and 0 otherwise.

C.2 Predicate-Level Matching

For two predicates p and q , SAVANT computes a predicate-level match score as

$$\text{PredMatch}(p, q) = s_{\text{type}}(p, q) \cdot s_{\text{value}}(p, q). \quad (32)$$

The type score s_{type} compares predicate functors using exact matching, case-insensitive matching, proxy or synonym mapping, and semantic proxy fallback.

The value score s_{value} measures semantic similarity between normalized argument values. When role information is available, values are compared under aligned roles, otherwise, SAVANT falls back to flat value matching. Exact normalized matches receive full support. Non-numerical values use biomedical semantic matching, while physical quantities use the numerical-aware scoring function below. Pseudo-synonymous values may be penalized to avoid treating surface similarity as equivalence.

C.3 Numerical-Aware Matching

For numerical values, SAVANT normalizes units and compares values on a log scale. Let x_q and x_r be the normalized numerical values, and let d_q and d_r be their physical dimensions. The numerical score is

$$s_{\text{num}} = \begin{cases} \exp(-\kappa |\log_{10} x_q - \log_{10} x_r|), & d_q = d_r, \\ \epsilon_{\text{dim}}, & d_q \neq d_r. \end{cases} \quad (33)$$

Thus, unit-scale differences receive a graded penalty, while cross-dimensional mismatches receive a small fallback score. Supported unit categories include size or length, mass or dose-like amount, time, percentage or ratio, concentration or molarity, voltage-related values, temperature, volume, and pH.

C.4 Stage 1 Construction Scoring

Stage 1 evaluates construction support. Let S_{type} , S_{method} , and S_{size} denote support scores for adjuvant type, preparation method, and size. The core construction score is

$$C_{\text{core}} = \alpha_{\text{type}} S_{\text{type}} + \alpha_{\text{method}} S_{\text{method}} + \alpha_{\text{size}} S_{\text{size}}. \quad (34)$$

Let S_{comp} denote optional component support and S_{antigen} denote antigen-related support. The proof-support score is

$$C_{\text{proof}} = \lambda_{\text{core}} C_{\text{core}} + \lambda_{\text{comp}} S_{\text{comp}} + \lambda_{\text{antigen}} S_{\text{antigen}}. \quad (35)$$

Stage 1 also uses theorem support S_{thm} and ontology-prior support S_{prior} :

$$C_{\text{legacy}} = \lambda_{\text{thm}} S_{\text{thm}} + \lambda_{\text{prior}} S_{\text{prior}}. \quad (36)$$

The final raw Stage 1 score is obtained by guarded aggregation:

$$R_1 = G_{\text{guard}}(C_{\text{legacy}}, C_{\text{proof}}, \Pi_{\mathcal{O}}), \quad (37)$$

where $\Pi_{\mathcal{O}}$ denotes ontology-prior checks. The function G_{guard} applies conservative caps when core fields are missing, size is absent, multiple core fields have only weak support, or ontology priors detect mild, moderate, or severe construction issues.

C.5 Stage 2 Immune-Effect Scoring

Stage 2 evaluates claimed terminal immune-effect predicates without proving the full causal path. Let e_j be the j -th terminal effect predicate. For a rule $r \in \mathcal{T}_2$, define its effect-side match as

$$g_2(e_j, r) = \eta_{\text{type}} s_{\text{type}}(e_j, p_r) + \eta_{\text{value}} s_{\text{value}}(e_j, p_r). \quad (38)$$

The support for e_j is the best matching rule conclusion:

$$G_{\text{eff}}(e_j) = \max_{r \in \mathcal{T}_2} g_2(e_j, r). \quad (39)$$

If the value match is very low, SAVANT applies an additional decay so that functor agreement alone cannot produce high support.

For m terminal effects, the raw concept support

$$R_{2,\text{concept}} = \frac{1}{m} \sum_{j=1}^m G_{\text{eff}}(e_j). \quad (40)$$

Stage 2 also uses a specificity calibration. Let F_{terminal} denote the fraction of terminal claims with specific immune-effect functors, and let D_{token} denote token-level diversity of terminal-effect descriptions. The specificity-adjusted score is

$$R_{2,\text{spec}} = \mu_{\text{raw}} R_{2,\text{concept}} + \mu_{\text{functor}} F_{\text{terminal}} + \mu_{\text{div}} D_{\text{token}}. \quad (41)$$

The Stage 2 discrete score is then

$$D_2 = \text{clip}(b_{\text{eff}} + \lambda_{\text{eff}}(R_{2,\text{spec}} - \tau_{\text{eff}}), 1, 5). \quad (42)$$

C.6 Stage 3 Mechanistic Chain Scoring

Stage 3 verifies mechanism chains through local transition support and robust coverage aggregation. For an edge $p_{t-1}^{(k)} \Rightarrow p_t^{(k)}$, let s_b be the premise-side predicate match and s_h be the conclusion-side predicate match. For a transition query matched to a rule, EdgeMatch is implemented as

$$\text{EdgeMatch}(s_b, s_h, \hat{\tau}, \hat{r}) = (\xi_{\text{cond}} s_b + \xi_{\text{eff}} s_h) d_{\text{functor}}, \quad (43)$$

where d_{functor} is a decay factor determined by functor compatibility, using the same exact, case-insensitive, proxy, and semantic-fallback hierarchy used for s_{type} . We denote the resulting edge score as

$$s_t^{(k)} = \text{EdgeMatch}(s_b, s_h, \hat{\tau}, \hat{r}). \quad (44)$$

For each chain c_k , the chain score is the mean of its edge scores:

$$\text{score}(c_k) = \text{Mean}_t s_t^{(k)}. \quad (45)$$

Multiple chains are averaged equally:

$$\bar{s}_{\text{chain}} = \frac{1}{K} \sum_{k=1}^K \text{score}(c_k). \quad (46)$$

Let n_{sup} be the number of supported edges, n_{weak} the number of weakly supported edges, and n_{all} the total number of edges. Supported edges satisfy $s_t^{(k)} \geq \theta_{\text{sup}}$, while weak edges satisfy $0 < s_t^{(k)} < \theta_{\text{sup}}$. The partial edge coverage is

$$C_{\text{edge}} = \frac{n_{\text{sup}} + \omega_{\text{weak}} n_{\text{weak}}}{n_{\text{all}}}. \quad (47)$$

The final Stage 3 raw score is

$$R_3 = \beta_{\text{chain}} \bar{s}_{\text{chain}} + \beta_{\text{cov}} C_{\text{edge}}. \quad (48)$$

C.7 Discretization and Verdicts

For Stage 1 and Stage 3, raw scores are mapped to the 1–5 scale using threshold-based discretization:

$$D_i = 1 + \sum_{\ell=1}^4 \mathbf{1}[R_i \geq \delta_{\ell}], \quad i \in \{1, 3\}. \quad (49)$$

Stage 2 uses the specificity-calibrated mapping defined above. The report-level score is computed from the three 1–5 stage scores:

$$S_{\text{report}} = \nu_1 D_1 + \nu_2 D_2 + \nu_3 D_3. \quad (50)$$

The report is mapped to four implementation labels: VERIFIED, PARTIAL, HYPOTHESIS, and UNVERIFIED. We use UNVERIFIED to match the implementation label.

C.8 Hyperparameter Configuration

Table 25 lists the SAVANT scoring configuration. These values are verifier settings, not parameters tuned to maximize expert-alignment scores.

D RAG Baseline Configuration

Both RAG baselines use identical retrieval and judging settings and differ only in the indexed corpus. For a controlled comparison with SAVANT, retrieval uses the same BioLORD-2023-M encoder with SentenceTransformers. Embeddings are L2-normalized and indexed using FAISS IndexFlatIP for cosine-similarity retrieval, with max_seq_length=128. Each query is constructed from the design question and the generated natural-language response. We first retrieve 40 candidates, deduplicate them by source page, and retain the top eight passages. The same evidence set is used for all three evaluation stages.

Paper-RAG indexes 20,577 passages obtained by splitting 53 domain papers and two textbooks into 96-word chunks with a 24-word overlap. Evidence-RAG additionally indexes 2,529 deduplicated natural-language evidence records associated with the frozen runtime theorem library, yielding 23,106 retrieval records in total. Each added evidence record is indexed independently without further chunking.

Each judge receives the design question, complete natural-language response, retrieved evidence, and three-stage scoring rubric. Retrieved evidence is placed after the response and before the rubric, with each passage accompanied by its source and page. The judges do not receive the ontology-constrained DSL, Horn rules, proof obligations, verification traces, SAVANT scores, or expert labels. GPT-5 and DeepSeek-v4-Flash use a temperature of 0.3 and a maximum output length of 4,096 tokens, with one run per setting.

Symbol	Value	Description
s_{exact}	1.00	Exact functor or value match
s_{case}	0.95	Case-insensitive functor match
s_{proxy}	0.90	Proxy / synonym functor mapping
τ_{proxy}	0.75	Semantic proxy threshold
κ	2.0	Log-scale numerical penalty
ϵ_{dim}	0.05	Cross-dimension numerical fallback
α_{type}	0.40	Stage 1 type support weight
α_{method}	0.35	Stage 1 method support weight
α_{size}	0.25	Stage 1 size support weight
λ_{core}	0.86	Stage 1 core support weight
λ_{comp}	0.10	Optional component support weight
λ_{antigen}	0.04	Antigen support weight
λ_{thm}	0.60	Theorem-support weight
λ_{prior}	0.40	Prior-support weight
η_{type}	0.30	Stage 2 functor/type weight
η_{value}	0.70	Stage 2 value weight
μ_{raw}	0.70	Raw effect-support weight
μ_{functor}	0.20	Terminal-functor specificity weight
μ_{div}	0.10	Token-diversity weight
b_{eff}	4.0	Stage 2 score offset
λ_{eff}	4.0	Stage 2 calibration slope
τ_{eff}	0.70	Stage 2 specificity center
ξ_{cond}	0.50	Stage 3 condition-side weight
ξ_{eff}	0.50	Stage 3 effect-side weight
θ_{sup}	0.65	Supported-edge threshold
ω_{weak}	0.50	Weak-edge coverage weight
β_{chain}	0.70	Chain-mean weight
β_{cov}	0.30	Edge-coverage weight
$\delta_1, \delta_2, \delta_3, \delta_4$	0.30, 0.50, 0.70, 0.90	Raw-to-1–5 thresholds
ν_1, ν_2, ν_3	0.30, 0.30, 0.40	Report-level stage weights

Table 25: SAVANT scoring hyperparameters.

E LLM Prompts

E.1 Generation Prompts

This section provides the prompt templates used in the two-stage generation process. The first prompt elicits an unconstrained natural-language design rationale from the input design goal. The second prompt uses this rationale as context and renders it into the ontology-constrained DSL used by SAVANT for downstream verification. We include the prompts to make the generation procedure reproducible and to clarify how exploratory language generation is separated from symbolic rendering.

E.1.1 Stage-I Natural-Language Rationale Prompt

Stage-I Prompt: Free-Form Scientific Rationale

Role. You are an expert scientist in immunology.

Goal. Design a novel vaccine adjuvant system based on the user's request.

Instruction. Provide a detailed scientific rationale explaining the design principles, mechanisms of action, delivery strategy, and expected immune outcomes. Structure the response around three aspects:

1. Adjuvant construction and physical/chemical properties, including type, size, formulation, and preparation method.
2. Expected immune effects and outcomes, including antibody responses, T cell responses, cytokine profiles, and protection.
3. Step-by-step causal mechanism from adjuvant design to immune response.

Write in natural scientific language. Do not output JSON or structured formats yet. If the design relies on specific physical properties, such as size, charge, or molecular weight, explicitly state their exact values in the rationale.

E.1.2 Stage-II Ontology-Constrained DSL Rendering Prompt

Stage-II Prompt: Constrained Symbolic Rendering

Role. You are a strictly constrained data extraction assistant.

Task. Convert the provided scientific rationale into a strict structured vaccine-adjuvant design object.

Input.

Scientific Rationale: {rationale_text}

Extract the structured vaccine design from above.

Zero-hallucination rule. Extract only entities, properties, and claims explicitly stated in the scientific rationale. Do not invent components, values, pathways, doses, sizes, or mechanisms. If a parameter is absent, leave it as null or omit it.

Required design structure.

- The vaccine antigen and adjuvant must each contain a `components` array.

- Each component must include a `component_type`, selected from the controlled component types.
- The adjuvant must include a controlled type, such as `Nanoparticle`, `Emulsion`, `Molecule`, `Inorganic_salt`, `Hydrogel`, or `Microneedling`.
- Vaccine, antigen, and adjuvant objects must include a `preparation` object with exactly one controlled preparation method.

Controlled preparation methods. The preparation method must be one of `mix`, `assemble`, `conjugate`, `emulsify`, `encapsulate`, `crosslink`, or `new_prepare`. Each method has its own required `composition_type` and optional method-specific fields.

Mandatory FSM-masked outputs. Generate `claim_statements` and `mechanism_chains`. These are the authoritative structured outputs. Legacy free-form DSL strings are serialized automatically and should not be generated directly.

Claim statements. Each claim statement must contain:

- **functor:** one controlled claim functor from the schema.
- **outcome:** a compact outcome phrase, not a sentence.

Do not include treatment groups, controls, magnitudes, doses, comparators, or evidence fields in claim statements. Generate two to five high-confidence claims and prefer fewer precise claims over generic or redundant ones.

Mechanism chains. Each mechanism chain must contain ordered nodes. Each node must include:

- **functor:** one controlled mechanism functor from the schema.
- **value:** a short canonical event phrase.

Adjacent nodes must form cause-to-effect relationships. A chain with only one node is invalid. Generate three to five non-overlapping chains, with two to five nodes per chain. Prefer compact biological mechanism nodes and place explanations in the chain rationale.

Mechanism constraints. Mechanism chains should describe biological causal mechanisms from adjuvant properties or first immune events to downstream immune effects. Chains should not end with upstream bookkeeping nodes such as `Size`, `Type`, `Method`, `Route`, `Contains`, `Release`, or `Target` unless no downstream immune event is stated. Chains should continue from internal receptor or pathway activation to biological endpoints when supported.

Controlled functor families. Use controlled functors for formulation entry, immune events, and endpoints. Examples include `Size`, `Type`, `Contains`, `Target`, `Release`, `Bind`, `Uptake`, `Present`, `Activate`, `Mature`, `Prime`, `Polarize`, `Secrete`, `Enhance`, `Increase`, `Reduce`, `Protect`, and `Kill`. Use `Size` for size, `Type` for adjuvant type, and `Contains` for composition.

Field placement. Place standard physicochemical properties in dedicated top-level fields when possible, including `type`, `size`, `shape`, `zeta_potential`, `dose`, and `target_disease`. Complex attributes may be placed in `properties`.

Component extraction. List all explicitly mentioned constituent materials in the components array. Component types include Protein, Antibody, NucleicAcid, Lipid, InorganicSalt, Polymer, SmallMolecule, and General. Do not omit components if materials are mentioned in the rationale.

Consistency constraint. If a mechanism chain starts from a design or property node, such as Size, Shape, Type, Method, Route, Contains, Form, Encapsulate, or Release, that value must be explicitly supported by a populated field or statement in the structured design.

Rationale field. Each rationale item must be a complete, self-contained point extracted from Stage I. Each item must stay within the word limit and end with proper punctuation.

E.2 LLM-as-Judge Evaluation Prompt

We use the following prompt for LLM-as-Judge baselines. The prompt asks each judge model to evaluate the same generated adjuvant design response along the three stage-aligned dimensions used in expert scoring. We only use the integer 1–5 scores in our evaluation.

LLM-as-Judge Prompt

Role. You are a senior immunologist evaluating AI-generated vaccine adjuvant design proposals.

Task. Assess each proposal across three stages that mirror the SAVANT verification pipeline:

1. Stage 1 — Adjuvant Construction.
2. Stage 2 — Immune Effect Claims.
3. Stage 3 — Causal Chain Integrity.

Evaluate solely based on scientific reliability. Do not evaluate writing quality, fluency, or rhetorical style.

Design Question.

{question}

Proposed Adjuvant Design.

{free_text_output}

Scoring instruction. Each stage should be scored from 1 to 5. For each stage, provide a brief reasoning in 2–3 sentences and an integer score from 1 to 5.

Stage 1 — Adjuvant Construction.

Evaluate whether the described adjuvant properties are physically and chemically reasonable, technically feasible, and consistent with published evidence or established domain knowledge.

Consider adjuvant type, particle size or dimensions, formulation strategy, preparation method, carrier material, antigen loading or antigen-binding strategy, physicochemical properties such as charge, molecular weight, component ratio, or release profile, and whether exact values are provided when the mechanism depends on specific physicochemical properties.

Scoring criteria.

- **5:** Key construction parameters are highly consistent with published evidence or established domain knowledge, and the design is technically feasible.

- **4:** Major construction parameters are reasonable, minor details are vague but not clearly wrong.
- **3:** The overall construction direction is plausible, but one or two parameters are inaccurate, unclear, or insufficiently supported.
- **2:** The construction contains notable problems or inconsistencies with domain knowledge or literature experience.
- **1:** The construction is seriously flawed or contains hallucinated parameters.

Stage 2 — Immune Effect Claims.

Evaluate whether the claimed immune outcomes are scientifically credible.

Consider antibody titers or humoral immune responses, T-cell responses, cytokine profiles, DC maturation, antigen presentation, Th1/Th2 polarization, CD8+ T-cell responses, protective immunity, comparisons with control groups if mentioned, and whether the claimed immune effects align with the design question and the proposed adjuvant system.

Scoring criteria.

- **5:** Major immune effect claims are highly consistent with published evidence or established immunological knowledge.
- **4:** Major effects are credible, minor claims are reasonable inferences.
- **3:** Some effects are credible, while others remain unverified hypotheses.
- **2:** Most claims lack support or contain implausible values, magnitudes, or extrapolations.
- **1:** The claims contradict known immunology or appear hallucinated.

Stage 3 — Causal Chain Integrity.

Evaluate whether the stepwise mechanistic chain from adjuvant structure to immune outcome is valid.

Consider whether the proposal forms a clear causal chain from adjuvant construction and physicochemical properties to intermediate immune mechanisms and terminal immune outcomes, whether each step is mechanistically justified, whether the causal direction is correct, whether there are major logical gaps, whether the answer over-infers directly from material properties to immune outcomes, whether the answer merely stacks technical terms without valid causal relations, and whether the mechanism contradicts known biology.

Scoring criteria.

- **5:** Every causal step is mechanistically justified, logically coherent, and consistent with known biology.
- **4:** The overall chain is correct, but some steps are loosely connected.
- **3:** The starting point and endpoint are plausible, but intermediate steps are unclear, incomplete, or imprecise.
- **2:** The chain contains significant logical gaps or unjustified inferential leaps.

- **1:** The causal reasoning is incoherent or contradicts known biological mechanisms.

Output format. Output strictly in JSON format. Do not output Markdown, explanations outside JSON, or any additional text. Use the following JSON schema:

```
{
  "stage1_construction": {
    "reasoning": "<2-3 sentences>",
    "score_1to5": 4
  },
  "stage2_effects": {
    "reasoning": "<2-3 sentences>",
    "score_1to5": 4
  },
  "stage3_causal_chain": {
    "reasoning": "<2-3 sentences>",
    "score_1to5": 3
  }
}
```

Important. Each score_1to5 must be an integer from 1 to 5.

F Ontology Schema and FSM Constraints

F.1 Overview

SAVANT uses an ontology-constrained schema to render free-form scientific rationales into structured design objects. The schema defines allowed adjuvant types, component types, preparation methods, immune-effect claims, and mechanism-chain functors. It is implemented with strict schema validation, so fields outside the selected branch are not permitted. This supports the finite-state generation procedure described in Section 3.3. At each generation step, the controller masks actions that are not valid under the current schema state.

F.2 Implementation Details

The ontology schema is implemented using Pydantic. All schema classes inherit from a strict base model with `extra='forbid'`, which rejects fields outside the selected schema branch. This provides the implementation basis for branch exclusivity: once an adjuvant type, preparation method, claim functor, or mechanism functor is selected, incompatible fields are not valid outputs.

For locally deployed models, SAVANT uses Outlines to enforce schema-constrained generation. The closed vocabularies for claim functors and mechanism functors are exposed as schema enums, allowing invalid functor tokens to be masked during generation. For API-based models, SAVANT uses Instructor to obtain schema-constrained outputs under the same Pydantic schema. Thus, local and API backends share the same structural constraints, even though the constrained decoding interface differs.

F.3 Top-Level Design Schema

The generated design object contains four main components: a vaccine object, a disease-model object, structured immune-effect claims, and structured mechanism chains. The vaccine object contains an antigen and an adjuvant, each with component and preparation information. The claim and mechanism fields are generated in structured form and then serialized into legacy DSL strings for downstream compatibility.

F.4 Adjuvant and Preparation Branches

Adjuvant generation uses mutually exclusive schema branches for nanoparticle, microparticle,

emulsion, molecule, inorganic salt, hydrogel, and microneedling adjuvant types. Once an adjuvant type is selected, fields from incompatible branches are rejected by the strict schema. Preparation methods are also branch-structured, with controlled options including `mix`, `assemble`, `conjugate`, `emulsify`, `encapsulate`, `crosslink`, and `new_prepare`.

F.5 Component Type Schema

Components are represented through a discriminated union over component types. Each component has a name and a required `component_type`. Specialized schemas are used for proteins, antibodies, nucleic acids, small molecules, photosensitizers, lipids, inorganic salts, polymers, and general components.

F.6 Claim and Mechanism Functors

Stage 2 claim statements use a controlled claim functor and an open compact outcome phrase. The outcome field is open because immune-effect targets can vary across diseases and experimental settings, while the functor field is closed to enable schema masking.

Mechanism chains use ordered nodes of the form `Functor(value)`. The functor is selected from a closed vocabulary, while the value remains an open compact scientific phrase. Each chain must contain at least two nodes and at most five nodes.

F.7 Verifier-Side Predicate Schema Registry

In addition to the generation-time schema, SAVANT maintains a verifier-side predicate schema registry. This registry is automatically generated from predicate rules and is used for parsing Horn predicates, normalizing functor names, and supporting symmetric instantiation. The registry is implementation-facing, its internal argument slots should not be interpreted as biological role labels.

The verifier-side registry is distinct from the generation-time mechanism-node schema. Generation uses compact `Functor(value)` nodes for constrained DSL rendering, whereas verification expands predicates into canonical Horn-rule objects for matching. This separation allows the generator to remain compact while the verifier performs structured comparison over normalized predicates.

Field	Required structure	Purpose
vaccine	Vaccine schema with name, target_disease, antigen, adjuvant, and preparation	Stores the full vaccine-adjuvant design object
disease_model	Disease model schema with name, host_animal, optional model type, route, and rationale	Records the biological context of the design
claim_statements	List of structured Stage 2 claim objects, each with functor and outcome	Represents terminal immune-effect claims
mechanism_chains	List of ordered mechanism chains, each containing controlled-functor nodes and an optional rationale	Represents causal mechanism chains
rationale	List of concise natural-language rationale items	Stores extracted scientific reasoning from the Stage-I rationale

Table 26: Top-level structured design schema used by SAVANT.

Functor group	Claim functors
High-level effect claims	Enhance, Increase, Reduce, Inhibit, Suppress, Protect, Avoid, Control, Balance, Equal, Positive, Establish, Tune, Sustain, Prolong
Event-level claims	Activate, Prime, Differentiate, Polarize, Express, Secrete, Present, Mature, Recruit, Induce, Neutralize, Kill, Disrupt, Irradiate, Route, Target, Release, Uptake, Migrate

Table 27: Controlled functors for Stage 2 claim statements.

F.8 FSM Constraints

The schema enforces finite-state constraints during DSL rendering. First, strict models reject extra fields outside the selected schema branch. Second, the adjuvant and preparation schemas are branch-structured, so selecting one type masks incompatible fields from other branches. Third, claim statements and mechanism chains must use controlled functors. Fourth, mechanism chains must contain at least two causal nodes. These constraints implement the grammar-level logic mask used by SAVANT.

For compatibility with downstream verification code, structured claims and mechanism chains are serialized into legacy DSL strings. Claim statements are serialized as `Claim = Functor(outcome)`, and mechanism chains are serialized by joining ordered nodes with the `>>` operator.

Functor group	Mechanism functors
Design / formulation entry	Size, Shape, Type, Method, Route, Target, Contains, Form, Display, Mix, Conjugate, Assemble, Encapsulate, Crosslink, New_Prepare, Release, Sustain
Immune mechanism events	Bind, Uptake, Process, Present, Activate, Mature, Migrate, Recruit, Prime, Differentiate, Polarize, Express, Secrete, Trigger, Enable, Induce
Effects / endpoints / constraints	Enhance, Increase, Reduce, Inhibit, Suppress, Protect, Avoid, Control, Balance, Equal, Positive, Establish, Tune, Neutralize, Kill, Disrupt, Irradiate, Prolong

Table 28: Controlled functors for mechanism-chain nodes.

Functor	Predicate family	Frequency in registry
Enhance	Immune-effect or endpoint support	1,625
Method	Construction or preparation method	1,242
Contains	Component or composition relation	835
Reduce	Negative or suppressive effect relation	469
Type	Adjuvant or object type relation	435
VaccineProperty	Vaccine-level construction property	286
AdjuvantProperty	Adjuvant-level construction property	273
Name	Name or identifier relation	264
AntigenProperty	Antigen-level construction property	172
Equal	Equivalence or balance relation	148
Increase	Increased immune or biological outcome	141
Activate	Receptor, pathway, or cell activation	101
Route	Delivery or administration route	62
Size	Size or scale relation	51
Target	Targeting relation	26

Table 29: Representative verifier-side predicate functors. Frequencies are computed from the predicate-rule registry.

G Full Closed-World Verification Results

This section reports the complete closed-world verification results. The verification contains 4,413 binary-labeled instances, including 1,840 positives and 2,573 negatives. All instances are used for classification and ranking evaluation. Because this verification evaluates verifier behavior under a fixed theorem library, we report aggregate metrics, threshold behavior, robustness gaps, and per-type diagnostics.

G.1 Metric Definitions

For a threshold τ , a sample is predicted positive if its verifier score is at least τ . We report standard threshold-based metrics, including Accuracy, Precision, Recall, F1, and Specificity. AUROC and PR-AUC are computed over all binary-labeled samples using the continuous verifier score.

In per-type diagnostics, Acc@0.5 , Acc@0.65 , and Acc@0.75 denote within-type accuracy under the corresponding threshold. Acc@vrd denotes accuracy induced by the implementation verdict threshold. VERI, HYP, PART, and UNV denote the percentages of samples assigned to VERIFIED, HYPOTHESIS, PARTIAL, and UNVERIFIED verdicts.

For robustness diagnostics, P-Exact Mean is the mean score of exact positive instances, and P-Synonym Mean is the mean score of synonym-rewritten positive instances. The synonym robustness ratio is

$$\text{SR} = \frac{\text{P-Synonym Mean}}{\text{P-Exact Mean}}. \quad (51)$$

Score gaps compare positive and perturbed subsets:

$$\Delta_g = \bar{s}_{\text{pos}} - \bar{s}_g, \quad (52)$$

where $\bar{s}_{\text{pos}}$ is the relevant positive reference mean and $\bar{s}_g$ is the mean score of perturbation group g . Larger gaps indicate stronger separation between supported claims and the corresponding perturbation type.

G.2 Overall and Threshold Results

Threshold	Acc.	Prec.	Rec.	F1	Spec.
0.50	0.8271	0.7069	1.0000	0.8283	0.7035
0.65	0.9837	0.9628	0.9995	0.9808	0.9724
0.75	0.9925	0.9829	0.9995	0.9911	0.9876

Table 30: Closed-world threshold metrics.

Table 30 reports performance at three operating thresholds. This table shows how the verifier be-

haves under different decision thresholds, rather than only under a single implementation setting.

The lower threshold preserves perfect recall but admits more false positives. Increasing the threshold improves precision and specificity while retaining nearly all positive instances, indicating that positive and negative examples are well separated by the continuous verifier score.

Table 31 reports the implementation-level verdict metrics, confusion matrix, and threshold-independent ranking metrics. This complements the threshold sweep by showing the final operating point used by the verifier together with AUROC and PR-AUC.

Metric	Value
Accuracy	0.9837
Precision	0.9628
Recall	0.9995
F1	0.9808
Specificity	0.9724
TP	1839
FP	71
TN	2502
FN	1
AUROC	0.9979
PR-AUC	0.9944

Table 31: Closed-world verdict-level and ranking metrics over all 4,413 instances.

At the implementation operating point, SAVANT rejects most perturbations while missing only one positive instance. The high AUROC and PR-AUC show that this performance is not an artifact of a single threshold, but reflects strong ranking separation across the full score range.

G.3 Robustness and Score-Gap Diagnostics

Table 32 evaluates robustness to synonym rewriting and different perturbation types. We include this table because aggregate AUROC does not reveal whether the verifier is robust to true functor synonyms, numerical perturbations, semantic hard negatives, or causal-direction errors.

The synonym robustness ratio remains high, indicating that functor-level synonym handling preserves support for valid synonym-rewritten positives. Invalid-value, numeric, semantic, and pseudo-synonym perturbations show large score gaps, suggesting strong rejection of unsupported values and false functor variants. Antonym and causal perturbations are harder, but their gaps remain positive, showing that the verifier still tends

Robustness / gap metric	Value
P-Exact Recall@0.75	0.9991
P-Exact Mean	0.9995
P-Synonym Mean	0.8998
SR	0.9003
Δ Score	0.6237
Δ Stage2	0.4223
Δ InvalidValue	0.8585
Δ Semantic	0.7593
Δ Numeric	0.8244
Δ Antonym	0.4905
Δ Causal	0.5885
Δ PseudoSynonym	0.8623

Table 32: Closed-world robustness and score-gap diagnostics. Larger gap values indicate stronger separation between supported claims and the corresponding perturbation type.

to downgrade polarity and directionality errors.

G.4 Per-Type Diagnostics

Table 33 provides the full per-type breakdown. This table localizes which perturbations are easy or difficult for the verifier. It also shows how different types are mapped into verdict categories, which is important for understanding whether errors are incorrectly verified or instead downgraded to lower-confidence labels.

The positive groups receive consistently high scores, including synonym-rewritten positives. Among negatives, invalid-value and numerical perturbations are strongly rejected, showing the effect of value validation and numerical-aware scoring. In-domain semantic negatives, antonym substitutions, and causal reversals are more difficult because they preserve plausible biomedical vocabulary or local structure. However, these cases are generally downgraded to hypothesis-level or partial-support verdicts rather than being broadly verified.

Type	N	Mean	Acc@0.5	Acc@0.65	Acc@0.75	Acc@vrd	VERI%	HYP%	PART%	UNV%
p_property	270	1.0000	1.0000	1.0000	1.0000	1.0000	100.0	0.0	0.0	0.0
p_stage2	400	0.9996	1.0000	1.0000	1.0000	1.0000	100.0	0.0	0.0	0.0
p_mechanism	400	0.9991	1.0000	0.9975	0.9975	0.9975	99.8	0.2	0.0	0.0
p_synonym	770	0.8998	1.0000	1.0000	1.0000	1.0000	100.0	0.0	0.0	0.0
n_property	540	0.3735	0.7519	1.0000	1.0000	1.0000	0.0	100.0	0.0	0.0
n_stage2	400	0.5772	0.1675	0.9425	0.9825	0.9425	5.8	0.0	94.2	0.0
n_invalid_value	500	0.1410	1.0000	1.0000	1.0000	1.0000	0.0	100.0	0.0	0.0
n_semantic	500	0.2402	0.9200	0.9700	0.9940	0.9700	3.0	97.0	0.0	0.0
n_numeric	75	0.1751	1.0000	1.0000	1.0000	1.0000	0.0	100.0	0.0	0.0
n_antonym	100	0.5090	0.4200	0.8800	0.9600	0.8800	12.0	88.0	0.0	0.0
n_causal	400	0.4110	0.5300	0.9725	0.9800	0.9725	2.8	97.2	0.0	0.0
n_pseudo_synonym	58	0.1371	0.8276	0.8276	0.8276	0.8276	17.2	82.8	0.0	0.0

Table 33: Per-type closed-world diagnostics. Positive types are prefixed with p_, negative types are prefixed with n_. VERI, HYP, PART, and UNV denote VERIFIED, HYPOTHESIS, PARTIAL, and UNVERIFIED.

H Ablation Study Details

This section reports the full ablation diagnostics. The ablation experiments use 5,013 samples in total, consisting of the 4,413 closed-world verification samples and the 600-sample chain-ablation extension. The primary ablation evidence is evaluated on targeted subsets designed to isolate each mechanism.

H.1 Metric Definitions

For each ablation variant, we report ranking, threshold, and score-separation metrics. PosMean and NegMean denote the mean verifier scores of positive and negative samples within a target subset. We define the positive–negative score gap as

$$\text{Gap} = \text{PosMean} - \text{NegMean}. \quad (53)$$

Larger gaps indicate stronger separation between supported and unsupported instances.

For negative-only diagnostics, Acc@0.65 denotes the fraction of negative samples assigned a score below 0.65. Thus, higher Acc@0.65 and lower mean score indicate stronger rejection of negative instances. In Table 34, Ablated--Full denotes the metric value of the ablated variant minus the full SAVANT value.

H.2 Targeted Primary Ablations

Table 34 reports the main targeted ablation results. Each row evaluates a mechanism on the subset where its removal should be most visible. This is more informative than relying only on aggregate performance, because some mechanisms affect specific failure modes rather than the full verification uniformly.

Removing minimal sub-theorem proof substantially weakens ranking on local chain-error cases. This supports the use of local proof obligations rather than treating the whole mechanism chain as a single undifferentiated object. Removing symmetric instantiation has the strongest effect on endpoint, polarity, and directionality subsets, which shows that matching generated claims and theorem rules in a shared predicate structure is necessary for preserving relation type and causal direction. Removing numeric-aware scoring causes numeric negatives to receive much higher support, confirming that physical quantities require unit- and scale-aware comparison.

H.3 Overall Variant Summary

Table 35 reports aggregate performance over the full 5,013-sample ablation pool. This table is useful for showing global behavior, but it should be interpreted together with the targeted diagnostics because some components are designed for specific error modes.

The aggregate table shows that removing symmetric instantiation causes the largest global degradation. Minimal sub-theorem proof and numeric-aware scoring have more localized effects, so their importance is clearer in the chain-local-error and numeric-negative subsets.

H.4 Target-Subset Diagnostics

Table 36 provides a broader view of how each ablation variant behaves across target subsets. The Gap column reports PosMean--NegMean.

The target-subset table shows that the ablations behave as intended. The minimal-sub-theorem ablation primarily affects the chain subset. The numeric ablation does not change most non-numeric subsets, but it substantially affects numeric-negative rejection. The symmetric-instantiation ablation broadly weakens structured matching, especially for Stage 2 endpoints, polarity-sensitive cases, and causal-direction reversals.

H.5 Targeted Negative Diagnostics

Table 37 reports negative-only diagnostics. These results show how much support each ablation variant assigns to different negative types.

The `n_chain_one_bad_edge` subset requires special interpretation. These negative chains intentionally retain one supported edge and corrupt only the second edge, so their absolute scores can remain high under a fixed threshold. For this reason, fixed-threshold negative accuracy is not the main evidence for the minimal-sub-theorem ablation. The primary evidence is the chain-subset AUROC and PosMean–NegMean separation in Table 34.

Mechanism	Ablation set	Metric	Full	Ablated	Ablated–Full
Minimal sub-theorem	Chain local-error	AUROC ↑	0.9881	0.6698	-0.3184
Minimal sub-theorem	Chain local-error	Pos–Neg gap ↑	0.1591	0.0406	-0.1184
Symmetric instantiation	Stage 2 endpoint	AUROC ↑	1.0000	0.7917	-0.2083
Symmetric instantiation	Hard semantic	AUROC ↑	0.9987	0.9662	-0.0324
Symmetric instantiation	Antonym / polarity	AUROC ↑	0.9999	0.6141	-0.3858
Symmetric instantiation	Directionality	AUROC ↑	0.9912	0.4637	-0.5275
Numeric-aware scoring	Numeric negative	NegAcc@0.65 ↑	1.0000	0.2133	-0.7867
Numeric-aware scoring	Numeric negative	Mean score ↓	0.1751	0.8128	+0.6377

Table 34: Targeted primary ablation results. Each ablation is evaluated on a subset designed to stress the corresponding mechanism. Pos–Neg gap is the mean positive–negative score difference. For Mean score ↓, lower is better.

Variant	AUROC	Acc@0.65	PosMean	NegMean
full	0.9921	0.9262	0.9632	0.3866
minimal_off	0.9914	0.9280	0.9396	0.3814
numeric_off	0.9891	0.9144	0.9632	0.4032
symmetric_off	0.7565	0.6858	0.7614	0.5364

Table 35: Overall ablation summary over the 5,013-sample ablation pool.

Variant	Target	N	AUROC	Acc@0.65	PosMean	NegMean	Gap
full	stage2	800	1.0000	0.9713	0.9996	0.5772	0.4224
full	chain	600	0.9881	0.5033	0.9962	0.8371	0.1591
full	semantic	1670	0.9987	0.9904	0.9337	0.2402	0.6936
full	antonym	1270	0.9999	0.9898	0.9337	0.5090	0.4248
full	causal	800	0.9912	0.9850	0.9991	0.4110	0.5881
minimal_off	stage2	800	1.0000	0.9713	0.9996	0.5772	0.4224
minimal_off	chain	600	0.6698	0.5183	0.8283	0.7877	0.0406
minimal_off	semantic	1670	0.9987	0.9904	0.9337	0.2402	0.6936
minimal_off	antonym	1270	0.9999	0.9898	0.9337	0.5090	0.4248
minimal_off	causal	800	0.9912	0.9850	0.9991	0.4110	0.5881
numeric_off	stage2	800	1.0000	0.9713	0.9996	0.5772	0.4224
numeric_off	chain	600	0.9881	0.5033	0.9962	0.8371	0.1591
numeric_off	semantic	1670	0.9987	0.9904	0.9337	0.2402	0.6936
numeric_off	antonym	1270	0.9999	0.9898	0.9337	0.5090	0.4248
numeric_off	causal	800	0.9912	0.9850	0.9991	0.4110	0.5881
symmetric_off	stage2	800	0.7917	0.7600	0.7232	0.6339	0.0894
symmetric_off	chain	600	0.6149	0.5100	0.8141	0.7874	0.0267
symmetric_off	semantic	1670	0.9662	0.8826	0.7604	0.2994	0.4610
symmetric_off	antonym	1270	0.6141	0.8102	0.7604	0.7111	0.0493
symmetric_off	causal	800	0.4637	0.4850	0.7875	0.7980	-0.0105

Table 36: Target-subset ablation diagnostics. Gap denotes PosMean–NegMean.

Variant	Negative type	N	Mean	Acc@0.65
full	n_stage2	400	0.5772	0.9425
full	n_numeric	75	0.1751	1.0000
full	n_semantic	500	0.2402	0.9700
full	n_antonym	100	0.5090	0.8800
full	n_causal	400	0.4110	0.9725
full	n_chain_one_bad_edge	300	0.8371	0.0067
minimal_off	n_stage2	400	0.5772	0.9425
minimal_off	n_numeric	75	0.1751	1.0000
minimal_off	n_semantic	500	0.2402	0.9700
minimal_off	n_antonym	100	0.5090	0.8800
minimal_off	n_causal	400	0.4110	0.9725
minimal_off	n_chain_one_bad_edge	300	0.7877	0.0467
numeric_off	n_stage2	400	0.5772	0.9425
numeric_off	n_numeric	75	0.8128	0.2133
numeric_off	n_semantic	500	0.2402	0.9700
numeric_off	n_antonym	100	0.5090	0.8800
numeric_off	n_causal	400	0.4110	0.9725
numeric_off	n_chain_one_bad_edge	300	0.8371	0.0067
symmetric_off	n_stage2	400	0.6339	0.7475
symmetric_off	n_numeric	75	0.8128	0.2133
symmetric_off	n_semantic	500	0.2994	0.9340
symmetric_off	n_antonym	100	0.7111	0.2200
symmetric_off	n_causal	400	0.7980	0.0700
symmetric_off	n_chain_one_bad_edge	300	0.7874	0.0333

Table 37: Targeted negative diagnostics for ablation variants. Lower mean and higher Acc@0.65 indicate stronger rejection of negative instances.

I Case Study and Verification Trace

I.1 Overview

This section provides representative qualitative examples of verification outputs from SAVANT. The goal is to illustrate how SAVANT processes GPT-4o-generated natural-language design responses, renders them into ontology-constrained DSL objects, and produces evidence-grounded verification traces. The two cases cover both high-support proof obligations and low-support knowledge gaps.

The case studies are not intended as additional quantitative evaluation. Instead, they demonstrate the interpretability of the verification process. For each case, we first show a compressed version of the design question, the generated response, and the corresponding DSL rendering. We then present proof obligations, matched theorem-library evidence, support scores, and verification status. A knowledge gap indicates insufficient support in the current theorem library, rather than evidence that the proposed mechanism is biologically invalid.

I.2 From Natural-Language Response to Structured DSL

We show two representative cases. Case A illustrates a high-support rendering example, while Case B illustrates a partially supported design that leads to lower-support proof obligations in the verification trace.

Case A. Figure 4 shows a representative rendering example. Given a design question: “*Design an adjuvant for classical swine fever virus that can achieve potent humoral immunity with Th1 polarization and strong dendritic cell activation and antigen presentation under a modular SpyCatcher/SpyTag constraint using stable virus-like nanoparticles. What would be the adjuvant formulation, immune response characteristics, and underlying mechanisms?*”, GPT-4o first generates a natural-language scientific rationale. SAVANT then converts the response into an ontology-constrained DSL object containing construction fields, immune-effect claims, and mechanism chains.

Case B. Figure 5 shows case B. Given a design question: “*Design an adjuvant for SARS-CoV-2 that can achieve significant anti-RBD IgG, IgG1, and IgG2c responses, elevated cross-variant RBD*

IgG against multiple variants, and maintain stability under refrigerated conditions. What would be the adjuvant formulation, immune response characteristics, and underlying mechanisms?”. The case B illustrates a partially supported generated design. We use it to show that SAVANT does not only produce a scalar score, but also separates supported structured claims from lower-support assumptions.

This example highlights a different use case from Case A. The generated response contains structured construction fields, terminal effect claims, and multiple mechanistic chains, but not all proof obligations receive equally strong evidence support. The next subsection uses the rendered DSL objects from both cases to show how SAVANT assigns support scores and identifies knowledge gaps.

I.3 Evidence-Grounded Verification Trace

The verification trace uses the ontology-constrained DSL objects shown in the previous subsection as input. SAVANT does not verify the free-form natural-language response directly. Instead, it converts the rendered construction fields, immune-effect claims, and mechanism chains into proof obligations over the shared predicate space. Construction fields and terminal immune-effect claims become predicate-level obligations, while each adjacent pair in a mechanism chain becomes a transition-level obligation.

For each obligation, SAVANT searches the theorem library for the best matching rule, computes a continuous support score, and assigns a status such as supported, partial support, or knowledge gap. The following visualizations show how these local proof decisions are produced for the two cases. This trace-level view is intended to make clear which parts of a generated design are evidence-grounded and which parts remain unsupported under the current theorem library.

The expert scores and comments reported for these cases are based only on the GPT-4o-generated natural-language responses. The experts did not see the verification traces produced by SAVANT. We therefore treat the expert comments as independent qualitative references rather than as supervision for the traces.

The trace below shows how SAVANT evaluates the proposal through soft-symbolic matching against the available theorem library. Low-support edges can reflect either incomplete theorem-library

coverage or coarse mechanism steps in the generated response.

I.3.1 Case A: Mostly Supported Verification Trace

Figures 6 and 7 show the verification for GPT-4o-generated adjuvant design proposal. The proposal describes a nanoparticle formulation of approximately 100–120 nm containing MPLA, R848, and PEI. The expert scores for this case are 4/4/4 for construction quality, immune-effect support, and mechanistic reasoning, respectively.

Stage 1 – Construction. Five of six representative construction obligations are strongly supported. The weaker obligation is *Contains*(PEI), which matches a scaffold-related construction rule with a score of 0.46. This is an approximate construction-level match rather than a direct material match. Guarded aggregation prevents this partial component match from dominating the stage-level score, yielding a Stage 1 score of 4.0.

Stage 2 – Immune Effects. All four immune-effect claims receive support, although several are supported through generic endpoint or ontology-derived rules rather than disease-specific evidence. The specificity calibration module accounts for these broader matches, yielding an average concept score of 0.810 and a Stage 2 score of 4.068. This indicates that the terminal effects are broadly supported, while some claims remain matched at a coarse concept level.

Stage 3 – Mechanism. Four causal chains are constructed. The trace evaluates each chain through local edge-level proof obligations, so the score reflects both the intended causal direction and the specificity of the matched theorem rule.

- **Chain 1** (0.743): Size → DC uptake → TLR activation. The first edge is directionally compatible with particle-size support for cellular uptake. The second edge is more indirect: the matched theorem supports downstream signaling activation rather than TLR7/TLR4 engagement specifically. Thus, the chain preserves a broadly plausible direction from formulation property to innate immune activation, but the evidence is approximate.
- **Chain 2** (0.838): MPLA/R848 → DC maturation → IL-12/IFN- γ . The MPLA component is supported through a receptor-linked

ontology rule for TLR4. The R848 component is not covered by this matched rule, reflecting limited component-specific coverage. The cytokine endpoint is supported at a coarse level: the matched evidence captures an IL-12/IFN- γ immune-activation axis, but it does not prove that dendritic cells directly secrete IFN- γ .

- **Chain 3** (0.823): VLP-polymer complex → MHC I/II presentation → T cell priming. The first edge matches an antigen-presentation-related rule, providing indirect support for the presentation step. The second edge supports the CD4/Tfh side of the priming claim through MHC-II presentation, while the CD8 component is only partially covered by the broader antigen-presentation context.
- **Chain 4** (0.419, Partial): Sustained antigen exposure → antibody enhancement → CSFV neutralization. The first edge receives partial directional support from a sustained-release-to-immune-activation rule, but the matched endpoint is cytokine secretion rather than antibody production. The second edge receives no high-support match, because the current theorem library does not contain a direct rule linking antibody production to CSFV neutralization. This gap may reflect missing CSFV-specific evidence in the theorem library, a coarse generated transition, or both. SAVANT therefore marks this step as a knowledge gap rather than forcing a cross-pathogen generalization.

Summary. Overall, the trace shows that the proposal is directionally plausible and well supported at the construction and terminal-effect levels, while several mechanism matches remain approximate. Chains 1–3 receive partial-to-high support from related theorem-library rules, but the trace also exposes specificity limitations in the current evidence base. Chain 4 identifies a localized evidence gap for the downstream neutralization step. The final verdict (VERIFIED, 4.0204) is consistent with the expert scores of 4/4/4: the design is broadly supported, but not all mechanistic steps are directly grounded at high specificity. This illustrates how SAVANT separates supported proof obligations from lower-support assumptions instead of reducing the generated proposal to a single plausibility score.

I.3.2 Case B: Partial-Support Verification Trace

This case verifies a GPT-4o-generated adjuvant design proposal for SARS-CoV-2, such as Figures 8 and 9. The expert scores for this case are 4/4/3 for construction quality, immune-effect support, and mechanistic reasoning, respectively.

Stage 1 – Construction. The construction stage receives a score of 4.0. The adjuvant type and emulsification method are strongly supported, and the size claim receives moderate-to-high support. Several component matches are approximate rather than exact. For example, `Contains(R848)` is matched to a resiquimod-related construction rule, while `Contains(DOTAP)` is supported by a partial component-level match. The guarded aggregation module combines these construction signals with ontology-prior checks, so the stage-level score remains high while preserving information about weaker component matches.

Stage 2 – Immune Effects. The immune-effect stage receives a score of 4.237. The antibody-related claims, including `Enhance(IgG production)` and `Increase(cross-variant RBD IgG)`, receive high support through endpoint-level matching and ontology-derived evidence. However, these are broad endpoint matches rather than direct evidence for cross-variant protection. This is directionally consistent with the expert comment that the cross-protection claim may require further validation. The trace therefore supports the general immune-effect direction, while indicating that the strongest cross-variant claim is matched at a relatively coarse concept level.

Stage 3 – Mechanism. The mechanism stage receives a score of 3.0 and is assigned partial support. Five mechanism chains are evaluated through local edge-level proof obligations:

- **Chain 1** (0.882): `Size` → lymph-node targeting → dendritic-cell uptake. The first edge receives strong support from a size-to-lymph-node-targeting rule. The second edge is directionally plausible but more approximate, because the matched theorem links lymph-node targeting to an antibody-related outcome rather than directly proving dendritic-cell uptake.
- **Chain 2** (0.744): `TLR7/8 agonist` → controlled release → endosomal TLR activation.

Both edges receive moderate support. The matched evidence captures related immune activation and TLR-pathway activation, but the generated chain compresses release, intracellular delivery, and receptor engagement into a short transition. This makes the proof direction plausible but not highly specific.

- **Chain 3** (0.799): `Cationic nanoemulsion` → membrane binding → antigen presentation. The first edge is supported by a cationic-lipid-charge-to-cell-membrane binding rule. The second edge is supported by an antigen-presentation-related rule. This chain is one of the clearer mechanistic paths in the case, although the matched evidence remains formulation-general rather than antigen-specific.
- **Chain 4** (0.381, Partial): `Encapsulation of TLR7/8 agonist and RBD antigen` → sustained immune activation → Th1 priming. The first edge receives partial support from a sustained-release-related rule. The second edge receives weak support from a rule about dendritic-cell migration and antigen-specific T-cell priming. This is directionally consistent with the expert comment that antigen loading and sustained release were not specified in sufficient detail in the generated response. The trace therefore treats this chain as partially supported rather than fully verified.
- **Chain 5** (0.000, Hypothesis): `CD80/CD86 expression` → dendritic-cell maturation → T-cell activation. This chain contains immunologically related events, but the generated ordering is coarse because CD80/CD86 expression is more naturally interpreted as a marker or consequence of dendritic-cell maturation rather than its upstream cause. No high-support theorem rule is retrieved for either local transition. SAVANT marks the chain as a hypothesis-level knowledge gap rather than forcing support from loosely related evidence.

Summary. Overall, the trace is directionally consistent with the expert evaluation pattern of 4/4/3. The construction and terminal immune-effect claims are broadly supported, while the mechanism stage is weaker because several generated transitions are either too coarse or lack direct

theorem-library support. Importantly, the expert comments were made without access to the verification trace, yet they point to similar regions of uncertainty, cross-variant protection is supported mainly at a broad endpoint level, and the antigen-loading or sustained-release mechanism is only partially grounded in the Stage 3 proof trace. This case illustrates that SAVANT can assign a partial final verdict when the generated design is plausible but the mechanistic chain contains under-specified or weakly supported steps.

GPT-4o: Vaccine Adjuvant Design for Classical Swine Fever Virus (CSFV) with a SpyCatcher/SpyTag -Constrained Virus-Like Nanoparticle System To address the need for a modular vaccine platform for CSFV, leveraging the SpyCatcher/SpyTag system and virus-like nanoparticles (VLPs), the adjuvant must be carefully designed to enhance humoral immunity, promote Th1 polarization, and drive robust dendritic cell (DC) activation. Below, I outline a scientifically grounded design for such an adjuvant system Adjuvant Construction and Physical/Chemical Properties Type and Formulation: The adjuvant will be a hybrid system composed of a stable Toll-like receptor (TLR) agonist-conjugated lipid nanoparticle (LNP) combined with a cationic polymer scaffold. The LNP will encapsulate a TLR7/8 agonist (e.g., resiquimod, R848) for endosomal delivery and surface-conjugate a TLR4-agonist (e.g., monophosphoryl lipid A, MPLA) for extracellular signaling. Additionally, the adjuvant will incorporate a cationic polymer, polyethyleneimine (PEI, molecular weight ~25 kDa), functionalized with SpyCatcher domains to physically bind SpyTag-labeled VLPs. Size and Charge: The LNPs will be optimized to a size of ~100–120 nm in diameter with a slightly negative zeta potential (~ -10 mV) for efficient uptake by DCs. The polymer scaffold will be positively charged (+20–30 mV) to enhance electrostatic interactions with cell membranes and facilitate endosomal escape. Preparation Method: LNP Formation: The TLR7/8 agonist will be encapsulated into liposomes using thin-film hydration followed by extrusion. The lipid composition will include phosphatidylcholine, cholesterol, and MPLA-conjugated phospholipids. Polymer Functionalization: PEI will be conjugated with SpyCatcher via N-hydroxysuccinimide (NHS) chemistry to allow modular assembly with SpyTag-labeled VLPs. Adjuvant Assembly: SpyTag-labeled VLP antigens will be mixed with the SpyCatcher-functionalized PEI scaffold, followed by complexation with LNPs via mild agitation. Expected Immune Effects and Outcomes Humoral Immunity: The adjuvant will elicit high titers of CSFV-neutralizing antibodies, driven by the robust activation of follicular helper T cells (T_{fh}) and germinal center B cells. The display of antigens on VLPs, in combination with strong DC activation, will enhance B cell receptor cross-linking and antigen processing, amplifying antibody production. Th1 Polarization: The simultaneous activation of TLR4 and TLR7/8 will skew the CD4⁺ T cell response toward a Th1 phenotype, characterized by the production of cytokines such as interferon-gamma (IFN-γ), interleukin-12 (IL-12), and tumor necrosis factor-α (TNF-α). This polarization will be crucial for controlling intracellular viral replication. Dendritic Cell Activation: DCs will be potentially activated by the dual TLR agonists. MPLA will engage cell surface TLR4, while the R848 encapsulated in LNPs will target endosomal TLR7/8. This dual activation will upregulate major histocompatibility complex (MHC) class I and II molecules, as well as co-stimulatory molecules (CD80, CD86), enhancing antigen presentation to T cells. Cytokine Profiles: The adjuvant will induce a pro-inflammatory cytokine milieu, including IL-12, TNF-α, and IFN-γ, to promote Th1 polarization. IL-6 will also be produced to support T_{fh} differentiation and B cell responses. Protection: The vaccine system is expected to induce sterilizing immunity against CSFV by neutralizing the virus in the extracellular space and eliminating infected cells via Th1-mediated mechanisms. Step-by-Step Causal Mechanism from Adjuvant Design to Immune Response Adjuvant Uptake by APCs: The LNPs and VLP-polymer complexes are taken up by DCs via macropinocytosis and receptor-mediated endocytosis, facilitated by their nanoscale size (~100–120 nm) and surface charge properties. TLR Activation: - MPLA on the LNP surface interacts with TLR4 on the DC plasma membrane, triggering MyD88- and TRIF-dependent signaling pathways. - R848 is released in the endosome following LNP degradation and activates TLR7/8, further amplifying MyD88-dependent signaling. Cytokine Secretion and DC Maturation: The dual TLR stimulation induces the production of IL-12, TNF-α, and type I interferons, promoting DC maturation. Mature DCs upregulate MHC molecules and co-stimulatory proteins (CD80, CD86), enhancing their ability to interact with T cells. Antigen Presentation: - SpyTag-labeled VLPs bound to the polymer scaffold are efficiently processed and presented on MHC class I and II molecules. - Cross-presentation of antigens to CD8⁺ T cells is facilitated by the cationic PEI scaffold, which promotes endosomal escape of antigenic peptides. T Cell Activation and Polarization: - CD4⁺ T cells primed by DCs differentiate into Th1 cells under the influence of IL-12 and type I interferons. - CD8⁺ T cells are activated via cross-presentation, enabling cytotoxic responses against infected cells. B Cell Activation and Humoral Immunity: - The particulate nature of the VLPs enhances B cell receptor cross-linking, leading to robust activation. - T_{fh} cells, driven by IL-6 and IL-21, provide cognate help to B cells in germinal centers, promoting high-affinity antibody production. Immune Effector Functions: - Neutralizing antibodies target CSFV particles in the extracellular space, preventing infection. - Th1-mediated cytotoxic responses eliminate virus-infected cells, providing intracellular control of the virus. This adjuvant system combines modularity, potent humoral immunity, and Th1 polarization, making it an ideal candidate for a SpyCatcher/SpyTag-constrained VLP-based CSFV vaccine. The dual TLR activation and antigen delivery strategy ensure robust activation of both the innate and adaptive immune systems, leading to long-lasting protection against CSFV.

(a) GPT-4o natural-language response for Case A.

SAVANT: Vaccine and construction Vaccine: Name = CSFV SpyCatcher/SpyTag VLP Vaccine Target disease = Classical Swine Fever Virus (CSFV) Structure = SpyCatcher/SpyTag-constrained VLP system Antigen: Name = SpyTag-Labeled VLP Antigen Component = SpyTag-labeled VLP [Protein] Preparation = conjugate (SpyCatcher/SpyTag system) Adjuvant: Name = TLR Agonist-Conjugated LNP with PEI Scaffold Type = Nanoparticle Size = 100–120 nm Shape = Spherical Zeta potential = ~-10 mV (LNP), +20–30 mV (polymer) Components = [R848 [SmallMolecule], MPLA[Lipid], Polyethyleneimine (PEI) [Polymer], SpyCatcher[Protein]] Preparation = assemble Immune effect claims Claim = Enhance (humoral immunity) Claim = Polarize (Th1 response) Claim = Activate (dendritic cells) Claim = Protect (CSFV infection) Mechanism chains Size (100–120 nm) >> Uptake (dendritic cells) >> Activate (TLR7 and TLR4 pathways) Contains (MPLA and R848) >> Activate (maturation of dendritic cells) >> Secrete (IL-12 and IFN-gamma) Form (SpyCatcher/SpyTag VLP-polymer complex) >> Present (antigens on MHC I and II) >> Prime (CD8 and CD4 T cells) Sustain (antigen exposure) >> Enhance (antibody production) >> Neutralize (CSFV particles)
--

(b) SAVANT ontology-constrained DSL rendering for Case A.

Figure 4: Case visualization of natural-language response rendering. (a) shows the GPT-4o-generated scientific rationale, and (b) shows the corresponding SAVANT DSL representation after null fields are removed. The DSL separates construction information, terminal immune-effect claims, and ordered mechanism chains for downstream theorem-library verification.

GPT-4o:
Adjuvant Design for SARS -CoV-2 Vaccine

Adjuvant Construction and Physical/Chemical Properties
To achieve robust anti -receptor-binding domain (RBD) IgG, IgG1, and IgG2c responses, as well as cross -variant RBD IgG, the adjuvant system will be based on a nanoemulsion platform combined with a Toll -like receptor (TLR) agonist. The proposed formulation integrates two key components:

- **Cationic Nanoemulsion (CNE):**
 - Composition: The nanoemulsion is composed of squalene (an FDA -approved biodegradable oil), stabilized with Tween 80 (a nonionic surfactant) and DOTAP (a cationic lipid to impart positive surface charge).
 - Size and Charge: The particle size will be maintained at ~150 nm, as particles in this range are effectively taken up by antigen -presenting cells (APCs) such as dendritic cells. The zeta potential will be approximately +20 mV to facilitate interaction with negatively charged cell membranes.
 - Preparation Method: The nanoemulsion will be formulated using high -energy ultrasonication, which ensures homogeneity and stability of the formulation. It will be optimized for long -term stability under refrigerated conditions (2 -8°C) by adding cryoprotectants like trehalose during lyophilization.
- **TLR7/8 Agonist:**
 - Molecular Component: Imidazoquinolinone derivatives (e.g., resiquimod, R848) will be encapsulated in the nanoemulsion. These molecules are potent activators of TLR7/8, which are expressed on dendritic cells and monocytes.
 - Encapsulation Efficiency: The hydrophobic nature of the TLR agonist ensures high encapsulation efficiency (>90%) within the squalene core of the nanoemulsion. This encapsulation ensures controlled release and prevents systemic toxicity.
- **Stability and Storage:** The nanoemulsion system is optimized for long -term stability under refrigerated conditions, ensuring vaccine integrity during storage and transport. Lyophilized formulations can be reconstituted without loss of efficacy.

Expected Immune Effects and Outcomes
The proposed adjuvant system is designed to induce the following immune effects:

- **Anti-RBD IgG Response:** Cationic nanoemulsions enhance the delivery of RBD antigen to dendritic cells, boosting overall IgG production.
- **IgG Subclass Responses:** The addition of TLR7/8 agonists promotes a strong Th1 -skewed response, leading to elevated IgG2c levels (in mice, equivalent to IgG1 in humans). Simultaneously, the balanced adjuvant properties allow for IgG1 production, ensuring a mixed Th1/Th2 response.
- **Cross-Variant RBD IgG:** By enhancing antigen uptake and presentation, the adjuvant system facilitates the generation of broadly neutralizing antibodies against conserved epitopes of the RBD, potentially effective against multiple variants.
- **Cytokine Profiles:** TLR7/8 activation leads to the production of type I interferons, IL -12, and TNF -α, which drive Th1 responses. IL -6 and IL -4 secretion are moderated to avoid excessive inflammation while supporting antibody production.
- **T Cell Responses:** The adjuvant system enhances CD4+ T cell activation (Th1 -biased) and promotes differentiation of cytotoxic CD8+ T cells, critical for clearing infected cells.

Step-by-Step Causal Mechanism from Adjuvant Design to Immune Response

- **Antigen Delivery and Uptake:**
 - The cationic surface of the nanoemulsion facilitates electrostatic interactions with negatively charged cell membranes, enhancing uptake by dendritic cells and macrophages via endocytosis.
 - The 150 nm particle size ensures efficient drainage to lymph nodes, where professional APCs reside.
- **Controlled Antigen Release:**
 - The encapsulated TLR7/8 agonist and the RBD antigen are co -delivered to APCs, ensuring simultaneous activation of innate immune sensors and antigen processing.
 - The slow release of the TLR agonist from the nanoemulsion prolongs immune activation, enhancing the magnitude of the response.
- **Innate Immune Activation:**
 - Upon uptake, the TLR7/8 agonist activates endosomal TLRs in dendritic cells and monocytes, leading to the production of pro-inflammatory cytokines (e.g., IL -12, IFN -α) and upregulation of costimulatory molecules (CD80, CD86).
 - The resulting cytokine milieu favors a Th1 -biased adaptive immune response.
- **Antigen Presentation and T Cell Activation:**
 - Processed RBD peptides are presented on MHC class I and II molecules to CD8+ and CD4+ T cells, respectively.
 - CD4+ T cells differentiate into Th1 cells, driven by IL -12, producing IFN -γ to support IgG2c production and cytotoxic T cell activation.
- **B Cell Activation and Antibody Production:**
 - The balanced cytokine environment supports germinal center formation and B cell differentiation into plasma cells, producing high -affinity anti-RBD antibodies (IgG, IgG1, IgG2c).
 - Cross-variant neutralization is achieved by targeting conserved epitopes, enabled by robust antigen presentation and T cell help.
- **Memory Formation:**
 - The strong innate and adaptive immune activation drives the generation of long -lived memory B and T cells, ensuring durable immunity and rapid recall responses upon exposure to SARS-CoV-2 variants.

This adjuvant system leverages synergistic innate immune activation and precision antigen delivery to achieve potent and durable immune responses against SARS -CoV-2 and its evolving variants.

(a) GPT-4o natural-language response for Case B.

SAVANT:

Vaccine and construction

Vaccine:

```
Name = SARS-CoV-2 vaccine with cationic nanoemulsion
Target disease = SARS-CoV-2
preparation = Adjuvant and antigen assembly
Preparation type = assemble
Details = Combination of nanoemulsion and RBD antigen
```

Antigen:

```
Name = RBD antigen
Component = Receptor Binding Domain (RBD) [Protein]
Targeted response = anti-RBD IgG, IgG1, IgG2c
Preparation = Encapsulation into nanoemulsion
Preparation type = encapsulate
Encapsulation technique = encapsulation in squalene core
```

Adjuvant:

```
Name = Cationic Nanoemulsion with TLR7/8 agonist
Type = Nanoparticle
Size = ~150 nm
Shape = spherical
Zeta potential = +20 mV
Stability condition = refrigerated at 2-8°C with cryoprotectants
Components = [ squalene [Lipid], Tween 80 [Lipid], DOTAP [Lipid],
R848 [SmallMolecule] ]
Preparation = High-energy ultrasonication
Preparation type = emulsify
Emulsification method = high-energy ultrasonication
```

Immune effect claims

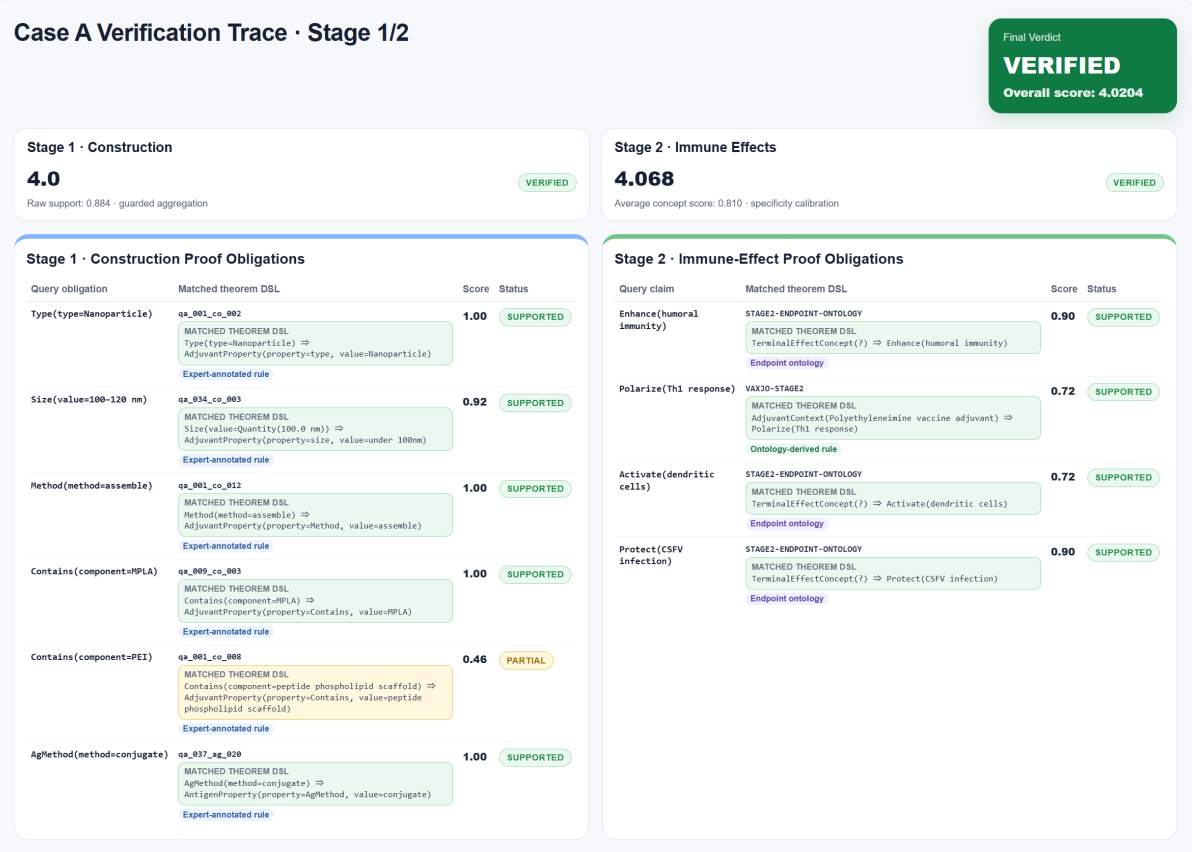
```
Claim = Enhance(IgG production)
Claim = Induce(Th1-skewed T cell responses)
Claim = Increase(cross-variant RBD IgG)
Claim = Protect(against SARS-CoV-2 variants)
Claim = Sustain(antigen release)
```

Mechanism chains

```
Size(150 nm) >> Target(lymph nodes) >> Uptake(dendritic cells)
Contains(TLR7/8 agonist) >> Release(controlled TLR agonist delivery) >>
Activate(endosomal TLRs)
Form(cationic nanoemulsion) >> Bind(cell membranes) >> Enhance(antigen
presentation)
Encapsulate(TLR7/8 agonist and RBD antigen) >> Sustain(immune activation)
>> Prime(Th1 responses)
Express(CD80 and CD86) >> Mature(dendritic cells) >> Activate(T cells)
```

(b) SAVANT ontology-constrained DSL rendering for Case B.

Figure 5: Second case visualization of natural-language response rendering. (a) shows the generated scientific rationale, and (b) shows the corresponding structured DSL representation after null fields are removed. The DSL separates construction information, terminal immune-effect claims, and ordered mechanism chains for downstream verification.



Stage 2 · Immune-Effect Proof Obligations

Query claim	Matched theorem DSL	Score	Status
Enhance(humoral immunity)	STAGE2-ENDPOINT-ONTOLOGY MATCHED THEOREM DSL TerminalEffectConcept(?) ⇒ Enhance(humoral immunity) Endpoint ontology		

 0.90 | SUPPORTED || Polarize(Th1 response) | VAX30-STAGE2 MATCHED THEOREM DSL AdjuvantContext(Polyethyleneimine vaccine adjuvant) ⇒ Polarize(Th1 response) Ontology-derived rule |

Figure 6: Verification trace for Case A, Stages 1–2. Green indicates high-support matches and yellow indicates partial or approximate matches. Full evidence text, long source titles, internal debug fields, and null fields are omitted for readability.

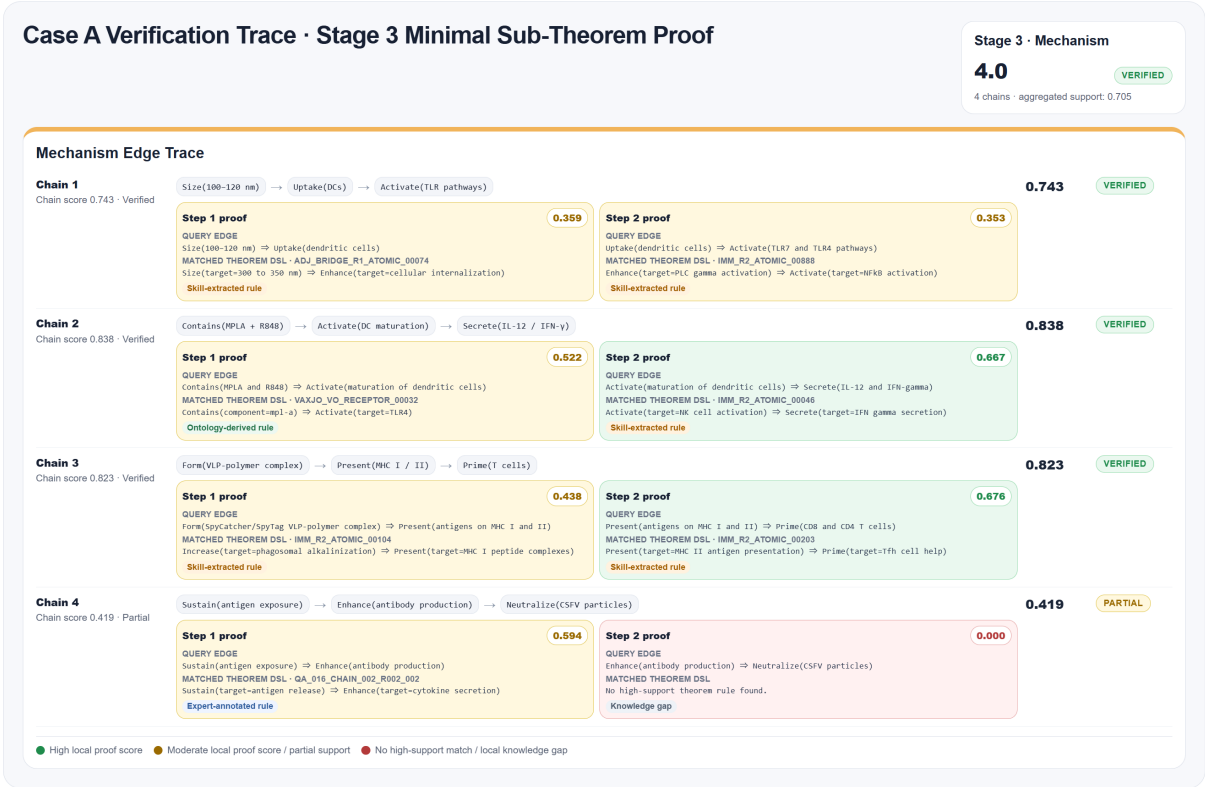
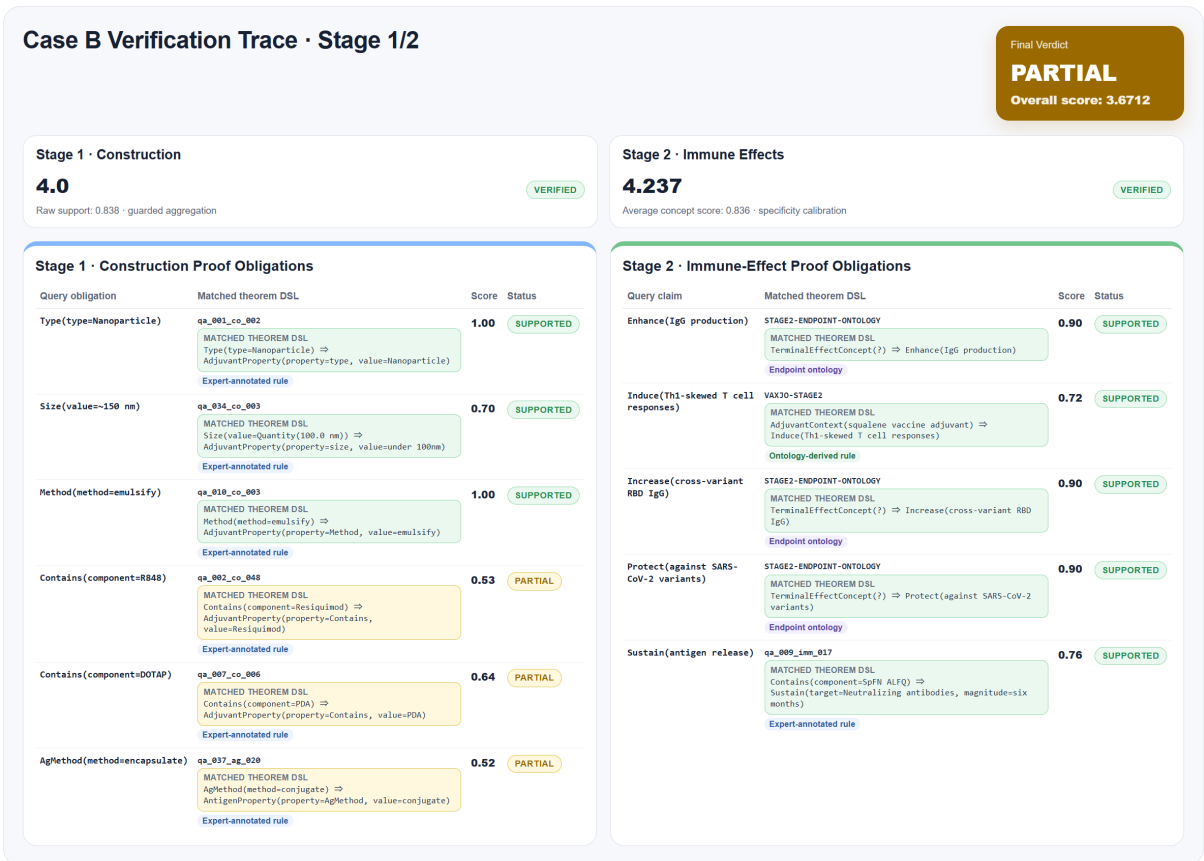


Figure 7: Verification trace for Case A, Stage 3. The figure shows minimal sub-theorem proof over adjacent mechanism edges. Green indicates high local support, yellow indicates partial support, and red indicates a local knowledge gap.



Stage 2 · Immune-Effect Proof Obligations

Query claim	Matched theorem DSL	Score	Status
Enhance(IgG production)	STAGE2-ENDPOINT-ONTOLOGY MATCHED THEOREM DSL TerminalEffectConcept(?) ⇒ Enhance(IgG production) Endpoint ontology		

 0.90 | SUPPORTED || Induce(Th1-skewed T cell responses) | VAX30-STAGE2 MATCHED THEOREM DSL AdjuvantContext(squalene vaccine adjuvant) ⇒ Induce(Th1-skewed T cell responses) Ontology-derived rule |

Figure 8: Verification trace for Case B, Stages 1–2. Green indicates high-support matches and yellow indicates partial or approximate matches. Full evidence text, long source titles, internal debug fields, and null fields are omitted for readability.

