

HAAAG

Human-AI Agentic Framework for Scientific Research and Publication

FROM RESEARCH IDEA TO REPRODUCIBLE PUBLICATION



HUMAN-AGENTIC ARTICLE GENESIS

HAAG FRAMEWORK v1.1

A HUMAN-AGENT FRAMEWORK FOR CONDUCTING SCIENTIFIC RESEARCH AND PRODUCING PUBLICATIONS

Scientific explanations for researchers and a formal specification for agent-based systems

Dmytro Lande · 2026

Abstract

This volume systematizes the methodological, architectural, and operational foundations of Human-Agent Article Genesis (HAAG), a human-agent framework for organizing the research process. Unlike approaches focused directly on generating scientific prose, HAAG treats the sequence of research states, hypotheses, verification procedures, human decisions, and documented results as the object of control. The presentation combines explanations of scientific reasoning intended for human readers with formal identifiers, commands, and procedural descriptions intended for interpretation by agent-based systems. Particular attention is paid to falsification, experiment preregistration, claim provenance, reproducibility, and the limits of agent autonomy.

Principle of Dual Interpretation

The human-oriented layer provides definitions, motivations, assumptions, methodological constraints, and explanations of why research transitions occur. The machine-oriented layer comprises stable identifiers (AGENT, HUMAN, VERIFY, SOURCE, PREREGISTER, RUN, AUDIT), PROFILE parameters, the Research State structure, transition rules, and protocols for preserving research artifacts. Machine-oriented instructions should be understood as a declarative specification: their actual execution depends on the implementation of the orchestrator, available tools, and validation procedures. Merely specifying a command does not establish that the corresponding operation has been performed.

Editorial Note

This edition retains the original concept and principal body of HAAG v1.1. Mathematical expressions for which the source document provides valid LaTeX equivalents have been converted into editable Word mathematical objects (OMML). Ambiguous or incomplete formulas have not been reconstructed arbitrarily. Statements concerning source verification, experiment execution, and reproducibility describe normative requirements of the framework; they do not certify that any particular study has been carried out.

HAAG v1.1

Human-AI Augmented Generation Framework
for Reproducible Research and Scholarly Publication

FROM IDEA
TO VERIFIED KNOWLEDGE
HUMAN JUDGMENT × AI CAPABILITIES × SCIENTIFIC RIGOR



1. Purpose and Concept of HAAG

1.1. From Text Generation to Conducting Research

Large language models have substantially changed the preparation of scientific texts. They can produce coherent prose, propose article structures, explain mathematical constructions, analyze data, assist with programming and computational experiments, identify contradictions, and prepare illustrations. However, the ability to generate well-written text does not, in itself, imply the ability to conduct a scientifically sound investigation.

Human-Agent Article Genesis (HAAG) starts from precisely this distinction.

HAAG should not be understood as a large, general-purpose prompt of the “write a scientific paper on a given topic” variety. Rather, it is a framework for organizing interactive scientific research in which a human researcher and a set of specialized LLM agents proceed from an initial topic through verified results to a completed publication.

The formal HAAG specification makes this distinction explicit: the framework is intended to generate and execute interactive research programs, while a publication emerges from a traceable human-agent research trajectory.

The central object of HAAG is therefore the research process, not the article text.

Topic → Research Program → Human-Agent Research Trajectory → Verified Results → Scientific Publication

This fundamentally distinguishes HAAG from the conventional use of generative AI to draft scientific papers.

1.2. Core Principle: Agents Explore, Humans Determine the Trajectory

HAAG rests on a concise principle:

Agents explore the space of possibilities; the human researcher shapes the research trajectory.

This is how the principle is formulated in the framework specification. It means that a large language model must not independently make all scientifically consequential decisions without the researcher's knowledge.

For example, the system may propose several hypotheses. It may identify supporting arguments and counterarguments for each, retrieve related publications, construct mathematical models, and even propose an experiment.

However, if choosing between two alternatives changes the substance of the research, the final transition must be a deliberate decision.

HAAG introduces a dedicated operator, HUMAN, for this purpose.

The human role is therefore not limited to writing the initial request and editing the final answer. The researcher remains an active participant, intervening specifically at decision points with scientific consequences.

In schematic terms:

LLM agents → alternatives, arguments, models, verification procedures;

HUMAN → selection or modification of the research trajectory.

This combines the breadth of machine-assisted exploration with human responsibility for the scientific substance of the investigation.

1.3. HAAG as a Human-Agent System

Within HAAG, the large language model effectively ceases to function as a single, general-purpose conversational partner. Instead, the research task is distributed among agents with distinct responsibilities.

One agent identifies gaps in existing research. Another formulates a theoretical model. A third deliberately seeks weaknesses in the proposed hypothesis. Another designs an experiment. Dedicated agents handle numerical verification, sources, figures, peer review, and the preparation of reproducibility materials.

An important feature of the HAAG architecture is that an agent is not necessarily expected to endorse the proposals of a preceding agent.

On the contrary, some agents are specifically designed to provide criticism.

Research thus takes the form of a structured, controlled scientific debate:

idea → hypothesis → counterhypothesis → falsification → experiment → verification → interpretation.

This makes the agent system resemble a process of collective scientific analysis rather than an automated text-writing mechanism.

1.4. The Article Is an Outcome, Not the Starting Point

For HAAG, a fundamental change in the conventional workflow with LLMs is essential.

A conventional workflow might look as follows:

Topic → Prompt → Article Text.

HAAG follows a different logic:

Topic → Problem → Hypotheses → Verification → Method → Experiment → Results → Interpretation → Article.

Text is generated only after the relevant part of the Research State has become sufficiently well defined.

This leads to the fundamental question posed by HAAG. The system should not primarily ask:

“What text should be generated next?”

Instead, it should ask:

“What scientifically justified research transition should be performed next?”

This proposition is subsequently formalized in the fundamental rule of HAAG v1.1.

1.5. Research Trajectories Rather Than Linear Generation

Actual research is rarely a straight path from problem formulation to conclusions.

A hypothesis may fail to receive support. A mathematical model may prove unstable. An experiment may reveal a null effect. A newly identified publication may show that the claimed novelty has already been established. A selected metric may turn out to be unsuitable. Real-world data may demonstrate only the applicability of a method, without establishing a causal relationship.

Accordingly, HAAG supports not only forward progression but also:

FORK, BACKTRACK, REJECT, REWRITE, MERGE.

In other words, a Research Trajectory may branch, return to earlier states, and discard unsuccessful directions.

Importantly, a rejected hypothesis or failed experiment must not simply disappear from the research record. Both belong to TRACE, the traceable history of the research process.

HAAG therefore has the potential to make reproducible not only the final text but also the reasoning that led to it.

1.6. A Negative Result Is Still a Result

One of the most fundamental features of HAAG v1.1 is its rejection of the implicit assumption that research must necessarily end by confirming its initial hypothesis.

If an experiment finds no expected effect, this must not automatically trigger repeated parameter adjustments until the desired effect appears.

The framework explicitly establishes the following rule:

Negative and null results are valid RESULTS.

This means that HAAG must be able to produce a scientifically meaningful result of the following form:

Not confirmed under the specified conditions.

Such a result may be no less important than confirmation of a hypothesis.

For this purpose, the framework introduces a dedicated cycle:

PREREGISTER → FREEZE → RUN → AUDIT

The model, parameters, metrics, success criteria, and stopping conditions are specified and recorded first. Only then is the experiment performed.

This sequence reduces the risk of silently adapting the model to obtain a desired outcome.

1.7. Scientific Artifacts as Independent Research Outputs

HAAG v1.1 broadens the very notion of a research output.

A scientific result is not limited to the paragraphs ultimately included in an article.

The process also produces substantive outputs, including:

mathematical models;

experimental specifications;

computational results;

negative results;

verified sources;

prompts;
figures and prompts used to generate them;
decision histories;
rejected hypotheses;
tables;
research reproducibility materials;
repository structures.

The original specification expresses this principle by treating calculations, figures, prompts, references, negative results, and reproducibility materials as first-class outputs rather than by-products of writing.

1.8. HAAG as a Prompt-Native, No-Code Framework

HAAG is conceived as a prompt-native, no-code framework.

Prompt-native means that a substantial portion of research procedures is described not in conventional program code but through structured natural language and a set of control primitives:

AGENT, HUMAN, VERIFY, SOURCE, PREREGISTER, RUN, AUDIT, FIGURE, CITE.

The complete set of these primitives is defined in the HAAG specification.

In this context, no-code does not imply abandoning formal methods, mathematical models, or computation. Rather, research logic can be specified in a human-readable form without writing a dedicated program for each article.

For example:

CALL: theorist

CALL: falsifier

IF hypothesis survives:

PREREGISTER experiment

FREEZE experiment

RUN experiment

AUDIT results

ELSE:

BACKTRACK

This fragment is intelligible to both a human reader and an LLM-based system.

HAAG thus occupies an intermediate position between an ordinary textual prompt and a formal programming language for research processes.

1.9. Traceability as a Central Requirement

Whenever the system produces a claim, it should be possible to answer the question:

Where did this claim come from?

The answer may take several forms:

claim $\leftarrow$ {source, mathematical derivation, experiment, dataset, HUMAN decision, interpretation of a result}

HAAG therefore retains not only the text of a claim but also its provenance.

This is fundamental to the use of generative models in science: a fluent sentence does not, by itself, constitute evidence that the statement is correct.

The ultimate goal is therefore not merely a publication, but:

verified publication

a publication in which the provenance of the principal claims can be traced.

1.10. Overall Purpose of HAAG

HAAG can therefore be defined as a human-agent framework for organizing the complete cycle of scientific research and publication development. Large language models are used to explore hypotheses, construct models, conduct criticism and falsification, design experiments, analyze results, work with sources, prepare figures, and synthesize text, while the human researcher retains control over scientifically consequential transitions between Research States.

2. User Interaction with HAAG

2.1. Minimal Entry Requirements

Despite its complex internal organization, HAAG is deliberately designed to make the initial interaction as simple as possible.

In the simplest case, the researcher needs to specify only a topic:

USER:

TOPIC:

String Theory of Information Influence in Society

through Meanings, Narratives, and Memes

COMMAND:

Run HAAG.

This is sufficient to initiate the first research cycle.

The original framework explicitly supports this mode: TOPIC is effectively the only mandatory input. The publication type, language, target venue, length, subject area, data, figure style, and repository requirements can all be specified optionally.

This is an important property of HAAG: the complexity of the internal research process should not be imposed on the user in the form of an excessively long initial prompt.

2.2. Extended Initial Request

If the researcher already knows the requirements for the intended publication, these can be provided together with the topic.

The formal interface is as follows:

USER:

TOPIC:

[research topic]

OPTIONAL:

PUBLICATION TYPE:

[publication type]

LANGUAGE:

[language]

TARGET VENUE:

[target journal or conference]

LENGTH:

[desired length]

DOMAIN:

[subject area]

SPECIAL REQUIREMENTS:

[additional requirements]

DATA / FILES / CORPUS:

[data, files, document corpus]

FIGURE STYLE:

[figure style]

REPOSITORY REQUIREMENTS:

[requirements for GitHub / supplementary materials]

COMMAND:

Run HAAG.

This structure directly corresponds to the interface specified in version 1.1.

TOPIC defines the initial subject area or research idea. It need not yet be formulated as a scientific hypothesis. One of HAAG's tasks is precisely to move from a broad topic to a researchable problem.

PUBLICATION TYPE specifies the expected output: conference abstract, short paper, full-length article, review, monograph, or book chapter.

LANGUAGE specifies the language of the final publication and associated materials.

TARGET VENUE enables the structure and formatting to be aligned with a particular journal, conference, or publisher.

LENGTH specifies the intended scope of the text.

DOMAIN clarifies the subject area, which is particularly important for interdisciplinary topics.

SPECIAL REQUIREMENTS allows the researcher to specify constraints such as mathematical depth, the need for a particular experiment, citation format, or article structure.

DATA / FILES / CORPUS supplies the empirical or documentary basis of the study.

FIGURE STYLE specifies the requirements for scientific graphics.

REPOSITORY REQUIREMENTS defines the materials to be prepared for GitHub or another repository.

2.3. HAAG Should Not Ask Unnecessary Questions

An important aspect of the interaction model is the principle of adaptive human interaction.

If the system can safely determine a technical parameter on its own, it should not interrupt the researcher with a question.

For example, there is no need to ask repeatedly:

“What heading format would you prefer?”

or:

“Should numbered lists be used?”

Such decisions do not alter the scientific trajectory.

The situation is different when the system identifies two fundamentally distinct alternatives:

A. Test the hypothesis using a synthetic model;

B. Use a real-world dataset.

This choice affects both the methodology and the strength of the conclusions that can legitimately be drawn.

At this point, HAAG should consult the human researcher.

The specification expresses this principle precisely: when only a topic is provided, the system should autonomously determine only those parameters that can be determined safely. It should involve the human when a decision materially changes the research trajectory, evidence base, experimental design, interpretation, or form of publication.

2.4. The Principle of Adaptive Human Involvement

Thus, HAAG prescribes neither fully autonomous generation nor continuous manual control of every step.

Instead, it introduces an adaptive mode between these extremes:

automate routine operations + refer scientifically consequential decisions to the human researcher.

This can be represented by a simple decision logic:

$$D = \begin{cases} \text{AGENT,} & \text{if the decision is technical or unambiguous,} \\ \text{HUMAN,} & \text{if the decision changes the research trajectory.} \end{cases}$$

For this reason, HAAG does not seek to replace the researcher. Its purpose is to expand the range of alternatives that can be investigated meaningfully while preserving an account of why the research followed a particular path.

2.5. Example of Initiating a Real Research Project

Suppose the user enters only:

TOPIC:

Information Memory of Society

COMMAND:

Run HAAG.

The system should not immediately generate an article under this title.

Its first output should instead refine the research problem. For example, the agents may identify several possible interpretations of information memory:

H1: memory as residual activity;

H2: memory as a change in semantic structure;

H3: memory as a change in response to a subsequent influence.

Critical analysis may reveal that these three phenomena should not be combined into a single construct.

HAAG would then formulate a HUMAN decision point along the following lines:

Context. Three distinct interpretations of information memory have been identified.

Problem. Combining them into a single quantity could conceal different underlying mechanisms.

Options.

A — investigate residual activity only;

B — investigate structural memory;

C — distinguish several components of memory.

Question. Which conceptual model should be developed?

This is no longer text generation.

It is the interactive development of a research investigation.

2.6. Why the Initial Prompt Can Remain Short

Conventional approaches to working with LLMs often attempt to embed the entire methodology in a single, very long prompt.

HAAG proposes a different model.

The initial request can remain brief because the framework itself contains rules for subsequently expanding the research process:

Short Request + HAAG → Comprehensive Research Program.

Thus, complexity is shifted from the user's request to the framework's architecture.

This is why the command

Run HAAG.

has a deeper meaning than the ordinary instruction “write an article.”

In effect, it means:

Construct and execute an interactive research program for the specified topic, involving specialized agents, testing hypotheses and sources, referring scientifically consequential decisions to the human researcher, preserving the research history, and producing a publication only from results that have undergone the required verification.

3. Scientific Publication Profile

3.1. Why HAAG Needs a Publication Profile

The same scientific topic can be presented in fundamentally different ways. The outcome may be a short conference paper, a full journal article, a literature review, a monograph chapter, or an extensive monograph. This determines not only the length of the text but also

the research design, the required level of verification, the structure of the argument, and the composition of the final artifacts.

For example, one clearly formulated hypothesis and one experiment may suffice for a short paper. A full journal article may require several experiments, control scenarios, an analysis of limitations, and comparisons with other methods. A review, in turn, requires a systematic examination and organization of the literature.

Therefore, HAAG establishes a publication profile—PROFILE—at the outset.

Its formal representation is as follows:

PROFILE:

TYPE:

THESIS |

SHORT_PAPER |

ARTICLE |

REVIEW |

MONOGRAPH |

MONOGRAPH_CHAPTER |

AUTO

DOMAIN: AUTO

LANGUAGE: AUTO

TARGET_LENGTH: AUTO

TARGET_VENUE: AUTO

CITATION_STYLE: AUTO

HUMAN_INTERACTION: ADAPTIVE

NOVELTY_LEVEL: HIGH

STRUCTURE: AUTO

FIGURE_POLICY: LACONIC_SCIENTIFIC

REPRODUCIBILITY_PACKAGE: AUTO

REFERENCE_WORKING_MODE: SCHOLAR_QUERY

This block is not merely a set of formatting parameters. It establishes the context in which HAAG must plan the subsequent research process. This set of profile fields is defined in specification v1.1.

3.2. Type of Intended Publication

The parameter

TYPE

specifies the type of scholarly output to be produced.

HAAG provides the following basic options.

THESIS — conference abstract or extended abstract. This compact form should present the problem, central idea, method, and most important result without excessive theoretical elaboration.

SHORT_PAPER — a short research paper. It must already present a complete scientific argument, although the number of research questions, experiments, and figures is generally limited.

ARTICLE — a full-length research article. This can be considered HAAG's primary mode. It includes problem formulation, analysis of prior work, identification of a research gap, methods, experiments or other forms of verification, results, discussion, and conclusions.

REVIEW — a review paper. In this mode, the research logic itself changes. It is necessary to define source search and selection criteria, classify approaches, identify contradictions and unresolved problems, and outline promising directions.

MONOGRAPH — a monograph. Here HAAG must address not only the structure of an individual article but also the hierarchy of chapters, consistent terminology throughout the book, and relationships among multiple models and findings.

MONOGRAPH_CHAPTER — an individual monograph chapter that must be self-contained while remaining consistent with the overall concept of the book.

Finally,

AUTO means that the publication type has not been explicitly specified and HAAG may propose the most appropriate form on the basis of the task.

An automatic selection is not necessarily irreversible. If the publication type substantially affects the research trajectory, the system may refer the decision to the HUMAN operator.

3.3. Subject Area

The parameter

DOMAIN

specifies the subject area of the research.

For example:

DOMAIN:

Cybersecurity

or

DOMAIN:

Complex Networks / NLP / Information Dynamics

The subject area influences terminology, source types, the nature of models, acceptable verification procedures, and the expected style of scientific argumentation.

In mathematical research, a proof may constitute the central result. Experimental research requires data and statistical verification. Cybersecurity research may rely on real incidents, CTI/OSINT sources, and official reports. NLP research may require corpora, evaluation datasets, and linguistic validation.

The setting

DOMAIN: AUTO

means that HAAG infers the subject area from the topic and initial materials.

For interdisciplinary research, multiple subject areas may be specified.

3.4. Language of the Work

The parameter

LANGUAGE

specifies the language of the final publication.

In AUTO mode, it is determined from the interaction context or the target venue's requirements.

This parameter concerns more than translating the completed text. Language also affects terminology, figure captions, tables, search-query wording, and bibliographic work.

At the same time, the language used to communicate with the researcher need not match the language of the future article. For example, the research process may be conducted in Ukrainian while the final article is written in English.

3.5. Target Length

The parameter

TARGET_LENGTH

sets the expected length of the publication.

This matters not only at the final editing stage.

If a six-page paper is planned, it makes little sense to design ten extensive theoretical subsections and seven experiments. Conversely, excessive compression of theoretical explanations may undermine a monograph.

Thus,

Scope → Depth of Structure → Number of Research Components → Level of Detail.

In AUTO mode, HAAG determines a reasonable length in view of the publication type and target venue.

3.6. Target Venue

The parameter

TARGET_VENUE

identifies the journal, conference, publisher, or other venue for which the work is being prepared.

If the target venue is known, HAAG can account for its requirements concerning structure, length, citation style, figures, and supplementary materials.

An important rule nevertheless applies:

A journal's requirements may change how research is presented, but they must not alter the scientific results obtained.

In other words, the result comes first and is then adapted to the publication format.

3.7. Citation Style

The parameter

CITATION_STYLE

specifies the final format of bibliographic references.

This may be a numbered style,

[1], [2], [3],

an author–year style, or a publisher-specific style.

During the research stage, HAAG deliberately separates searching for and verifying sources from their final bibliographic formatting.

Reference formatting must therefore not obstruct the process of verifying the literature.

3.8. Adaptive Human Interaction

The setting

HUMAN_INTERACTION: ADAPTIVE

is of fundamental importance.

It means that HAAG operates neither in a fully autonomous mode nor by constantly questioning the user.

The system should independently perform transitions that do not change the scientific substance of the work and consult the researcher when a decision genuinely affects its trajectory.

For example:

Font size selection → AUTOMATIC,

but

Selection between two competing hypotheses → HUMAN.

This implements the principle:

The machine performs routine tasks and explores alternatives; the human retains control over scientifically significant decisions.

3.9. Novelty Level

The setting

NOVELTY_LEVEL: HIGH

directs the framework toward identifying a genuinely significant scientific contribution.

However, HIGH does not mean that HAAG should declare the resulting work novel.

On the contrary, it imposes a requirement for more rigorous verification.

Before making a strong novelty claim, it is necessary to:

New Idea → Search for Closest Analogues → Search for Alternative Terms
→ Comparison → Falsification of Novelty
→ Cautious Formulation of the Scientific Contribution.

If closely related work is found, HAAG must not conceal it; instead, it must clarify the specific contribution of the present study.

Thus, a high novelty level primarily entails a demanding standard of novelty verification.

3.10. Publication Structure

The parameter

STRUCTURE: AUTO

allows HAAG to establish a structure appropriate to the type of research.

For articles and short papers, the basic logic is IMRaD:

Introduction → Methods → Results → Discussion

The original framework explicitly establishes IMRaD as the standard research logic for ARTICLE and SHORT_PAPER.

HAAG does not, however, apply IMRaD mechanically.

This is particularly important for theoretical studies.

The Methods section may include:

definitions;

a system of concepts;

assumptions;

mathematical constructs;

equations;

algorithms;

models;

simulation procedures.

The Results section may include:

mathematically derived properties;

model behavior;

simulation results;

comparisons among regimes;

negative findings;

results of hypothesis tests.

Thus, HAAG does not require the artificial creation of 'experimental data' merely because the structure contains a Results section.

3.11. Prohibition of Fabricated Results

This is one of the strictest requirements of the profile.

The source framework states it unequivocally:

Never invent empirical results to fill a section.

If a result is unavailable, HAAG must not generate plausible numbers simply to make the article appear complete.

For example, the following reasoning is unacceptable:

Results are empty → Generate results.

The correct logic is:

Results are empty → Determine which results must actually be obtained.

The system must then return to the research stage:

Methods → Experiment → RUN → AUDIT → Results.

This distinction between HAAG and an ordinary article generator is particularly clear here.

3.12. Figure Policy

The setting

FIGURE_POLICY: LACONIC_SCIENTIFIC

defines the overall principle of scientific visualization.

A figure in HAAG is not a decorative illustration. It must serve a specific scientific function:

explain a concept;

show a method;

present a result;

compare regimes;

demonstrate operationalization.

Priority is therefore given to simple, readable scientific graphics without unnecessary decoration.

HAAG specifies more detailed rules for figure creation through the FIGURE and FIGURE_PROMPT primitives.

3.13. Reproducibility Package

The setting

REPRODUCIBILITY_PACKAGE: AUTO

means that HAAG may prepare, alongside the article, the materials needed to understand and reproduce the research.

For example:

README.md

MODEL.md

REPRODUCIBILITY.md

prompts/
experiments/
figures/
references/
real_case/

Thus, the output is not limited to the article file.

3.14. Working Reference Search Mode

The setting

REFERENCE_WORKING_MODE: SCHOLAR_QUERY

defines a special mode of working with scholarly literature.

During drafting, a citation may temporarily take the form:

[5; "adaptive coevolutionary networks" Gross Blasius]

This simultaneously displays:

the number of the prospective source;

the query that can be used to locate and verify that source.

After verification, the working reference

query

is converted into the ordinary citation

[5].

This separates the need for evidence from the automatic fabrication of a bibliographic entry.

3.15. The Profile as a Research Contract

PROFILE can therefore be regarded as an initial contract between the researcher and HAAG.

It answers the question:

What scholarly product are we going to create, and according to which general rules?

In compact form:

PROFILE={type, domain, language, scope, publication, citations, human, novelty, structure, figures, reproducibility}

Once this profile has been established, HAAG can proceed from general configuration to a specific research program.

Editorial note: The original formal identifiers and mathematical objects have been retained. The source contains a duplicated citation expression in §3.7; the English text uses the normalized citation [1], [2], [3].

4. HAAG Architecture and Research State

4.1. From a Universal Framework to a Specific Study

HAAG is a universal framework. By itself, it does not yet contain a specific scientific problem.

The first architectural step is therefore defined by the COMPILE operation:

COMPILE:

HAAG Framework

+ Topic

+ Profile

→ Topic-Specific Interactive Research Program

In ordinary language:

HAAG + topic + profile → topic-specific interactive research program
--

This two-stage COMPILE / EXECUTE scheme is built into the framework specification.

This distinction is important.

HAAG is not a ready-made script for a particular article. It is a metaframework that uses general rules to construct a research program for a specific topic.

For example:

HAAG

+

"String model of meaning dynamics"

+

ARTICLE / English / IMRaD

is transformed into a program in which specific agents, hypotheses, mathematical models, experiments, verification criteria, and figures can be defined.

4.2. Two Phases: COMPILE and EXECUTE

HAAG distinguishes two conceptual phases.

COMPILE

At this stage, the universal rules of the framework are adapted to the particular task:

Framework + Topic + Profile → Research Program.

This stage determines what is to be investigated, which agents are required, what verification procedures are feasible, which sources are needed, whether experiments are necessary, and what kinds of figures and final materials should be produced.

EXECUTE

The resulting research program is then executed:

Research Program

→ Research States

→ Human Decision Points

→ Verified Research Artifacts

→ Verified Publication

In other words:

Program → research states → human decisions → verified artifacts → verified publication
--

HAAG therefore distinguishes the specification of how research should be conducted from the actual conduct of that research.

4.3. Research as a Sequence of States

To formalize the process, HAAG introduces the concept of Research State.

At time or iteration t , the state is represented by a vector:

$$S_t = (T, H, Q, E, M, R, P, B, A, D, C, X, V, G)$$

This is one of the central constructs of the framework.

The idea is that HAAG should know the current position of the investigation at every stage.

It must track not only the text already written, but also:

the topic currently under investigation;

the hypotheses that remain active;

the questions that remain open;

the sources identified;

the model adopted;

the results obtained;

the branches rejected;

the decisions made by the human researcher;

the claims already supported by evidence;

the figures required;

the materials already prepared for the repository.

In this sense, HAAG maintains a memory of the research process.

4.4. T: Topic and Scope

The first component,

T

denotes the topic and current scope of the research.

The initial topic may be very broad:

“Information influence in society.”

After several iterations, it may be narrowed to:

“A model of semantic memory following successive information influences.”

Thus, T is not an immutable title.

It reflects the current scope of the investigation.

4.5. H: Hypotheses

The component

H

contains the set of current hypotheses:

$$\{H_1, H_2, \dots, H_k\}.$$

Hypotheses may have different statuses:

proposed, under test, supported, rejected, reformulated.

Importantly, HAAG should not delete a hypothesis simply because it has not been supported.

A rejected hypothesis remains part of the research history.

4.6. Q: Research Questions

The component

Q

contains the research questions.

For example:

Does the structural change persist after activation decays? Q2: *Does this change affect the response to the ne*

The distinction between H and Q matters.

Not every research question should immediately be turned into a strong hypothesis.

Sometimes it is more appropriate to formulate a question first, obtain a result, and only then develop a generalization.

4.7. E: Evidence and Sources

The component

E

denotes evidence and sources.

It may include:

scholarly articles;

monographs;

official documents;

datasets;

text corpora;

search results;

external facts needed to verify claims.

Crucially, E is not merely a bibliography.

A source must be linked to the particular claim it supports.

The preferred structure is therefore:

Claim $\leftrightarrow$ Evidence.

4.8. M: Methods and Models

The component

M

contains the methods and models adopted at the current stage.

These may include:

mathematical equations;

algorithms;

network models;

statistical procedures;

entity extraction rules;

experimental designs;

text analysis procedures;

evaluation methods.

It is important to retain not only the current model but also its version.

For example:

$\rightarrow M^{(2)} \rightarrow M^{(3)}.$

This makes it possible to identify the exact model version that produced a given result.

4.9. R: Results

The component

R

contains results actually obtained.

These must be observed or derived results, not expected ones.

They may include:

numerical values;

tables;

properties of a mathematical model;

simulation results;

statistical estimates;

findings from a real-world case;

null effects;

negative results.

Consequently,

$\neq$ expected results.

Conversely,

observed/derived/verified results.

4.10. P: Current Publication Draft

The component

P

contains the current publication draft.

In HAAG, however, it is deliberately treated as only one component of the Research State.

This emphasizes an important principle:

article text $\neq$ the research itself.

The text represents what has already been established in H, Q, E, M, and R.

When the research changes, the text must be rewritten.

The reverse is not acceptable: the desire to preserve an elegant paragraph must not force the system to retain a hypothesis that has not been supported.

4.11. B: Research Branches

The component

B

contains the branches of the research trajectory.

For example, an initial idea may give rise to three directions:

network model, B_2 : *agent model*, B_2 : agent model, B_3 : *hybrid model*. B_3 : hybrid model.

HAAG may investigate all three provisionally.

HUMAN may then select B_3 .

The remaining branches, B_1 and B_2 , need not be destroyed. They may be retained as rejected branches or directions for future work.

4.12. A: Active Agents

The component

A

indicates which agents are currently participating.

For example:

{Theorist, Falsifier, Source Auditor}.

At another stage:

{Experimentalist, Numerical Auditor}.

This allows HAAG to activate the configuration needed for the current task instead of using the entire agent society at once.

4.13. D: Decisions

The component

D

contains decisions made by agents and by the human researcher.

HUMAN decisions are particularly important.

For example:

D_17:

Use real data only as operationalization,
not as causal validation.

DECISION_BY:

HUMAN

Such a decision must remain in the history and influence subsequent statements of results.

This prevents the system from “forgetting” an established methodological limitation after several steps.

4.14. C: Claims and Their Provenance

The component

C

denotes claims and citation provenance.

For each significant claim, HAAG should know:

claim,support,status).

For example:

CLAIM:

Previous influence leaves a structural semantic trace.

SUPPORT:

EXP01

STATUS:

SUPPORTED

Or:

CLAIM:

Structural trace amplifies the next influence.

SUPPORT:

EXP01

STATUS:

UNSUPPORTED

This is a particularly important difference from ordinary text generation.

The model must not keep repeating the second claim merely because it fits the theory elegantly.

4.15. X: Research History

The component

X

contains TRACE/HISTORY: the history of transitions.

In its simplest form:

$S_0 \rightarrow S_1 \rightarrow S_2 \rightarrow \dots \rightarrow S_n$

For each transition, the following should be known:

what changed;
why it changed;
which agent initiated the change;
which source or result justifies it;
whether HUMAN made the decision.
Thus, X is effectively a record of the genesis of the scientific work.

4.16. V: Figures and Visual Materials

The component

V

contains the figure plan and visual assets.

These include:

the list of planned figures;
their scientific purpose;
the data they represent;
generation prompts;
completed images;
captions;
text preceding each figure;
explanations following each figure.

A figure is therefore not added at the last moment as decoration.

It is developed as part of the research architecture.

4.17. G: Reproducibility Package Status

The final component,

G

represents the status of the reproducibility/GitHub package.

For example:

{README, MODEL, PROMPTS, EXPERIMENTS, FIGURES, REFERENCES}.

Initially, this component may be almost empty.

It is progressively populated as the research proceeds.

This avoids having to reconstruct the research history retrospectively when creating a GitHub repository: the materials are accumulated while the investigation is carried out.

4.18. Transitions Between States

The HAAG architecture can now be described formally.

Let

S

denote the current Research State.

Following an action

u

the system moves to a new state:

$$S_{t+1} = \Phi(S_t, u_t)$$

where u_t may represent:

an agent action;

an experimental result;

a new source;

a mathematical derivation;

a HUMAN decision;

an audit result.

Thus, HAAG can be regarded as a controlled system of transitions between research states.

4.19. Not Every Transition Is Permitted

The existence of the transition operator Φ does not mean that the system may change its state arbitrarily.

The original framework establishes a fundamental requirement:

every significant transition must be associated with an agent action, verified evidence, a mathematical or logical rule, a numerical result, or HUMAN intervention.

In other words, for

$$S_t \rightarrow S_{t+1}$$

there must be a justification.

This can be written as:

$$\text{Reason}(S_t \rightarrow S_{t+1}) \neq \emptyset.$$

This is one of the foundations of HAAG traceability.

4.20. Example of Research State Evolution

Consider a simplified example.

Initially:

S_0 :

T = information memory

$H = \emptyset$

$Q = \emptyset$

E = several initial sources

$M = \emptyset$

$R = \emptyset$

$P = \emptyset$

After Explorer and Theorist have worked:

S_1 :

H_1 = previous influence changes the response to the next influence

H_2 = previous influence leaves a structural trace

Q_1 = are H_1 and H_2 manifestations of the same phenomenon?

After an experiment:

S_2 :

R:

a structural trace is present;

no appreciable amplification of the subsequent response is observed.

The Claims Ledger is then updated:

$C(H_2) = \text{SUPPORTED}; C(H_1) = \text{UNSUPPORTED}.$

Consequently, the Discussion section must no longer claim that structural memory automatically leads to functional amplification.

The publication draft P must therefore adapt to R and C:

$\rightarrow P,$

not the reverse.

4.21. Architectural Meaning of the HAAG State

Thus, the state

$S_{\square} = (T, H, Q, E, M, R, P, B, A, D, C, X, V, G)$

can be viewed as a digital representation of the current condition of a scientific investigation.

“The formalized Research State comprises three interrelated groups of components: the subject-related group, which defines the topic, hypotheses, and research questions T, H, Q;

the methodological group, which characterizes the evidence base and research methods E, M; and the results-related group, which records obtained results R. Other components support publication, branching, agent configuration, decisions, claim provenance, research history, visual artifacts, and reproducibility.”

This can be written as:

$$P_t = \Pi_{\text{publication}}(S_t)$$

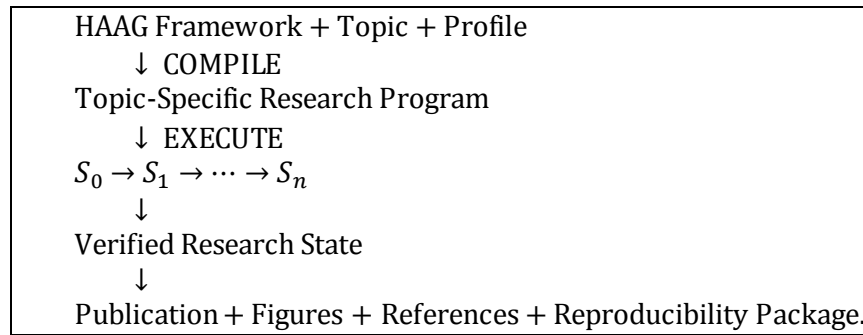
where $\Pi_{\text{publication}}$ is the operator that transforms a verified Research State into publication text.

This offers another important interpretation of HAAG:

HAAG does not generate the article directly.

The framework first constructs and progressively refines the state of a scientific investigation; only then are the article, figures, citations, and reproducibility package generated from the verified state.

The architectural logic of Chapter 4 can therefore be summarized as follows:



This moves HAAG beyond the notion of a “complex article-writing prompt” and toward a formal human–agent architecture for managing scientific research.

5. HAAG Primitive System

5.1. Purpose of the Primitive System

HAAG is designed not merely to produce the text of a scientific publication, but to describe and carry out a research process. The framework therefore requires its own set of basic constructs for specifying agents, tasks, conditions, transitions, verification procedures, and research outcomes.

In HAAG, these basic constructs are called primitives.

The complete basic set is as follows:

PROFILE STRUCTURE

AGENT ROLE TASK
 STATE TRACE
 FUNCTION CALL RETURN
 IF ELSE FOR
 LABEL GOTO STOP
 HUMAN
 FORK MERGE BACKTRACK
 SPAWN FREEZE RESUME KILL
 VERIFY ACCEPT REJECT
 WRITE REWRITE
 SOURCE CITE
 PREREGISTER RUN AUDIT
 FIGURE FIGURE_PROMPT
 PACKAGE EXPORT

HAAG primitives resemble programming-language operators in their external form. However, they are not intended to be executed by a processor in the traditional sense. Their interpreter is a large language model or a system of LLM agents.

HAAG can therefore be regarded as a human-readable, prompt-native domain-specific language for describing scientific research.

An important property of this language is that the same research program must be understandable to two participants:

Researcher ↔ HAAG Program ↔ LLM-Based Agentic System

The human researcher sees the logic of the investigation, while the agent system receives a sufficiently formalized description of what must be done, in what sequence, under which conditions, and with which checks.

The primitives can usefully be divided into functional groups:

research configuration;

definition of agents and tasks;

states and traceability;

functions and execution control;
branching of the research trajectory;
human involvement;
verification;
experimentation;
text generation;
working with sources;
figure generation;
preparation of the final research package.

Each primitive is examined individually below.

5.2. PROFILE — Publication and Research Profile

The primitive

PROFILE

defines the overall configuration of the intended scholarly product.

A typical form is:

PROFILE:

TYPE: ARTICLE

DOMAIN: Complex Networks / Information Dynamics

LANGUAGE: English

TARGET_LENGTH: AUTO

TARGET_VENUE: AUTO

CITATION_STYLE: AUTO

HUMAN_INTERACTION: ADAPTIVE

NOVELTY_LEVEL: HIGH

FIGURE_POLICY: LACONIC_Scientific

REPRODUCIBILITY_PACKAGE: AUTO

REFERENCE_WORKING_MODE: SCHOLAR_QUERY

PROFILE answers the question:

What scholarly product are we creating, and under which general rules should the research be conducted?

It influences the subsequent choice of structure, agents, depth of verification, literature-search procedures, number of figures, and composition of the final package.

PROFILE must not be used to record scientific results. It determines the research mode, not the conclusions.

5.3. STRUCTURE — Structure of the Scholarly Product

The primitive

STRUCTURE

defines the logical structure of the intended publication.

For example:

STRUCTURE:

1. Introduction
2. Methods
3. Results
4. Discussion
5. Conclusions

For an article, this may be IMRaD; for a monograph, a hierarchy of parts, chapters, and subsections; for conference abstracts, a compact structure without fully expanding every component.

Importantly, STRUCTURE specifies not only the future table of contents but also where different types of scientific results belong.

For example:

STRUCTURE:

METHODS:

definitions

assumptions

mathematical model

experiment design

RESULTS:

derived properties

numerical experiments

null results

Thus, STRUCTURE answers the question:

How should the research results be organized in the eventual publication?

Agents and Tasks

5.4. AGENT — Definition of a Specialized Agent

The primitive

AGENT

creates or describes a functional participant in the agent-based research process.

For example:

AGENT:

Falsifier

An agent should not be treated as a separate 'personality.' It is primarily a functional research role performed by an LLM in a particular context.

An agent definition is usually supplemented by the ROLE and TASK primitives.

5.5. ROLE — Persistent Function of an Agent

The primitive

ROLE

defines an agent's relatively stable function.

For example:

AGENT:

Falsifier

ROLE:

Attempt to reject hypotheses, claims,

models and interpretations.

ROLE answers the question:

What function does this agent perform in the research process?

For example, the Source Auditor verifies sources, whereas the Experimentalist designs experiments.

The role should not change every time the agent is invoked.

5.6. TASK — Specific Assignment for an Agent

Unlike ROLE, the primitive

TASK

defines a specific assignment in the current research state.

For example:

AGENT:

Falsifier

ROLE:

Critical hypothesis testing

TASK:

Try to reject H2.

Search for alternative explanations.

Determine what experimental result

would falsify H2.

Thus:

ROLE = Long-Term Function, TASK = Current Assignment.

A single agent may perform many different TASKs without changing its ROLE.

State and Traceability

5.7. STATE — Current Research State

The primitive

STATE

records the current Research State.

In HAAG, it is conceptually represented as

$St = (T, H, Q, E, M, R, P, B, A, D, C, X, V, G)$.

An example of a simplified representation is:

STATE:

TOPIC:

Semantic informational memory

ACTIVE HYPOTHESES:

H1, H2

REJECTED HYPOTHESES:

H3

MODEL:

M2

VERIFIED RESULTS:

R1, R2

OPEN QUESTIONS:

Q4, Q5

STATE answers the question:

What is currently known, accepted, rejected, under examination, and still unresolved?

This is fundamentally broader than the current version of the article.

5.8. TRACE — History of State Provenance

The primitive

TRACE

preserves the history of transitions.

For example:

TRACE:

T01:

Explorer proposed Q1.

T02:

Theorist formulated H1.

T03:

Contrarian proposed alternative H1a.

T04:

HUMAN selected comparison H1 vs H1a.

T05:

Experiment EXP01 preregistered.

T06:

EXP01 executed.

T07:

Numerical Auditor verified results.

T08:

H1 rejected under tested conditions.

TRACE answers the question:

Why did the current state take this particular form?

If STATE is a snapshot of the investigation, TRACE is its history.

Functions and Reusable Procedures

5.9. FUNCTION — Defining a Research Procedure

The primitive

FUNCTION

allows a recurring sequence of research operations to be encapsulated as a single procedure.

For example:

FUNCTION:

EvaluateHypothesis(H)

CALL:

Theorist

Contrarian

Falsifier

Source Auditor

VERIFY:

logical consistency

existing literature

falsifiability

RETURN:

HypothesisStatus

FUNCTION is useful when the same logic is applied to multiple hypotheses, models, sources, or results.

5.10. CALL — Invoking a Function or Agent

The primitive

CALL

activates a function or an agent.

For example:

CALL:

EvaluateHypothesis(H2)

or:

CALL:

Source Auditor

TASK:

Verify references supporting claim C7.

CALL does not imply automatic acceptance of the invoked agent's output. That output may require VERIFY, examination by another agent, or a HUMAN decision.

5.11. RETURN — Result of a Procedure

The primitive

RETURN

specifies the result that a function or agent returns to the research process.

For example:

RETURN:

HYPOTHESIS:

H2

STATUS:

PARTIALLY_SUPPORTED

UNCERTAINTY:

High

NEXT_ACTION:

Design discriminating experiment.

RETURN should be sufficiently structured to be incorporated into STATE or used by a subsequent operator.

Conditions and Loops

5.12. IF — Conditional Transition

The primitive

IF

specifies a condition for a transition.

For example:

IF:

H2 survives falsification

THEN:

PREREGISTER EXP02

Where possible, conditions should be based on explicit criteria.

Prefer:

IF:

P_D >= 1.20

rather than:

IF:

the effect looks strong

IF thus facilitates a transition from vague interpretation to a formalized research decision.

5.13. ELSE — Alternative Transition

The primitive

ELSE

specifies the action to take when an IF condition is not met.

For example:

IF:

Hypothesis H1 survives

PREREGISTER EXP01

ELSE:

REJECT H1

BACKTRACK

Through ELSE, HAAG does not assume that the desired outcome will necessarily be obtained.

A negative result must have its own valid research trajectory.

5.14. FOR — Repeating an Operation

The primitive

FOR

applies an operation to a set of objects.

For example:

FOR each major claim:

 VERIFY claim

 SOURCE supporting evidence

or:

FOR each figure:

 VERIFY scientific accuracy

 VERIFY labels

 VERIFY numerical values

FOR is particularly useful for systematic audits.

Research Program Control

5.15. LABEL — Naming a Stage

The primitive

LABEL

creates a named point in the research program.

For example:

LABEL :

MODEL_SELECTION

or:

LABEL :

SOURCE_VERIFICATION

This makes it possible to refer explicitly to a stage in subsequent transitions.

5.16. GOTO — Transition to a Defined Stage

The primitive

GOTO

transfers control to a LABEL.

For example:

IF :

model fails stability test

GOTO:

MODEL_SELECTION

GOTO is especially useful in iterative processes where the investigation must return to an earlier logical stage.

5.17. STOP — Terminating a Process or Branch

The primitive

STOP

terminates the execution of the program or a particular branch.

The reasons may vary:

STOP:

research objective achieved

or:

STOP:

hypothesis cannot be tested with available evidence

or:

STOP:

branch rejected

STOP must not automatically be interpreted as success. A justified termination without confirmation of the hypothesis is also an admissible outcome.

Human Involvement

5.18. HUMAN — A Scientifically Consequential Human Decision

HUMAN is one of the central HAAG primitives.

HUMAN

is used when the next choice substantially changes:

the research trajectory;

the method;

the experiment;

the evidence base;

the interpretation;

the permissible strength of a claim;

the form of the final publication.

Recommended structure:

HUMAN:

CONTEXT:

Two interpretations remain compatible

with the current evidence.

OPTIONS:

A:

Interpret the real-data case as validation.

B:

Interpret the real-data case only as
proof of operationalization.

CONSEQUENCES:

A:

Stronger claim, but requires causal evidence.

B:

Weaker claim, supported by available data.

QUESTION:

Which interpretation should determine
the next research state?

After the response, the decision is recorded:

DECISION:

B

DECISION_BY:

HUMAN

and incorporated into TRACE.

HUMAN should not be used for minor technical choices that the system can make without
changing the scientific substance.

Branching the Research Trajectory

5.19. FORK — Creating Alternative Branches

The primitive

FORK

creates several independent alternatives for developing the investigation.

For example:

FORK:

BRANCH A:

Network diffusion model

BRANCH B:

Semantic string model

BRANCH C:

Hybrid adaptive model

Each branch may have its own:

H, M, R, C.

FORK is particularly useful when there is insufficient evidence at an early stage to select a single alternative.

5.20. MERGE — Combining Compatible Branches

The primitive

MERGE

combines the results of two or more branches.

For example:

MERGE :

BRANCH A + BRANCH C

KEEP :

network representation from A

adaptive semantic dynamics from C

MERGE must not mechanically combine incompatible assumptions. Conceptual compatibility must be checked before merging.

5.21. BACKTRACK — Returning to an Earlier State

The primitive

BACKTRACK

returns the active trajectory to an earlier state.

For example:

BACKTRACK :

STATE S12

REASON :

Current model failed stability audit.

The unsuccessful branch is not deleted; it remains in TRACE.

Therefore:

BACKTRACK ≠ DELETE.

This is essential to scientific traceability.

Managing Active Processes

5.22. SPAWN — Creating a New Agent or Process

The primitive

SPAWN

activates a new agent process or auxiliary branch.

For example:

SPAWN:

Numerical Auditor

TASK:

Independently reproduce EXP01.

or:

SPAWN:

AlternativeModelBranch

SPAWN is used when the current process requires an independent, parallel investigation.

5.23. FREEZE — Fixing a State or Parameters

The primitive

FREEZE

prevents specified elements from being changed without an explicit record after a certain point.

For example:

FREEZE:

MODEL:

M3

PARAMETERS:

gamma = 1.5

alpha = 0.12

beta = 0.08

METRICS:

P_D

P_L

SUCCESS_THRESHOLD:

P_D >= 1.20

This is particularly important before an experiment.

After FREEZE, parameters must not be changed merely because the result failed to confirm expectations.

If a change is nevertheless necessary, it must create a new experiment version and be recorded in TRACE.

5.24. RESUME — Resuming a Suspended Process

The primitive

RESUME

reactivates a previously suspended branch.

For example:

RESUME :

BRANCH B

REASON:

New evidence makes the branch relevant again.

This allows promising directions that were temporarily set aside to be retained.

5.25. KILL — Terminating a Branch or Agent

The primitive

KILL

terminates an active process.

For example:

KILL :

BRANCH C

REASON:

Equivalent model already exists in literature.

KILL means:

Stop execution,

but not:

Erase the history.

The reason for termination must remain in TRACE.

Verification

5.26. VERIFY — Explicit Verification Procedure

The primitive

VERIFY

indicates that an object must be verified before use.

For example:

VERIFY:

OBJECT:

Claim C17

CHECK:

logical consistency

experimental support

source support

scope of conclusion

The objects to be verified may include:

$$\{H, M, R, C, E, P, V, G\}.$$

That is, hypotheses, models, results, claims, sources, text, figures, and the reproducibility package.

VERIFY must not be reduced to formal approval. Its outcome may be negative.

5.27. ACCEPT — Acceptance Following Verification

The primitive

ACCEPT

indicates that an object has passed a specified verification procedure.

For example:

ACCEPT:

OBJECT:

Claim C17

SCOPE:

Supported under EXP01 conditions.

It is important to record the scope of acceptance.

ACCEPT does not establish the absolute truth of a claim. It means that the claim is supported by the available evidence within specified limits.

5.28. REJECT — Rejection Following Verification

The primitive

REJECT

is used when an object fails verification.

For example:

REJECT:

OBJECT:

H3

REASON:

No measurable functional order effect
under preregistered EXP02 conditions.

A rejected object does not disappear.

It becomes part of the research history:

→ *REJECTED*.

A negative result forms part of the knowledge gained through the investigation.

Generating Scientific Text

5.29. WRITE — Generating Text from a Verified State

The primitive

WRITE

generates a publication section from the current research state.

For example:

WRITE:

SECTION:

Results

USE:

R12, R13, R14

CLAIMS:

C21, C22

DO NOT USE:

unverified preliminary values

WRITE must not generate new results.

Its function is:

verified research state $\rightarrow$ scientific text.
--

5.30. REWRITE — Updating Text After a Research Change

The primitive

REWRITE

revises previously written text when the evidence base, results, or interpretation have changed.

For example:

REWRITE:

SECTION:

Discussion

REASON:

EXP02 demonstrated structural

but not functional order dependence.

This is an important mechanism: the text must not “resist” new findings.

The HAAG rule is:

$R, C \rightarrow P, \quad P \nrightarrow R, C.$
--

That is, results and verified claims determine the text, not the other way around.

Sources and Citations

5.31. SOURCE — Finding and Verifying a Source

The primitive

SOURCE

initiates a source search for a specific scientific need.

For example:

SOURCE:

CLAIM:

Adaptive networks allow topology and
node dynamics to coevolve.

PREFERRED:

primary or peer-reviewed source

WORKING_QUERY:

"adaptive coevolutionary networks" Gross Blasius

SOURCE must specify not only the topic of the search but also the claim requiring support.

5.32. CITE — Linking a Verified Source to a Claim

The primitive

CITE

is executed only after a source has been verified.

For example:

CITE:

CLAIM:

C14

SOURCE:

REF05

STATUS:

VERIFIED

Thus:

Claim → SOURCE → VERIFY → CITE.

HAAG explicitly separates searching for a reference from citing it.

Experimental Cycle

5.33. PREREGISTER — Freezing an Experiment Specification Before Execution

The primitive

PREREGISTER

creates an immutable specification of an experiment before its results are known.

A typical record is:

PREREGISTER:

EXPERIMENT:

EXP01

HYPOTHESIS:

H2

MODEL:

M3

INITIAL CONDITIONS:

...

PARAMETERS:

...

CONTROLS:

C0, C1, C2

METRICS:

P_D, P_L

SUCCESS CRITERION:

$P_D \geq 1.20$

NULL REGION:

$P_D < 1.05$

STOP CONDITION:

...

RANDOM SEED:

2026

The purpose is to distinguish:

Experiment design

from

Knowledge of its outcome.

5.34. RUN — Actual Execution

The primitive

RUN

denotes the execution of an experiment that has already been specified.

For example:

RUN:

EXP01

USING:

Frozen specification EXP01-v1

RUN must produce:

raw results;

execution logs;

intermediate data;

information about errors;

the actual run parameters.

At this stage, the observed results must not be replaced by their desired interpretation.

5.35. AUDIT — Independent Verification of a Result

The primitive

AUDIT

checks whether the result of RUN is scientifically acceptable.

For example:

AUDIT:

EXPERIMENT:

EXP01

CHECK:

parameter freeze

initial conditions

metric calculations

numerical stability

reproducibility

consistency with preregistration

Result:

AUDIT STATUS:

PASSED

or:

AUDIT STATUS:

FAILED

REASON:

Metric calculation does not match
the preregistered definition.

Only after a successful audit can a numerical result be classified as VERIFIED.

Therefore:

PREREGISTER → FREEZE → RUN → AUDIT → VERIFIED RESULT.

Scientific Figures

5.36. FIGURE — Specification of a Scientific Figure

The primitive

FIGURE

defines a figure as a scientific object.

For each figure, it is advisable to specify:

FIGURE:

ID:

FIG01

TYPE:

CONCEPTUAL_MODEL

PURPOSE:

Show the interaction of activation,
semantics and structure.

CLAIM_SHOWN:

Information influence affects A, S and W.

DATA_SOURCE:

Theoretical model

TEXT_BEFORE:

...

CAPTION:

...

TEXT_AFTER:

...

SCIENTIFIC_CONSTRAINTS:

Do not imply unsupported causality.

Do not add variables absent from the model.

Thus, a figure in HAAG has at least four textual components:

Text Before the Figure + Figure Itself + Figure Caption + Post-Figure Explanation.

The figure title should not be embedded in the image itself unless explicitly required.

5.37. FIGURE_PROMPT — Prompt for Generating Scientific Graphics

FIGURE_PROMPT produces a ready-to-use request for an image-generation system.

Unlike an ordinary image prompt, it must contain not only a stylistic description but also the scientific message and constraints.

The basic form is:

FIGURE_PROMPT:

PURPOSE:

[what scientific idea must be shown]

CONTENT:

[objects, variables, relations]

LAYOUT:

[desired composition]

STYLE:

laconic scientific figure

white background

large readable labels

clean vector-like graphics

minimal visual decoration

publication quality

SCIENTIFIC_CONSTRAINTS:

show only relations supported by the model

use only verified numerical values

do not invent nodes, variables or arrows

do not imply unsupported causality

TEXT POLICY:

do not place figure number or caption

inside the image

avoid long explanatory text

OUTPUT:

high-resolution publication-ready image

For a conceptual figure:

FIGURE_PROMPT:

Create a laconic scientific diagram of the model.

Show:

Information Influence F

Activation A

Semantic State S

Structure W

Relations:

$F \rightarrow A$

$A \leftrightarrow S$

$S \leftrightarrow W$

$W \rightarrow A$

Use:

white background,

large readable labels,

simple arrows,

clean vector-like design,

minimal colors.

Do not:

add unsupported variables,

invent numerical values,

add decorative 3D elements,

place the figure caption inside the image.

For an experimental result, the following must be added:

Use only numerically verified values.

Do not reconstruct missing values.

For real-data operationalization:

Do not draw causal arrows between

time slices unless causality was actually established.

Thus, FIGURE_PROMPT serves as a kind of contract between the scientific model and the image generator.

Assembling the Final Product

5.38. PACKAGE — Assembling the Research Package

The primitive

PACKAGE

collects verified artifacts into a single deliverable.

For example:

PACKAGE:

MANUSCRIPT:

article.docx

FIGURES:

FIG01

FIG02

FIG03

REFERENCES:

verified_references.md

EXPERIMENTS:

EXP00

EXP01

EXP02

PROMPTS:

research_prompts/

figure_prompts/

REPRODUCIBILITY:

README.md

MODEL.md

REPRODUCIBILITY.md

TRACE:

research_trace.md

PACKAGE should include only results whose status is known.

For example, provisional results may be retained in the repository, but they must be explicitly marked as provisional.

5.39. EXPORT — Creating External Representations

The final primitive

EXPORT

converts the research package into formats suitable for transfer, publication, or archiving.

For example:

EXPORT:

DOCX

PDF

Markdown

PNG

SVG

GitHub Repository

ZIP

It is important to distinguish:

PACKAGE = Logical Assembly of Materials,

as distinct from

EXPORT = Creation of Their Physical Representations.

5.40. Relationships Among the Primitives

The primitives should not be considered in isolation. Their value emerges through composition.

For example, a hypothesis-testing cycle may look like this:

AGENT: Theorist

TASK: Formulate H1

CALL: Falsifier

TASK: Try to reject H1

VERIFY: H1

IF H1 survives:

CALL: Experimentalist

PREREGISTER: EXP01

FREEZE: EXP01

RUN: EXP01

AUDIT: EXP01

IF AUDIT passed:

ACCEPT: verified result
ELSE:
REJECT: numerical result
ELSE:
REJECT: H1
BACKTRACK
A source-verification cycle:
FOR each major claim:
SOURCE:
Find supporting source
CALL:
Source Auditor
VERIFY:
bibliographic metadata
claim-source relation
IF verified:
ACCEPT source
CITE source
ELSE:
REJECT source
A figure-creation cycle:
FIGURE:
Define scientific message
FIGURE_PROMPT:
Generate laconic scientific prompt
CALL:
Figure Designer
VERIFY:
scientific accuracy
labels
arrows
numerical values
IF verified:
ACCEPT figure
ELSE:

REWRITE FIGURE_PROMPT

And completion of the work:

CALL:

Reviewer

VERIFY:

manuscript

references

figures

results

reproducibility

IF major problems:

BACKTRACK

ELSE:

PACKAGE

EXPORT

STOP

Thus, the primitives constitute not merely a vocabulary of commands but a compositional language for research procedures.

5.41. Functional Classification of the Primitives

The complete set can be summarized as follows:

Function	Primitives
Configuration	PROFILE, STRUCTURE
Agents	AGENT, ROLE, TASK
State and Memory	STATE, TRACE
Functions	FUNCTION, CALL, RETURN
Conditional Logic	IF, ELSE, FOR
Execution Control	LABEL, GOTO, STOP
Human	HUMAN
Branching	FORK, MERGE, BACKTRACK
Life Cycle	SPAWN, FREEZE, RESUME, KILL
Verification	VERIFY, ACCEPT, REJECT
Text	WRITE, REWRITE
Sources	SOURCE, CITE
Experiment	PREREGISTER, RUN, AUDIT
Figures	FIGURE, FIGURE_PROMPT
Final Product	PACKAGE, EXPORT

This table shows that the primitive system covers virtually the entire research life cycle, from configuration to publication.

5.42. HAAG as a Domain-Specific Language for Research

Having examined the primitives in detail, it becomes clear that HAAG can be interpreted as more than a prompt or even a framework.

It has the properties of a domain-specific language for describing scientific processes:

$$\boxed{\text{HAAG DSL} = \text{Vocabulary} + \text{Control Flow} + \text{Research State} + \text{Human Decisions} + \text{Verification Rules.}}$$

At the same time, its syntax intentionally remains close to natural language.

This allows both a human and an LLM to read the same HAAG program.

6. HAAG Agent Society

6.1. From a General-Purpose LLM to Functionally Distributed Research

A language of primitives answers the question of how to describe a research process. A second question remains: who performs the individual research functions?

HAAG uses the concept of an Agent Society.

Instead of a single general-purpose agent, it introduces a set of specialized functional roles:

$$\mathcal{A} = \{A_E, A_T, A_C, A_F, A_X, A_N, A_S, A_G, A_R, A_Y, A_P, A_O\}.$$

These need not be twelve physically separate models. One LLM may perform different roles in succession. What matters is functional separation.

The reason is straightforward: creating a scientific claim and evaluating it require different perspectives.

An agent that has just constructed a theory should not be the sole judge of its validity.

Consequently, a fundamental HAAG principle is:

$$\boxed{\text{Generation} \neq \text{Verification.}}$$

6.2. Explorer — Identifying Research Problems

Explorer searches for:

gaps in knowledge;

contradictions;

anomalies;

underexplored relationships;

new problem formulations;

questions that can be operationalized and tested.

Its central question is:

What remains insufficiently explained in this subject area?

Formally:

$$\{Q_1, Q_2, \dots, Q_n\}.$$

Explorer must not declare an identified gap to be novel without checking the literature.
Its output is a candidate research problem, not automatically established scientific novelty.

6.3. Theorist — Formulating Concepts, Hypotheses, and Models

Theorist transforms a problem into a formalized construct.

It:

defines concepts;
introduces variables;
formulates hypotheses;
constructs mathematical models;
specifies assumptions;
derives consequences;
identifies testable model properties.

Schematically:

$$Q \rightarrow \text{Theorist} \rightarrow (H, M).$$

Theorist must clearly distinguish:

definition, assumption, hypothesis, derived result.

This prevents a model assumption from later being presented as an established result.

6.4. Contrarian — Seeking Competing Interpretations

Contrarian deliberately seeks an alternative explanation.

If Theorist proposes:

$$H: A \Rightarrow B,$$

Contrarian must ask:

Could B arise because of C rather than A?

Its function is:

$$\{H, H'_1, H'_2, \dots\}.$$

This agent is particularly useful before experiment design because it helps identify necessary control scenarios.

6.5. Falsifier — Actively Attempting to Reject a Hypothesis

Falsifier has a more stringent function than Contrarian.

It seeks to establish:

Under what conditions should this hypothesis be judged false or insufficiently supported?

Falsifier searches for:
counterexamples;
logical contradictions;
limiting regimes;
known findings incompatible with the hypothesis;
control experiments;
falsification criteria.
Its task is not necessarily to destroy the hypothesis.
Its task is to make the hypothesis testable.

6.6. Experimentalist — Designing Experiments

Experimentalist translates a hypothesis into an experimental protocol.
It specifies:

$$\{H, M, I, P, C, \mu, \Theta, S\},$$

where, by convention:

H — hypothesis;
M — model;
I — initial conditions;
P — parameters;
C — controls;
 μ — metrics;
 Θ — criteria;
S — stopping conditions.

Experimentalist must not select a design that guarantees the desired result.

Its task is to construct a discriminating test capable of distinguishing competing explanations.

6.7. Numerical Auditor — Numerical Verification

Numerical Auditor checks numerical findings independently of their theoretical appeal.

It checks:

whether implementation matches the model;
recorded parameters;
initial conditions;
random seeds;
metric definitions;
calculation correctness;

stability;
reproducibility;
consistency between RUN and PREREGISTER.

Its function is:

$R_{\text{raw}} \rightarrow \text{Numerical Auditor} \rightarrow R_{\text{verified}}$.

HAAG requires an explicit distinction between:

$\boxed{PROVISIONAL \neq VERIFIED_NUMERICALLY.}$

A number must not enter the final Results merely because it appeared once in the dialogue.

6.8. Source Auditor — Verifying the Evidence Base

Source Auditor is responsible for scholarly literature and external sources.

For each source, it checks:

existence;
authors;
title;
year;
publication venue;
DOI or another identifier;
whether the source is primary or authoritative;
relevance to the specific claim.

The last criterion is decisive.

An accurate bibliography does not automatically imply accurate citation.

It is necessary to check:

$\boxed{\text{Claim} \leftrightarrow \text{Source}.}$

6.9. Figure Designer — Designing Scientific Figures

Figure Designer converts scientific content into a concise visual form.

It must determine:

what the figure actually communicates;
which claim it illustrates;
which objects are necessary;
which relationships are supported;
which numerical values may be shown;
what the figure must not imply.

A `FIGURE_PROMPT` is then prepared.

The basic rule is:

$$\boxed{\text{Scientific fidelity} > \text{Visual attractiveness.}}$$

Figure Designer must not add causal arrows, variables, numbers, or objects solely to make the composition attractive.

6.10. Reviewer — Internal Scientific Peer Review

Reviewer acts as a skeptical external reviewer.

It checks:

- whether the research problem is clearly stated;
- whether the claimed novelty is substantiated;
- whether the method fits the task;
- whether controls are sufficient;
- whether the results follow from the experiment;
- whether the conclusions overreach;
- whether correlation is distinguished from causation;
- whether negative results are considered;
- whether limitations are disclosed;
- whether the research is reproducible.

Reviewer should not seek to improve the impression made by the paper; it should look for reasons the paper might be rejected.

6.11. Synthesizer — Maintaining a Consistent Research State

The outputs of different agents may contradict one another.

Synthesizer must integrate them into a consistent STATE without concealing uncertainty.

Its function is:

$$\{A_1, A_2, \dots, A_n\} \rightarrow S_{t+1}.$$

It:

- combines compatible results;
- records contradictions;
- maintains consistent terminology;
- synchronizes the Claims Ledger;
- prevents the text from contradicting verified results.

Its principle is:

$$\boxed{\text{Coherence without hiding uncertainty.}}$$

6.12. Repository Curator — Preparing Reproducibility Materials

Repository Curator assembles a structured research package.

For example:

README.md

MODEL.md

REPRODUCIBILITY.md

prompts/

experiments/

figures/

references/

real_case/

docs/

It:

organizes materials;

preserves prompts;

separates verified and provisional results;

includes negative results;

documents experiments;

prepares the README;

checks that the repository agrees with the article.

In HAAG, the repository is not an archive of “supplementary files,” but part of the research output.

6.13. Research Orchestrator — Research Coordinator

Research Orchestrator manages execution of the HAAG program.

It determines:

which agent is needed now;

which task to assign;

whether a parallel branch is needed;

whether the evidence is sufficient;

whether it is time to conduct an experiment;

whether VERIFY is required;

whether HUMAN must be consulted;

whether the process can proceed to WRITE;

whether BACKTRACK is necessary.

Formally:

$$O(S_t) \rightarrow (A_t, u_t) \rightarrow S_{t+1},$$

where O denotes the Orchestrator, A_t the required agent configuration, and u_t the next action.

Its central question is:

What scientifically justified research transition should be performed next?

6.14. Agent Society as a System of Mutual Checks

Agent Society is more than a way to distribute a large workload.

It implements a system of checks and balances.

For example:

Explorer $\rightarrow$ Theorist $\rightarrow$ Contrarian $\rightarrow$ Falsifier,

then:

Experimentalist $\rightarrow$ Numerical Auditor,

for sources:

SOURCE $\rightarrow$ Source Auditor,

for figures:

Figure Designer $\rightarrow$ VERIFY,

for the article:

Synthesizer $\rightarrow$ Reviewer.

The general rule follows:

The entity that creates a scientific artifact should not be the sole entity responsible for verifying it.

This applies to hypotheses, experiments, results, sources, figures, and text.

6.15. Dynamic Formation of Agent Groups

Not all agents need to operate simultaneously.

Different stages require different configurations.

At the beginning:

{Explorer, Theorist, Contrarian}.

During hypothesis evaluation:

{Falsifier, Source Auditor}.

Before an experiment:

{Experimentalist}.

After execution:

{Numerical Auditor, Synthesizer}.

At the publication stage:

{Synthesizer, Figure Designer, Reviewer, Repository Curator}.

Thus, Agent Society is a dynamic configuration of functional roles that changes with the Research State.

6.16. HUMAN Is Not an Agent of Agent Society

It is essential to distinguish the human researcher from the Agent Society.

HUMAN is not simply another:

$A_{(n+1)}$.

LLM agents perform functions within the research program.

The human controls the trajectory itself when a scientifically consequential choice arises.

The architecture therefore takes the form:

HUMAN $\leftrightarrow$ Research Orchestrator $\leftrightarrow$ Agent Society

and all these components interact with

S_t .

Agent Society explores the space of possibilities. Orchestrator organizes transitions. HUMAN determines the direction at points where an automatic decision would have methodological consequences.

6.17. General Functional Structure of Agent Society

HAAG agents can be grouped by their principal functions:

Exploration and generation: Explorer, Theorist; Alternatives and falsification: Contrarian, Falsifier; Experimentation: Experimentalist; Verification: Numerical Auditor, Source Auditor; Presentation: Figure Designer; Critical assessment: Reviewer; Integration: Synthesizer; Reproducibility: Repository Curator; Coordination: Research Orchestrator.

Human scientific oversight remains above this system.

6.18. General Execution Scheme

Once the primitives and agents have been specified, the full cycle can be represented as follows:

TOPIC + PROFILE $\downarrow$ STATE $S_0 \downarrow$ Explorer / Theorist $\downarrow$ Contrarian / Falsifier $\downarrow$ HUMAN, when a strategic decision is required $\downarrow$ Experimentalist $\downarrow$ PREREGISTER $\rightarrow$ FREEZE $\rightarrow$ RUN $\rightarrow$

AUDIT ↓ Numerical Auditor / Source Auditor ↓ Synthesizer ↓ WRITE / FIGURE / CITE ↓
Reviewer ↓ Repository Curator ↓ PACKAGE → EXPORT

This scheme is not, however, a rigid pipeline. At any scientifically justified stage, the following remain possible:

FORK, BACKTRACK, REJECT, REWRITE, HUMAN.

HAAG therefore implements not linear document generation but the controlled evolution of the Research State:

$$\boxed{S_0 \xrightarrow{u_0} S_1 \xrightarrow{u_1} S_2 \rightarrow \dots \rightarrow S_n,}$$

where each significant transition u_t must be justified by an agent action, a verified source, a mathematical or logical rule, a numerical result, or a HUMAN decision.

The detailed system of primitives makes this trajectory describable and human-readable, while Agent Society makes it executable. Together, they transform HAAG from an extensive article-generation prompt into a prompt-native language and architecture for managing human-agent scientific research.

7. HUMAN as a Full-Fledged Operator of Scientific Research

7.1. The Role of the Human in the HAAG Architecture

One of HAAG's fundamental features is that the human is not treated as a passive user who supplies an initial topic and then simply receives a completed article. Nor is the human merely an external observer consulted only when the system encounters a difficulty.

In HAAG, the human is incorporated directly into the logic of the research process through the dedicated primitive

HUMAN

and acts as a full-fledged operator of transitions between research states.

If the current research state is denoted by

S_t ,

a human decision can directly determine the next state:

$$\boxed{S_t \xrightarrow{\text{HUMAN}} S_{t+1}.}$$

This differs substantially from an approach in which the LLM makes all decisions independently and the human merely edits the resulting text.

In HAAG, a human decision can change not only the text P , but virtually any component of the research state:

H, Q, E, M, B, D, C, V, G.

For example, the human may reject a hypothesis, change the mathematical model, select a different evidence base, create a new research branch, or limit the strength of a conclusion.

HUMAN is therefore not an interface function but part of HAAG's formal architecture.

7.2. Why a Fully Autonomous Trajectory Is Undesirable

An LLM-agent system can effectively:

- generate alternatives;
- analyze the literature;
- construct mathematical formulations;
- propose experiments;
- perform calculations;
- check consistency;
- produce text;
- create figures.

It does not follow, however, that the system should make every decision autonomously.

Scientific research regularly presents situations in which several methodologically admissible trajectories exist:

B1, B2,...,Bn.

The choice between them may determine the entire subsequent content of the study.

For example:

Network Model or Agent-Based Model, Real Data or Synthetic Experiment.

At such points, an automatic choice may quietly become a methodological decision made by the system instead of the author.

HAAG makes these decision points explicit.

7.3. Standard Structure of HUMAN

A request to the human should be structured and concise. It should not transfer the agent system's entire workload to the researcher.

The basic template is:

HUMAN:

CONTEXT:

[brief current scientific state]

PROBLEM:

[why a meaningful choice exists]

OPTIONS:

A: ...

B: ...

C: ...

QUESTION:

[one clear scientific question]

ALLOW:

FREE_TEXT

Each element of this template has a distinct function.

7.4. CONTEXT — Minimal Scientific Context

The field

CONTEXT

should briefly describe the current research state needed to make a decision.

For example:

CONTEXT:

EXP01 shows a persistent structural change after activation has decayed.

No substantial amplification of the second influence was observed.

The context should not turn into a multi-page report.

Its purpose is to answer the question:

What do we already know at the moment of decision?

In terms of Research State:

$\text{CONTEXT} = \Pi_{\text{relevant}}(S_{\text{?}})$,

that is, the relevant projection of the current state $S_{\text{?}}$.

7.5. PROBLEM — Why a Human Decision Is Needed

The field

PROBLEM

explains why the system should not automatically continue the research.

For example:

PROBLEM:

The structural-memory hypothesis is supported, but the functional-preconditioning hypothesis is not supported under the tested conditions.

The article can now follow two scientifically different interpretations.

The point is not simply to report uncertainty, but to explain the scientific significance of the choice.

7.6. OPTIONS — Substantive Alternatives

The field

OPTIONS

contains alternatives already identified by the agent system.

For example:

OPTIONS:

A:

Treat structural memory as the main result
and reformulate the theory accordingly.

B:

Modify the model and test whether another
parameter regime produces functional memory.

C:

Keep both ideas but explicitly distinguish
structural and functional memory.

The alternatives must represent genuinely different research decisions, not cosmetic rephrasings of a single option.

Where possible, their consequences should also be indicated:

A:

More conservative theory;
no additional experiment required.

B:

Potentially stronger result;
requires a new preregistered experiment.

C:

Broader conceptual framework;
requires explicit separation of memory types.

HUMAN then receives not merely a menu, but information about the implications of each choice.

7.7. QUESTION — One Clear Scientific Question

After presenting the context, problem, and alternatives, the system should ask one clear question.

For example:

QUESTION:

Which interpretation should define

the next version of the model?

It is undesirable to ask the human five independent questions at once.

If several decisions are required, it is preferable to use a sequence:

$$\rightarrow S_{t+1} \rightarrow HUMAN_2.$$

This reduces cognitive load and makes TRACE less ambiguous.

7.8. FREE_TEXT — The Human Is Not Limited to Proposed Alternatives

The field

ALLOW:

FREE_TEXT

is of fundamental importance.

Options A, B, and C are proposals from the agent system, not the boundaries of the decision space.

The human may formulate:

An alternative that the system did not propose at all.

This is why HUMAN must not be reduced to a SELECT(A,B,C) mechanism.

7.9. Permitted HUMAN Actions

The original HAAG specification defines a broad set of possible human responses.

SELECT

Select one of the proposed alternatives:

SELECT:

B

REJECT

Reject a proposed alternative:

REJECT:

A

Rejection does not necessarily mean choosing another ready-made option.

MODIFY

Modify a proposed decision:

MODIFY:

Use option B,

but keep the original primary metric.

COMBINE

Combine several alternatives:

COMBINE:

A + C

Synthesizer must then check whether they are methodologically compatible.

EXPAND

Expand the scope of the problem:

EXPAND:

Include semantic and structural memory
as separate phenomena.

NARROW

Narrow the scope of the research:

NARROW:

Study only structural memory
in the present article.

REFORMULATE

Reformulate the problem or hypothesis:

REFORMULATE:

Do not ask whether memory amplifies
the next influence.

Ask whether prior influence changes
the system's response in any measurable way.

RETURN

Return to an earlier stage:

RETURN:

MODEL_DEFINITION

This action naturally combines with BACKTRACK.

CREATE_BRANCH

Create a new research branch:

CREATE_BRANCH:

B4:

Structural-memory-only model

ABANDON_BRANCH

Stop working on a particular branch:

ABANDON_BRANCH:

B2

Its history is nevertheless retained.

NEW_DIRECTION

The human may propose a direction that was absent from the agent-generated alternatives:

NEW_DIRECTION:

Separate semantic memory from
network-topological memory and
model them independently.

This is particularly important: HUMAN does not merely select among the outcomes of machine exploration; it can expand the research space itself.

7.10. A HUMAN Decision Must Change STATE

After the human responds, simply continuing the dialogue is insufficient.

The decision must be incorporated into the Research State.

For example:

DECISION_ID:

D17

DECISION_BY:

HUMAN

DECISION:

Use the real-world case only as
proof of operationalization.

CONSEQUENCE:

Do not claim causal empirical validation.

After this:

$$\rightarrow D_{t+1},$$

and the corresponding limitation must also be reflected in:

C, P, V, G.

Thus, the decision must affect the article's claims, text, figures, and repository materials.

7.11. When HUMAN Is Mandatory

HAAG identifies a class of situations in which automatic selection is undesirable.

Choosing Between Genuinely Different Hypotheses

For example:

Memory is structural. H2: Memory is functional. H2: Memory is functional.

If both directions are scientifically admissible, the choice may determine the subsequent architecture of the article.

Changing the Mathematical Model

A transition

$\rightarrow M_2$

may change the meaning of variables, parameters, and experiments.

A substantial change in the model therefore requires HUMAN.

Choosing Between Real and Synthetic Data

The decision

real data vs synthetic data

affects the nature of the evidence base and the strength of permissible conclusions.

Changing an Experiment After Observing the Result

This is a particularly important case.

If an experiment produces an undesirable result and the system proposes changing parameters, HUMAN must be informed that the change is being made after observing the result.

It is not permissible to quietly transform

EXP01EXP01

into a different experiment while continuing to call it EXP01.

Changing the Primary Metric

If the original primary metric was

$M_1, M_1,$

and another metric

M_2, M_2

appears more attractive after the experiment, it cannot automatically be declared the primary metric.

This must be an explicit decision.

Changing the Success Threshold

It is particularly dangerous to change

$\theta_{\text{success}}, \theta_{\text{success}}$

after obtaining the result.

For example, if the criterion was

$P_D \geq 1.20, P_D \geq 1.20,$

and the observed result was

$P_D = 1.08, P_D = 1.08,$

one cannot then automatically reset the criterion to:

$P_D \geq 1.05, P_D \geq 1.05.$

Making a Strong Novelty Claim

The statement

“this approach is proposed for the first time”

carries much greater risk than:

“the literature search conducted did not identify studies combining these components in this formulation.”

A strong novelty claim must therefore not only be verified by Source Auditor, but also consciously accepted by the author.

Interpreting an Ambiguous Negative Result

A negative result may mean:

the hypothesis is incorrect;

the effect is absent only in the regime investigated;

the metric is insensitive;

the experiment is insufficiently discriminating;

the model has limitations.

If the data cannot distinguish these interpretations unambiguously, the final formulation requires HUMAN.

Final Synthesis

Before the final version of the publication is prepared, HUMAN should confirm that:

the research trajectory has been completed;

substantive alternatives have been considered;

the results have been interpreted correctly;

the claimed level of novelty is acceptable;

the process may proceed to the final PACKAGE.

7.12. When HUMAN Need Not Be Invoked

Excessive questioning of the human makes the system practically unusable.

HUMAN should therefore not be invoked for:

changing indentation;

formatting a table;

numbering figures;

correcting an obvious typographical error;

technical normalization of notation;

choosing harmless numerical-integrator parameters;

correcting an obvious internal inconsistency;

reformatting the bibliography to match the journal's style.

The principal criterion is:

Can this decision change the scientific substance or the strength of the conclusion?

If not, Orchestrator may perform the action automatically.

7.13. Adaptive Frequency of HUMAN

Thus,

HUMAN_INTERACTION:

ADAPTIVE

does not mean “ask the human frequently” or “almost never ask the human.”

It means:

Frequency of HUMAN Involvement $\propto$ Scientific Significance of the Decision.

The more strongly a decision affects hypotheses, methods, evidence, or conclusions, the greater the reason to refer it to the human.

8. Experiment Preregistration and Freezing

8.1. Why HAAG Needs a Preregistration Mechanism

An LLM system poses a particular risk: it can adapt remarkably easily to results that have already been obtained.

If an experiment fails to support a hypothesis, the system may propose:

changing a parameter;

choosing another threshold;

changing the metric;

removing an inconvenient control;

extending the duration;

changing the initial state;

using another random seed.

Each of these changes may seem entirely reasonable when considered in isolation.

However, if they are made after the result has been observed, there is a risk that a scientific experiment will turn into a procedure for searching for the desired outcome.

HAAG therefore introduces a strict cycle:

PREREGISTER $\rightarrow$ FREEZE $\rightarrow$ RUN $\rightarrow$ AUDIT.

8.2. PREREGISTER — Specification Before Obtaining Results

Before a substantive numerical experiment, the following is created:

PREREGISTER

A typical structure is:

PREREGISTER:

EXPERIMENT_ID
HYPOTHESIS
MODEL_VERSION
INITIAL_STATE
PARAMETERS
INPUT / INFLUENCE
CONTROLS
PRIMARY_METRICS
SECONDARY_METRICS
RELAXATION / STOP CONDITIONS
DECISION_THRESHOLDS
RANDOM_SEED when applicable
FAILURE_CRITERIA

The purpose is to establish, before running the experiment, an answer to the question:
What exactly are we testing, and what will count as a result?

8.3. EXPERIMENT_ID — Experiment Identifier

Each experiment receives a unique identifier:

EXPERIMENT_ID:

EXP01

If the specification needs to be changed substantially after execution, a new version is created:

EXP01-v2

or a new experiment:

EXP02

This prevents several different procedures from being concealed under a single name.

8.4. HYPOTHESIS — What Exactly Is Being Tested

The field

HYPOTHESIS

links the experiment to a particular hypothesis.

For example:

HYPOTHESIS:

H2:

A previous influence produces
functional preconditioning of

the response to a later influence.

An experiment without an explicit hypothesis may be useful as an exploratory experiment, but it must not retrospectively be presented as a confirmatory test of a hypothesis formulated after the results were observed.

8.5. MODEL_VERSION — Which Model Is Being Tested

The field

MODEL_VERSION

records the version of the mathematical model.

For example:

MODEL_VERSION:

M3.1

This is essential because even a minor change in an equation can change the interpretation of a result.

It must be possible to establish the relationship:

$R_i \rightarrow M_i$.

In other words, every numerical result must be linked to a particular model version.

8.6. INITIAL_STATE — Initial State

The field

INITIAL_STATE

records the initial conditions.

For example:

$A(0) = 0, S(0) = S^{(0)}, W(0) = W^{(0)}$.

The initial state is part of the experiment, not a minor technical detail.

Changing it effectively changes the experimental design.

8.7. PARAMETERS — Model Parameters

All substantive parameters must be defined before execution.

For example:

PARAMETERS:

gamma = 1.50

mu = 0.15

lambda = 0.03

alpha = 0.12

beta = 0.08

delta = 0.02

$\sigma = 1.00$

It is also advisable to specify where the parameters came from:

PARAMETER_ORIGIN:

gamma:

selected by stability pretest EXP00

alpha:

fixed theoretical parameter

sigma:

normalization choice

This helps distinguish a justified choice from arbitrary parameter tuning.

8.8. INPUT / INFLUENCE — Experimental Influence

The external influence must be specified precisely.

For example:

INPUT:

F1:

nodes = v5, v6, v7, v8

amplitude = 0.50

duration = 5

For the second signal:

F2:

nodes = v9, v10

amplitude = 0.50

duration = 5

Thus, the influence becomes part of the preregistered specification rather than a parameter that can be changed arbitrarily after the result is examined.

8.9. CONTROLS — Control Scenarios

Control conditions must be specified before execution.

For example:

CONTROLS:

C0:

No previous influence

C1:

Previous influence with adaptive dynamics

C2:

Previous influence with frozen structure

C3:

Shuffled topology control

The control answers the question:

What exactly is the main result being compared against?

Without controls, it is difficult to determine which mechanism produced the observed effect.

8.10. PRIMARY_METRICS — Primary Metrics

Primary metrics are the quantities used to make the main decision concerning the hypothesis.

For example:

$$\frac{D_3(F_2|F_1)}{D_3(F_2|\emptyset)}.$$

In the preregistration:

PRIMARY_METRICS:

P_D

The primary metric must be specified before the result is obtained.

After execution, one must not select whichever of twenty metrics happened to produce the largest effect and present it as the primary metric.

8.11. SECONDARY_METRICS — Secondary Characteristics

Secondary metrics can help explain a mechanism or characterize other properties.

For example:

SECONDARY_METRICS:

P_L

Delta_S

Delta_W

max_activation

relaxation_time

They may be important, but it must remain clear that they were not the primary criterion for supporting the hypothesis.

8.12. RELAXATION / STOP CONDITIONS — Termination Conditions

The experiment must not be terminated at an arbitrarily selected moment.

For example:

$$\max_i a_i < 10^{-3}$$

and

$$\|\dot{A}\|_2 < 10^{-4}.$$

Then:

RELAXATION CONDITIONS:

$$\max(a_i) < 1e-3$$

AND

$$\text{norm}(dA/dt) < 1e-4$$

A maximum time can also be specified:

STOP CONDITION:

$$t \geq T_{\text{max}}$$

This prevents selection of a “convenient” measurement time after the trajectory has been inspected.

8.13. DECISION_THRESHOLDS — Interpretation Thresholds

Before execution, it is advisable to define not only the metric but also the rules for interpreting it.

For example:

$$P_D \geq 1.20$$

— noticeable preconditioning;

$$1.05 \leq P_D < 1.20$$

— weak effect;

$$P_D < 1.05$$

— the hypothesis is unsupported under the specified criterion.

Then:

DECISION_THRESHOLDS:

$$P_D \geq 1.20:$$

noticeable effect

$$1.05 \leq P_D < 1.20:$$

weak effect

$$P_D < 1.05:$$

hypothesis unsupported

This substantially reduces the opportunity to adjust the interpretation after seeing the numerical value.

8.14. RANDOM_SEED — Controlling Randomness

For stochastic procedures, the following must be recorded:

RANDOM_SEED

For example:

RANDOM_SEED:

2026

If one seed is insufficient for a reliable conclusion, the preregistration should specify multiple runs:

$$\{s_1, s_2, \dots, s_n\}.$$

The purpose of a seed is reproducibility, not obtaining an attractive result.

8.15. FAILURE_CRITERIA — When an Experiment Is Invalid

It is very important to specify in advance not only the criteria for supporting the hypothesis but also the criteria for technical failure of the experiment itself.

For example:

FAILURE_CRITERIA:

numerical divergence

violation of stability conditions

undefined metric

control construction failure

parameter mismatch

implementation inconsistent with model

In this case, the result must be interpreted neither as support for nor as refutation of the hypothesis.

The appropriate status is:

EXPERIMENT INVALID.

8.16. FREEZE — Freezing the Specification

Once preregistration is complete, the following is executed:

FREEZE:

EXPERIMENT_SPECIFICATION

This means that the following have been fixed:

$$\mathcal{E} = (H, M, I, P, F, C, \mu, \theta, S, \xi),$$

where:

H — hypothesis;

M — model;

I — initial state;

P — parameters;

F — influence;

C — controls;

μ — metrics;

Θ — thresholds;

S — stopping conditions;

ξ — randomness parameters.

After FREEZE, this exact specification must be passed to RUN.

8.17. The Main Rule After RUN

Once execution begins, a strict rule applies:

Do not modify the experiment to obtain the desired result.

In HAAG notation:

RULE:

After RUN begins, do not tune
parameters, thresholds, targets,
controls, or metrics to obtain
a desired result.

This applies to:

parameters;

thresholds;

primary metrics;

controls;

initial conditions;

target nodes;

duration of influence;

relaxation criteria;

interpretation procedures.

8.18. What to Do If a Model Defect Is Discovered After Execution Begins

Preregistration does not mean that errors cannot be corrected.

If a genuine defect is discovered after RUN begins:

IF:

a model defect or missing specification
is discovered

the system must:

THEN:

STOP current experiment
record failure in TRACE
HUMAN if scientifically consequential
create a new preregistered
experiment version
In other words:

→ *FAILED*,

rather than:

→ Silently corrected EXP01.

After correction:

-v2

receives a new specification and undergoes a new cycle:

→ *FREEZE* → *RUN* → *AUDIT*.

8.19. Distinguishing Error Correction from Result Tuning

This is a subtle but important distinction.

An example of a genuine error:

a “+” sign was implemented in an equation instead of “-”.

This is an implementation defect.

A different situation:

the hypothesis was not supported at $\gamma = 1.5$, so let us try $\gamma = 0.8$.

This constitutes a new experimental regime.

It may be scientifically interesting, but it must not be presented as the original preregistered test.

The correct sequence is:

$\gamma = 1.5 \rightarrow$ null result,

followed by:

→ *EXP02*: $\gamma = 0.8$.

The research history then remains transparent.

8.20. A Negative Result Is Not an Error

HAAG explicitly establishes:

Negative and null results are valid RESULTS.
--

This is one of the framework's fundamental requirements.

If the hypothesis was formulated as:

H: $P_D \geq 1.20$,

and the experiment yielded:

$P_D = 1.01$,

this does not mean that the experiment “failed”.

If it was properly preregistered, executed, and audited, then:

$P_D = 1.01$

is a valid scientific result.

The appropriate conclusion is:

under the investigated conditions, the hypothesis was not supported according to the criterion established in advance.

8.21. A Negative Result Can Change the Theory

Moreover, a negative result can sometimes be more theoretically interesting than the expected positive one.

For example, a model may show:

$$\approx 0, MS > 0, M_S > 0,$$

but

$$\approx 0.$$

This means that the system retains a structural trace of an earlier influence, but that trace does not necessarily cause a noticeable change in the subsequent response.

This gives rise to a new distinction:

Structural Memory $\neq$ Functional Memory.

Thus, preregistration does more than prevent numerical result tuning. It makes it possible to recognize a result not anticipated by the initial theory and to revise the theory in response to the actual evidence.

8.22. AUDIT After Execution

After RUN, a result must not automatically enter the article.

The following is required:

AUDIT

For example:

AUDIT:

EXPERIMENT_ID:

EXP01

VERIFY:

model version

parameter values

initial conditions

input specification

controls

metric implementation

decision thresholds

relaxation conditions

random seed

numerical reproducibility

Three fundamentally different statuses are then possible.

VERIFIED RESULT

VERIFIED

The experiment was performed in accordance with preregistration, and the result may be used in the article.

VALID NULL/NEGATIVE RESULT

VERIFIED NULL

The experiment is valid, but the hypothesis was not supported.

This is also a valid RESULT.

INVALID EXPERIMENT

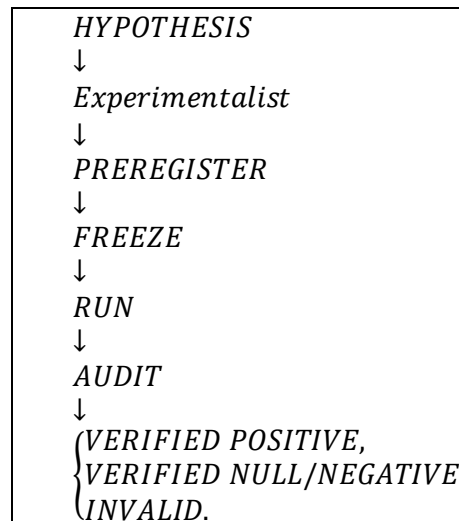
INVALID

A technical or methodological problem was identified that prevents interpretation of the result in relation to the hypothesis.

These three statuses must not be conflated.

8.23. Complete Experiment Life Cycle

Consequently, a numerical experiment in HAAG has a complete life cycle:



Only the first two outcomes constitute scientific results of the experiment.

The third requires correcting the design or implementation and creating a new experiment version.

8.24. Relationship Between PREREGISTER and HUMAN

The HUMAN and PREREGISTER primitives are closely related.

Before RUN, Experimentalist may determine many technical parameters.

However, if a proposal arises after RUN to:

change the primary metric;

change a threshold;

change the model;

remove a control;

change the interpretation of a negative result;

switch to a different parameter regime,

Orchestrator must assess whether the change is scientifically consequential.

If so:

RUN → RESULT → HUMAN → NEW PREREGISTRATION.

This does not prohibit adaptive research.

On the contrary, HAAG allows research to adapt, but requires an explicit distinction between what was determined before the result and what was decided after the result.

8.25. Two Modes: Confirmatory and Exploratory

This leads naturally to another useful distinction.

An experiment may be:

CONFIRMATORY

or

EXPLORATORY.

In confirmatory mode, strict preregistration applies:

MODE:

CONFIRMATORY

PREREGISTER

FREEZE

RUN

AUDIT

In exploratory mode, researchers may investigate parameters, identify regimes, and develop new metrics:

MODE:

EXPLORATORY

However, the findings of such exploration must not retrospectively be presented as preregistered confirmation.

If exploratory research reveals an interesting effect:

$\rightarrow H_{\text{new}},$

the next step should be:

$\rightarrow \text{PREREGISTER} \rightarrow \text{CONFIRMATORY RUN}.$

This combines freedom of scientific exploration with discipline in verification.

8.26. Role of TRACE in Preregistration

The entire history must remain visible:

TRACE:

EXP01:

preregistered

EXP01:

run completed

EXP01:

null result

HUMAN:

approved exploration of lower gamma

EXP02:

exploratory parameter scan

EXP03:

new hypothesis formulated

EXP03:

confirmatory preregistration

EXP03:

run

EXP03:

audit passed

A reader of the repository, or the researcher, can then reconstruct how the result actually emerged.

This is far more valuable than a repository containing only the final “successful” experiment.

8.27. Preregistration as a Mechanism of HAAG's Scientific Memory

In a broader sense, PREREGISTER, FREEZE, RUN, AUDIT, and TRACE together preserve a record of when a research decision was made.

This makes it possible to distinguish:

Prediction

as distinct from

Post Hoc Explanation.

Both can be scientifically useful, but they represent different types of knowledge.

For this reason, HAAG does not prohibit revising a theory after an experiment. It merely requires that the fact that the revision arose after observing the result not be concealed.

8.28. Shared Logic of HUMAN and PREREGISTRATION

Chapters 7 and 8 implement two complementary control mechanisms.

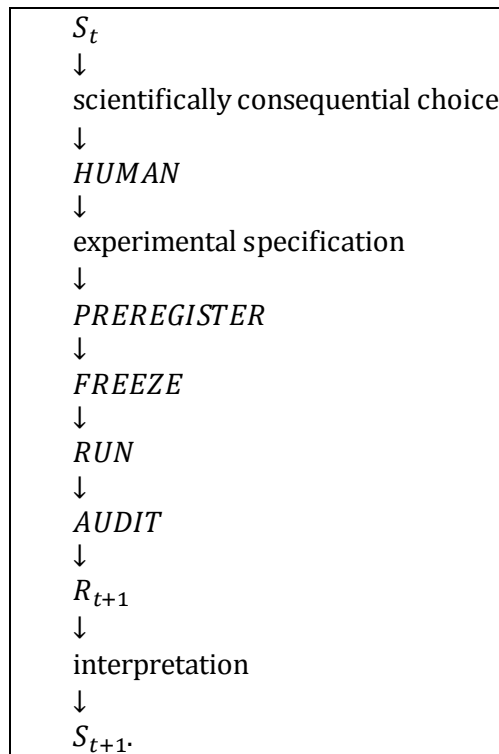
HUMAN controls scientifically consequential decisions:

meaningful choice → *HUMAN*.

PREREGISTER controls the timing of experimental decisions:

experimental choice → freeze before observing the result.

Together, they form a cycle:



If the result reveals a new fundamental alternative, the cycle returns to HUMAN.

Thus, HAAG does not seek to eliminate subjective scientific judgment or the adaptive character of research. It makes them explicit, traceable, and distinct from results obtained before those decisions.

This preserves the researcher's creative role while preventing the agent system from silently reshaping hypotheses, metrics, or experiments to fit an already known outcome.

9. RUN and Numerical Audit

9.1. From an Experiment Specification to an Actual Result

Once an experiment has been described using PREREGISTER and its specification frozen using FREEZE, HAAG moves from research design to actual execution.

For this purpose, the primitive

RUN

In its simplest form:

RUN:

OBJECT:

Thus, the execution object may be:

simulation — numerical simulation;

calculation — calculation;

analytical derivation — analytical derivation;

experiment — a computational or other formalized experiment.

The fundamental point is the distinction:

The essential distinction is:

Description of What Is to Be Performed $\neq$ Result of Actual Execution.

An LLM may explain convincingly what the outcome of a calculation should be, but such an answer is not yet an executed calculation.

HAAG therefore requires a clear distinction between reasoning about a result and a result actually obtained using the appropriate tool.

9.2. RUN Must Execute the Frozen Specification

RUN must not independently reinterpret the experiment.

If the following has been frozen:

$$\mathcal{E}_{\text{frozen}} = (M, I, P, F, C, \mu, \theta, S),$$

the actual run must execute precisely

$$\mathcal{E}_{\text{frozen}}).$$

The following situation is not permitted:

$$\mathcal{E}_{\text{frozen}} \rightarrow \mathcal{E}' \rightarrow \text{RUN}(\mathcal{E}'),$$

if the transition to $\mathcal{E}'$ has not been recorded anywhere.

Otherwise, the result no longer corresponds to the preregistered experiment.

9.3. Types of RUN Objects

Simulation

RUN:

OBJECT:

simulation

This may include:

integration of a system of differential equations;

simulation of network dynamics;

agent-based modeling;

a Monte Carlo experiment;

a stochastic process;

the temporal evolution of a complex system.

The result comprises not only a final number, but also trajectories, states, metrics, and diagnostic information.

Calculation

RUN:

OBJECT:

calculation

For example:

$$\Delta W, \quad P_D, \quad Q_C.$$

Even a simple calculation must be reproducible if it underpins a quantitative claim in the article.

Analytical derivation

RUN:

OBJECT:

analytical derivation

For example, from the equation

$$\beta x^2$$

one can analytically derive stationary states or a solution under specified initial conditions.

In this case, RUN denotes a formal derivation with its steps recorded.

Here, the audit checks the correctness of the mathematical transitions rather than numerical integration.

Experiment

RUN:

OBJECT:

experiment

This broader case may include:

processing real-world data;

a series of model runs;

comparison of control scenarios;

hypothesis testing;

application of an algorithm to a text corpus.

Thus, RUN in HAAG should not be understood exclusively as “run a program.” It is the operator for actually obtaining a research result.

9.4. What RUN Must Return

Standard result structure:

RETURN:

RAW_RESULT

METRICS

DIAGNOSTICS

WARNINGS

These four components must not be conflated.

9.5. RAW_RESULT — Primary Result

RAW_RESULT is the result prior to its substantive interpretation.

For example:

RAW_RESULT:

time_series

node_states

edge_weights

semantic_coordinates

final_state

For a numerical model, these may be:

$A(t), S(t), W(t). A(t), \sqcap S(t), \sqcap W(t).$

For text processing:

extracted entities;

relationships;

reconstructed semantic strings;

temporal slices.

For an analytical derivation:

a sequence of mathematical transformations;

the resulting expression;

the conditions under which it is valid.

RAW_RESULT should preferably be retained, as it allows metrics to be recomputed without repeating the entire experiment.

9.6. METRICS — Computed Metrics

The second component,

METRICS

contains quantities defined during preregistration.

For example:

METRICS:

R_F2 = ...

D3_F2 = ...

P_D = ...

P_L = ...

Delta_S = ...

Delta_W = ...

An important rule:

The metric must be calculated according to the definition fixed before RUN.

If, for example,

$$\frac{D_3(F_2|F_1)}{D_3(F_2|\emptyset)},$$

then, after obtaining the result, one must not silently replace D_3 with another characteristic that produces a more pronounced effect.

9.7. DIAGNOSTICS — Execution Diagnostics

Metrics alone are insufficient.

We must know whether the experiment operated as intended.

RUN therefore returns:

DIAGNOSTICS

For example:

DIAGNOSTICS:

relaxation_reached = TRUE

numerical_convergence = TRUE

max_residual = ...

integration_steps = ...

control_valid = TRUE

Diagnostics distinguish:

Negative Scientific Result

as distinct from

Technically Invalid Experiment.

These are different situations.

9.8. WARNINGS — Warnings

RUN must also return:

WARNINGS

A warning does not necessarily invalidate an experiment.

For example:

WARNINGS:

Metric P_L may be insensitive because
the graph diameter is small.

or:

WARNINGS:

The result is sensitive to integration tolerance.

or:

WARNINGS:

Control C3 changes both topology
and shortest-path distances.

Such warnings must be passed to AUDIT rather than concealed because the main result appears convincing.

9.9. Why a Separate AUDIT Is Needed After RUN

Even if RUN completes without a technical error, that does not mean every resulting number can be included in the article.

A separate stage is therefore introduced:

AUDIT

The logic is as follows:

$RUN \rightarrow RAW\ RESULT \rightarrow AUDIT \rightarrow VERIFIED\ RESULT.$

RUN answers the question:

What was obtained?

AUDIT answers a different question:

Can we trust this result specifically as the outcome of the declared experiment?

9.10. Recomputing Principal Results

The first audit requirement is:

Recompute all principal quantitative results from the frozen specification.

That is:

AUDIT:

1. Recompute all headline numerical results
from the frozen specification.

Headline numerical results are the numbers on which the article's conclusions directly depend.

For example:

$$\Delta W = 0.14, QC = 0.03. Q_C = 0.03.$$

If such a number appears in the Abstract, Results, Conclusions, a table, or a figure, it must be reproduced during the audit.

9.11. Checking Units, Indexing, and Normalization

The second requirement:

2. Check units, indexing, normalization, thresholds and initial conditions.

In practice, many numerical errors arise not from an incorrect equation but from small inconsistencies.

For example:

$v_0, \dots, v_{23}$

in the program implementation may correspond to

$v_1, \dots, v_{24}$

in the article.

Then an “influence on v_9 ” may in fact be applied to a different node.

The following are also checked:

units;

scales;

normalization;

thresholds;

indices;

initial values;

the ordering of vector components;

consistency of matrices with the model description.

9.12. Checking the Model Version

The third requirement:

3. Verify that reported numbers are produced by the current model version.

Suppose there were:

$M_1, M_2, M_3. M_1 \quad M_2 \quad M_3.$

For M_2 , the following value was obtained:

R=1.37.R=1.37.

Later, the model was changed to M_3 , but the old number accidentally remained in the text.
This is unacceptable.

There must be an explicit link:

$$\boxed{R_i \rightarrow EXP_j \rightarrow M_k.}$$

In other words, every result must have traceable provenance.

9.13. Provisional Numbers and Tool-Computed Numbers

The fourth requirement is especially important when working with LLMs:

4. Distinguish provisional hand/prose values
from tool-computed values.

During discussion, an LLM may write:

“We expect approximately $P_D \approx 1.2$.”

Alternatively, the researcher may make a manual estimate:

$$\Delta W \approx 0.15.$$

Such values can be useful when designing the study, but they have the status:

PROVISIONAL

They differ fundamentally from a number actually computed using the appropriate tool under the frozen specification.

Therefore:

A Number in LLM-Generated Text $\neq$ A Numerically Verified Result.

9.14. Numerical Sensitivity Testing

The fifth requirement:

5. Test numerical sensitivity when appropriate.

Not every experiment needs an extensive sensitivity analysis.

However, if a result may depend on:

the integration step;

tolerance;

seed;

sample size;

initial conditions;

discretization;

threshold;

this must be tested.

For example, if

$P_D = 1.21$

for one integration step, but

$P_D = 1.08$

for a slightly smaller step, the result cannot unreservedly be called robust.

Sensitivity analysis must not become a search for parameters that produce the desired effect.

It asks a different question:

Does the conclusion hold under reasonable numerical variations?

9.15. Recording Implementation Ambiguities

The sixth requirement:

6. Record implementation ambiguities.

For example, the mathematical description states:

“The weight of edges near the transition node equals 0.5.”

However, it does not specify whether this means:

all edges incident to the node;

only inter-string edges;

an additional coefficient;

replacement of the baseline weight.

In this case, Numerical Auditor must not independently select the most convenient interpretation.

It must record:

AMBIGUOUS_SPECIFICATION

and describe the ambiguity.

If it affects the scientific meaning, HUMAN intervention and a new specification are required.

9.16. The Auditor Must Not Help the Hypothesis

The seventh principle:

7. Never silently choose an interpretation
that favors the hypothesis.

For example, if two implementations are admissible:

$$\rightarrow P_D = 1.03, I_B \rightarrow P_D = 1.24, I_B \rightarrow P_D = 1.24,$$

one must not automatically choose I_B merely because it supports the hypothesis.

The appropriate response is:

STATUS:

AMBIGUOUS_SPECIFICATION

or, if both alternatives are justified, to report both results.

This is a central principle of Numerical Auditor:

The audit verifies the result rather than defending the hypothesis.

9.17. Numerical Audit Statuses

After verification, a result receives one of four statuses.

STATUS:

VERIFIED_NUMERICALLY |

PROVISIONAL |

FAILED_REPRODUCTION |

AMBIGUOUS_SPECIFICATION

Let us consider them individually.

9.18. VERIFIED_NUMERICALLY

VERIFIED_NUMERICALLY

means that:

the specification is unambiguous;

the experiment conforms to the frozen version;

the result has been reproduced;

the metrics have been calculated correctly;

the initial conditions have been checked;

no significant contradictions have been found.

Only this status permits a number to be used as a definitive quantitative claim.

9.19. PROVISIONAL

PROVISIONAL

means that a number exists but has not yet undergone a complete audit.

For example:

a manual estimate;

a preliminary run;

a number generated during dialogue;

an exploratory calculation;

a result from an older model version that has not yet been checked.

Such a number may be used to plan further work, but must not be presented as a final article result.

9.20. FAILED_REPRODUCTION

FAILED_REPRODUCTION

means that the reported result could not be reproduced.

For example, the draft states:

PD=1.31,P_D=1.31,

but a repeat run under the frozen specification yields:

PD=1.04.P_D=1.04.

In this case, the old number must not be “averaged,” the best run selected, or the value retained in the article.

The reason for the discrepancy must be established.

9.21. AMBIGUOUS_SPECIFICATION

AMBIGUOUS_SPECIFICATION

means that the specification does not permit unambiguous reproduction of the experiment.

This is not the same as FAILED_REPRODUCTION.

With FAILED_REPRODUCTION, we know what should have been executed but cannot obtain the reported result.

With AMBIGUOUS_SPECIFICATION, we cannot determine unambiguously what should have been executed.

9.22. The Fundamental Rule for Numerical Claims

Having introduced these statuses, HAAG establishes a strict rule:

Only VERIFIED_NUMERICALLY values may appear as final quantitative claims.

That is, only values with the status

VERIFIED_NUMERICALLY

may appear without qualification in:

the Abstract;

Results;

Conclusions;

tables;

figures;

figure captions;

principal quantitative claims.

For PROVISIONAL values, the status must be clearly visible in working materials.

10. Failure as a Legitimate Research State

10.1. Why Failed Experiments Must Not Be Deleted

A common temptation when preparing a paper is to show the reader only the successful trajectory:

$$\rightarrow M \rightarrow EXP \rightarrow RESULT.$$

Actual research is almost never so straightforward.

A typical trajectory looks more like this:

$$\rightarrow M_1 \rightarrow EXP_1 \rightarrow FAIL \rightarrow M_2 \rightarrow EXP_2 \rightarrow NULL \rightarrow H' \rightarrow \dots$$

HAAG treats this history not as noise but as part of the scientific process.

Therefore:

$$\boxed{FAILURE \in Research\ State.}$$

An unsuccessful branch is not erased.

10.2. FAILURE Does Not Always Mean the Same Thing

Failure can describe fundamentally different situations.

For example:

the relaxation criterion was not reached;

instability makes the planned test invalid;

a metric conflates several effects;

real-world data do not permit causal interpretation;

a hypothesis produced a null result.

These cases must not all be labeled simply as “failure.”

10.3. Failure to Reach Relaxation

Suppose preregistration requires:

$$\max_i a_i < 10^{-3}$$

and

$$\|\dot{A}\|_2 < 10^{-4}.$$

If these conditions are not satisfied by $T_{\max}$, the experiment cannot claim that the system has returned to a relaxed state.

The following is then recorded:

FAILURE_TYPE:

RELAXATION_NOT_REACHED

This may mean that:

the time horizon was insufficient;

the model exhibits slow dynamics;

self-sustained activity exists;

the assumption of relaxation is itself incorrect.

Thus, even an apparently technical failure may contain scientific information.

10.4. Model Instability

Another case:

FAILURE_TYPE:

MODEL_INSTABILITY

If an experiment is designed to compare residual structures after activity has decayed, but the activity itself does not decay, the planned test may lose its meaning.

This does not automatically mean that the model is “bad.”

A different regime of its behavior may have been discovered.

However, the particular experiment,

EXP_i,

does not test what it was designed to test.

10.5. Confounded Metric

Suppose a metric is intended to characterize propagation along a semantic string, but also depends strongly on changes in geodesic distance after rewiring.

Then:

FAILURE_TYPE:

CONFOUNDED_METRIC

The problem is not necessarily in the data, but in the fact that:

desired effect, topological effect).

In this case, the metric does not allow the observed change to be attributed unambiguously to the intended mechanism.

10.6. Real-World Data Do Not Support a Causal Interpretation

This is particularly important for real information flows.

For example, temporal slices may demonstrate:

$$\rightarrow \mathcal{G}_S(t_1), D_2 \rightarrow \mathcal{G}_S(t_2), D_3 \rightarrow \mathcal{G}_S(t_3).$$

Differences may be observed between

$$\mathcal{G}_S(t_1), \mathcal{G}_S(t_2), \mathcal{G}_S(t_3).$$

But this does not automatically establish:

$$\mathcal{G}_S(t_1) \rightarrow \mathcal{G}_S(t_2) \rightarrow \mathcal{G}_S(t_3)$$

in the causal sense.

The following may therefore be recorded:

FAILURE_TYPE:

CAUSAL_INTERPRETATION_NOT_SUPPORTED

This does not mean that the real-data case is useless.

Its role may be changed from:

CAUSAL_VALIDATION

to:

OPERATIONALIZATION

That is, the data demonstrate that the method can be applied in practice, but do not prove the causal hypothesis.

10.7. Null Result for a Hypothesis

Another important case:

FAILURE_TYPE:

NULL_HYPOTHESIS_RESULT

For example, the expected result was:

$$P_D \geq 1.20,$$

but the obtained result was:

$$\approx 1.$$

If the experiment was correctly performed, this is not a failure in the sense of poor research execution.

It is:

valid null result.

HAAG therefore distinguishes:

FAILURE_experiment

and

NULL_hypothesis.

The former may render a result unusable.

The latter is itself a result.

10.8. Structure of a Failure Record

Every significant failure should have a structured record:

FAILURE_TYPE

CAUSE

WHAT_WAS_LEARNED

WHICH_CLAIMS_ARE_INVALIDATED

WHICH_NEXT_TRANSITIONS_ARE_ALLOWED

This transforms “something did not work” into a formalized research state.

10.9. FAILURE_TYPE

For example:

FAILURE_TYPE:

CONFOUNDED_CONTROL

or:

FAILURE_TYPE:

FAILED_REPRODUCTION

or:

FAILURE_TYPE:

NULL_RESULT

The type of failure should be specific enough to determine subsequent actions.

10.10. CAUSE

The field

CAUSE

explains the established cause.

For example:

CAUSE:

Shuffled topology changed both
sequential organization and
shortest-path structure.

If the cause is not yet known, this must also be recorded:

CAUSE:

UNKNOWN

An explanation must not be invented merely to make the record appear complete.

10.11. WHAT_WAS_LEARNED

This is one of the most important fields.

For example:

WHAT_WAS_LEARNED:

The current shuffled-topology control
cannot isolate semantic sequentiality
from generic topological accessibility.

Thus:

→ *KNOWLEDGE*.

An unsuccessful control can reveal how the next control should not be designed.

10.12. WHICH_CLAIMS_ARE_INVALIDATED

Failure must automatically affect the Claims Ledger.

For example:

WHICH_CLAIMS_ARE_INVALIDATED:

C17

C19

or:

C17:

cannot claim topology-independent effect

C19:

cannot claim causal validation

In other words, failure must not remain confined to the experiment log while an outdated, overly strong claim continues to appear in the article.

10.13. WHICH_NEXT_TRANSITIONS_ARE_ALLOWED

After a failure, the system should identify permissible next steps.

For example:

WHICH_NEXT_TRANSITIONS_ARE_ALLOWED:

REDESIGN_CONTROL

REFORMULATE_HYPOTHESIS

HUMAN

PREREGISTER_NEW_EXPERIMENT

But not:

TUNE_CURRENT_EXPERIMENT_UNTIL_SUCCESS

Thus, failure does not terminate the investigation; it constrains the set of valid subsequent transitions.

10.14. A Failed Control and a New Experiment

The original HAAG rule is:

A failed control may justify redesigning the next experiment, but the new design must undergo preregistration, and the unsuccessful branch must be preserved.

Formally:

$$\rightarrow FAILURE \rightarrow LESSON \rightarrow DESIGN_2 \rightarrow PREREGISTER(EXP_2).$$

The following is not permitted:

$$\rightarrow FAILURE \rightarrow \text{secret modification} \rightarrow EXP_1.$$

This is the fundamental distinction between iterative research and experiment tuning.

10.15. FAILURE as a Mechanism of Research Evolution

In HAAG, a failure may change:

H, M, Q, E, C, B.

For example:

$$\rightarrow EXP_1 \rightarrow NULL$$

may generate a new question:

Why Does Structural Memory Not Transition into Functional Memory?

Then:

$$\rightarrow H_2 \rightarrow M_2.$$

Thus, failure may be not the end of a branch but the source of the next scientific problem.

11. Claims Ledger and Claim Provenance

11.1. Why a Claims Ledger Is Necessary

A substantial research article gradually accumulates dozens of claims of different kinds.

Some are definitions:

A semantic string is an ordered sequence of semantic units.

Others are mathematical consequences:

The system has a stationary state under the following conditions.

Others are numerical results:

In EXP01, the value $P_D = \dots$ was obtained.

Others are interpretations:

This indicates structural memory.

Still others concern novelty:

The proposed model differs from known adaptive-network models in that ...

These claims require different forms of justification.

HAAG therefore introduces a

CLAIMS LEDGER

—a register of the publication's principal claims and their provenance.

11.2. Basic Record Structure

For each substantive claim, a record is created:

CLAIM_ID

CLAIM_TEXT

TYPE

SUPPORT

STATUS

ALLOWED_STRENGTH

For example:

CLAIM_ID:

C17

CLAIM_TEXT:

The model retains semantic-structural deformation after activation decays.

TYPE:

NUMERICAL_RESULT

SUPPORT:

EXP01

STATUS:

SUPPORTED

ALLOWED_STRENGTH:

limited

This separates the wording of a claim from the evidence that permits it to be made.

11.3. CLAIM_ID — Unique Identifier

Every principal claim is assigned an identifier:

C01

C02

C03

...

This makes it possible to establish explicit links:

$$\rightarrow E_j, C_i \rightarrow EXP_k, C_i \rightarrow EXP_k, C_i \rightarrow HUMAN_m, C_i \rightarrow HUMAN_m.$$

This substantially simplifies the audit of lengthy manuscripts.

11.4. CLAIM_TEXT — The Claim Itself

The field

CLAIM_TEXT

should state the claim as closely as possible to the wording that will appear in the manuscript.

For example:

CLAIM_TEXT:

Structural deformation persists after
the activation variable returns close
to its baseline.

This matters because support is assessed for a specific formulation, not for a general topic.

11.5. Claim Types

HAAG provides the following types:

TYPE:

DEFINITION

DERIVATION

NUMERICAL_RESULT

EMPIRICAL_OBSERVATION

INTERPRETATION

NOVELTY

LIMITATION

Each type has its own logic of support.

11.6. DEFINITION — Definition

For example:

TYPE:

DEFINITION

A semantic string shall mean an ordered sequence of semantic units ...

A definition does not require experimental confirmation in the same sense as a numerical result.

It must, however, be:

unambiguous;

internally consistent;

compatible with other definitions;

clearly identified as the authors' definition if introduced in the study.

11.7. DERIVATION — Derived Result

For example:

TYPE:

DERIVATION

The claim is supported by a mathematical derivation:

→ Equations → Claim.

Here SUPPORT must refer to a specific equation, theorem, or sequence of transformations.

11.8. NUMERICAL_RESULT — Numerical Result

TYPE:

NUMERICAL_RESULT

Support must take the form:

SUPPORT:

EXPERIMENT:

EXP01

AUDIT_STATUS:

VERIFIED_NUMERICALLY

If the status is merely:

PROVISIONAL

the claim cannot be used as a final quantitative result.

11.9. EMPIRICAL_OBSERVATION — Empirical Observation

For example:

A recurring semantic sequence was reconstructed in temporal slice D₂ ...

Here, support may include:

dataset;
corpus;
observation protocol;
source documents;
annotation procedure.

It is essential to distinguish clearly between:

We Observe X

and

X Caused Y.

The latter is a much stronger claim and requires a different evidentiary basis.

11.10. INTERPRETATION — Interpretation

An interpretation is not the result itself.

For example:

$$\Delta S > 0, \quad \Delta W > 0$$

—is a result.

By contrast, the sentence:

This can be interpreted as the persistence of semantic-structural memory.

—is already an:

TYPE:

INTERPRETATION

This distinction is highly useful.

It prevents an interpretation from silently becoming a “fact established by the experiment.”

11.11. NOVELTY — Novelty Claim

TYPE:

NOVELTY

requires particularly rigorous auditing.

For example, the statement:

Proposed for the first time ...

is substantially stronger than:

We propose ...

or:

Unlike the models considered, this study ...

For NOVELTY, support must come not from a single experiment but from a systematic review of the relevant literature.

For example:

SUPPORT:

literature search

Source Auditor

comparison matrix

HUMAN decision

11.12. LIMITATION — Statement of Limitations

Limitations must also form part of the Claims Ledger.

For example:

TYPE:

LIMITATION

CLAIM_TEXT:

The real-data case demonstrates

operationalization but does not

establish causal validation.

Such a statement protects the manuscript against an interpretation stronger than the data warrant.

11.13. SUPPORT — The Basis of a Claim

For every CLAIM, specify:

SUPPORT:

equation

experiment

source

dataset

human decision

A single claim may have several types of support.

For example:

SUPPORT:

equation:

Eq. 7

experiment:

EXP01

source:

REF12

human decision:

D17

This creates a provenance graph:

$$CLAIM \leftarrow \{Equation, Experiment, Source, Dataset, HumanDecision\}.$$

11.14. Claim Statuses

Each claim is assigned a status:

STATUS:

SUPPORTED

PARTIALLY_SUPPORTED

UNSUPPORTED

CONTRADICTED

UNVERIFIABLE

PROVISIONAL

SUPPORTED

Sufficient verified support is available.

PARTIALLY_SUPPORTED

Only part of the claim, or a restricted formulation, is supported.

For example:

The effect exists in all networks.

may be too strong.

However:

The effect was observed in the configuration investigated.

may be supported.

UNSUPPORTED

There is insufficient evidence.

This does not necessarily mean that the claim is false.

It means only:

The available work does not support it.

CONTRADICTED

The obtained results contradict the claim.

For example:

H: $P_D \geq 1.20$,

but the verified result is:

$$\approx 1.$$

UNVERIFIABLE

The claim cannot be verified with the available means.

For example, a strong causal claim based solely on observational data.

PROVISIONAL

The claim has preliminary support but requires further auditing.

This is a natural status for a working research draft.

11.15. ALLOWED_STRENGTH — How Strongly May a Claim Be Worded?

HAAG introduces another important parameter:

ALLOWED_STRENGTH:

descriptive

limited

general

This is necessary because the same evidence may support claims of differing strength to different degrees.

11.16. Descriptive — Descriptive Level

ALLOWED_STRENGTH:

descriptive

For example:

Three types of semantic sequences were observed in the corpus investigated.

This describes an observation without broad generalization.

11.17. Limited — Restricted Claim

ALLOWED_STRENGTH:

limited

For example:

In the model configuration considered, the preceding influence caused persistent structural deformation.

The scope is explicitly specified here:

In the configuration under consideration.

11.18. General — General Claim

ALLOWED_STRENGTH:

general

For example:

A preceding informational influence creates structural memory in a semantic network.

This is a broad generalization.

It requires substantially stronger support:

theoretical justification;

multiple regimes;

robustness;

external validation;

possibly empirical data.

Therefore, the transition

→ limited → general

is not stylistic.

It increases the strength of the scientific claim.

11.19. Fundamental Rule of the Claims Ledger

The central rule of this chapter is:

A stronger sentence may not be written than the strongest verified support permits.

In plain language:

No claim in the manuscript may be stated more strongly than its strongest verified evidentiary support permits.

For example, if an experiment shows that:

in a particular configuration, a structural change persists after relaxation,

one cannot automatically write:

informational influences always produce long-term memory in society.

The gap between these sentences represents an enormous increase in generality.

11.20. CLAIMS_AUDIT Before Final Writing

Before the final WRITE stage, perform:

CLAIMS_AUDIT

For every major claim, verify:

CLAIMS_AUDIT:

FOR each major claim:

 identify CLAIM_ID

 verify CLAIM_TEXT

verify TYPE
verify SUPPORT
verify STATUS
verify ALLOWED_STRENGTH
check manuscript wording
check citation
check numerical status

If the wording exceeds the strength of the support:

REWRITE

If no support exists:

REJECT CLAIM

or:

MARK:

UNSUPPORTED

11.21. Claims Ledger as a Knowledge-Provenance Graph

In its extended form, the Claims Ledger can be viewed as a directed graph:

$G_C = (V_C, E_C)$,

where the vertices represent:

hypotheses;

equations;

experiments;

results;

sources;

datasets;

HUMAN decisions;

manuscript claims.

For example:

$\rightarrow EXP01 \rightarrow R_{01} \rightarrow C_{17} \rightarrow P_{Results}.$

For another claim:

$\rightarrow C_{04} \rightarrow P_{Introduction}.$

For a methodological limitation:

$\rightarrow C_{31} \rightarrow P_{Discussion}.$

Thus, every substantive sentence in the article has traceable provenance.

11.22. Relationship Between RUN, FAILURE, and the CLAIMS LEDGER

Chapters 9–11 form a unified mechanism.

First:

$\rightarrow FREEZE \rightarrow RUN.$

Then:

$\rightarrow AUDIT.$

If the result is valid:

$\rightarrow VERIFIED_NUMERICALLY \rightarrow CLAIM.$

If the result is null:

$\rightarrow VERIFIED_NULL \rightarrow CLAIM$

—but with an appropriately limited interpretation.

If the experiment is invalid:

$\rightarrow FAILURE \rightarrow TRACE.$

The failure then determines:

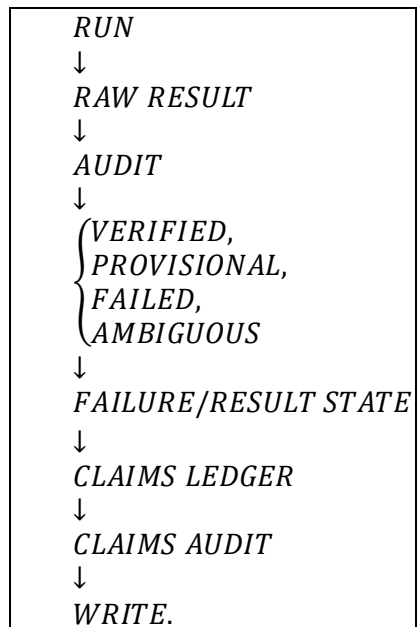
Which CLAIMS Are No Longer Permitted

and

Which Transitions Are Allowed Next.

11.23. General Logic of the Three Chapters

Thus, Chapters 9, 10, and 11 establish a form of scientific-integrity feedback loop for research results in HAAG:



This loop implements a simple but fundamental approach:

First obtain a result. Then verify it. Next determine exactly what it permits one to claim.

Only then formulate the sentences of the scientific publication.

This sequence prevents HAAG from becoming a system that first generates persuasive prose and then selects results and references to support it.

In HAAG, the direction must be the reverse:

Evidence → Verified Result → Allowed Claim → Scientific Text.

If the support weakens at any stage, the strength of the corresponding claim in the manuscript must weaken accordingly.

12. Mandatory IMRaD Logic

12.1. IMRaD in HAAG Is More Than an Article Format

HAAG uses the IMRaD structure—Introduction, Methods, Results, and Discussion—not merely as an editorial template for arranging text. It defines the logic by which a research problem is transformed into a scientifically justified conclusion.

In its most general form, this logic is expressed as the following sequence:

Gap → Objective → Method → Result → Interpretation → Conclusion

or, in words:

Gap → Objective → Method → Result → Interpretation → Conclusion.

Each successive element must follow logically from the preceding one.

If the Introduction identifies one scientific problem, the Methods investigate another, and the Conclusions advance a third, much broader claim, the mere presence of Introduction, Methods, Results, and Discussion headings does not make the work IMRaD-compliant in the HAAG sense.

HAAG therefore checks not only whether the sections are present, but also the logical transitions between them.

12.2. INTRODUCTION — Formulating the Scientific Problem

The basic structure of the Introduction is:

INTRODUCTION:

context

state of knowledge

explicit gap

objective

research question/hypothesis

contribution boundaries

The purpose of the Introduction is not simply to explain why a topic is interesting.

It must guide the reader from the subject domain to the specific problem addressed by the study:

context → established findings → unresolved problem → study objective.

12.3. Context — Research Context

First, the domain within which the problem arises must be identified.

The context answers the question:

What system, process, or class of phenomena are we studying, and why is it of scientific interest?

For example, it is insufficient to write:

Information influences play an important role in society.

The scope must gradually be narrowed to the subject of the study: information influences, semantic structure, network representation, state dynamics, and so forth.

The context establishes the domain within which the future article may legitimately make claims.

12.4. State of Knowledge — What Is Already Known

The next component,

state of knowledge

must concisely describe the existing state of research.

It is necessary to establish:

which classes of models already exist;
which mechanisms have already been investigated;
what results have been obtained;
which approaches are closest to the proposed one;
which of their capabilities and limitations are relevant to this study.
This component must rely on SOURCE and CITE, rather than the LLM's general impression of the literature.

12.5. Explicit Gap — An Explicitly Formulated Research Gap

Particular importance attaches to the explicit gap
HAAG does not permit the research gap to remain implicit.
It must be clear:

what existing approaches do not explain, model, test, or provide.

For example, a weak formulation is:

Existing methods have certain shortcomings.

A substantially more informative formulation is:

Existing models describe either information propagation or the evolution of network structure, but do not describe, within a unified mechanism, changes in semantic state together with structural reconfiguration induced by preceding information influences.

However, this statement may be used only after the literature has been checked.

Therefore:

$\leftarrow$ *SOURCE + Source Auditor.*

12.6. Objective — Research Objective

The objective must directly address the identified gap:

$\rightarrow$ *Objective.*

If the gap concerns the absence of a mechanism for modeling a particular process, the objective may be to develop such a mechanism.

If the gap is the absence of a means of testing a certain property, the objective must concern its investigation or assessment.

HAAG must ask a simple question:

If the objective is achieved, will the stated gap be eliminated or at least reduced?

If not, the Gap $\rightarrow$ Objective relationship is broken.

12.7. Research Question / Hypothesis

The objective is then specified in the form of research questions:

$Q_1, Q_2, \dots, Q_n$

or hypotheses:

$H_1, H_2, \dots, H_m$.

For example:

Can a preceding information influence change the structure of a system in such a way that its response to a subsequent influence becomes different?

A testable hypothesis can then be formulated.

It is important that the hypothesis be connected to the method to be used:

$$\rightarrow M_j \rightarrow EXP_k.$$

A hypothesis for which the study provides no testing procedure must not create the impression of an experimentally verified claim.

12.8. Contribution Boundaries — Limits of the Contribution

HAAG requires the Introduction not only to state what the study does, but also to delineate what it does not claim to do.

For example:

The proposed model investigates the mechanism of semantic–structural adaptation but does not purport to provide a complete psychological model of individual information perception.

Or:

The real-world information case is used to operationalize the proposed approach, not as proof of a causal influence.

This field is particularly important for the subsequent CLAIMS_AUDIT.

12.9. METHODS — How the Result Was Obtained

Structure of Methods:

METHODS:

definitions and assumptions

data/materials

mathematical model

algorithms/procedures

preregistered experiments

metrics and thresholds

reproducibility conditions

Methods must make it possible to understand:

How was the result obtained from the research question?

For a theoretical article, Methods need not imply experiments with physical or empirical objects. It may include definitions, assumptions, mathematical constructs, algorithms, and numerical experiments.

12.10. Definitions and Assumptions

The objects of a mathematical model must be defined before the model is introduced.

For example:

$$G=(V,E,W), G=(V,E,W),$$

where V is the set of nodes, E the set of edges, and W their weights.

States, parameters, external influences, and assumptions must likewise be defined.

Assumptions must be stated explicitly.

An assumption must not be disguised as an established fact.

12.11. Data / Materials

If data are used, the following must be described:

their provenance;

the time period;

selection criteria;

the unit of analysis;

preprocessing;

limitations;

the procedure for reproducing the sample.

For real-world data, this component is connected to the rules of Chapter 13.

12.12. Mathematical Model

A mathematical model must contain not only equations but also the meaning of their components.

For example:

$$\gamma_i a_i + (1 - a_i) \sum_j w_{ij} K_{ij} a_j + f_i(t).$$

It is necessary to explain:

what a_i represents;

what w_{ij} represents;

the role of K_{ij} ;

what $f_i(t)$ specifies;

which assumptions underlie the equation.

An equation without an interpretation does not yet constitute a scientific model.

12.13. Algorithms and Procedures

If the model is implemented procedurally, the algorithm must also be described.

For example:

→ Initialization → Influence → Evolution → Relaxation → Measurement.

The description must be sufficient for another researcher to understand and, where possible, reproduce the procedure.

12.14. Preregistered Experiments

For the principal confirmatory experiments, Methods must reflect the frozen specification:

→ FREEZE.

This includes:

the hypothesis;

the model;

parameters;

initial conditions;

the influence;

controls;

metrics;

thresholds;

termination conditions.

This connects Chapters 8 and 12.

12.15. Metrics and Thresholds

Metrics must be defined mathematically before they are used in Results.

For example:

$$\frac{D(F_2|F_1)}{D(F_2|\emptyset)}.$$

Thresholds, too, must be specified before Results if they are used to classify outcomes.

Thus:

what we measure and how,

whereas

what was actually obtained.

12.16. Reproducibility Conditions

Methods must specify sufficient conditions for reproducibility:

$M, P, I, F, C, \mu, \xi, M, \quad P, \quad I, \quad F, \quad C, \quad \mu, \quad \xi$.

That is, the model version, parameters, initial conditions, influences, controls, metrics, and randomness parameters.

Detailed materials may be placed in a GitHub repository or reproducibility package, but the article must clearly identify what was actually performed.

12.17. RESULTS — Only What Was Actually Obtained

Basic specification:

RESULTS:

only actually obtained results

negative/null results included

diagnostics and failed controls

when scientifically informative

tables/figures

validation

This is one of HAAG's strictest principles:

Results $\neq$ Expected Results.

A number must not appear in Results merely because it is theoretically expected or fits the hypothesis well.

12.18. Only Actually Obtained Results

Results includes only findings that were actually obtained.

For quantitative results, the rule from Chapter 9 additionally applies:

STATUS=VERIFIED_NUMERICALLY.STATUS=VERIFIED_NUMERICALLY.

Therefore:

expected value $\neq$ result; manual estimate $\neq$ verified result.

12.19. Negative and Null Results

If a preregistered hypothesis is not supported, that is not a reason to remove the result.

For example:

$H1: PD \geq 1.20$,

but

$PD = 1.03$. $P_D = 1.03$.

This is precisely what must be reported in Results.

A negative result must not be relegated solely to Limitations, leaving the main text with the misleading impression that every experiment supported its hypothesis.

12.20. Diagnostics and Failed Controls

If a failed control helps clarify the limits of a result, it also has scientific value.

For example:

The shuffled-topology control proved unsuitable for isolating the effect of semantic sequentiality because it simultaneously changed the network's geodesic structure.

This may be an important methodological finding.

12.21. Tables and Figures

Tables and figures must present already verified results rather than introduce new claims.

The proper logic is:

→ *FIGURE*.

Not the reverse:

→ search for an interpretation.

If a figure is conceptual, its role and caption must make that clear.

12.22. Validation

Results must also show how the reliability of the findings was assessed:

repeated runs;

control scenarios;

sensitivity analysis;

comparisons;

numerical audit;

external data, if their actual role is validation.

The strength of the validation must not be exaggerated.

12.23. DISCUSSION — What the Results Mean

Structure:

DISCUSSION:

interpretation

comparison with literature

alternative explanations

what the model does NOT show

limitations

conditions for future generalization

Discussion is where a result is transformed into scientific understanding.

However, it is also where it is particularly easy to move from a supported result to excessive generalization.

12.24. Interpretation

The first task is to explain:

What does the result mean in the context of the research question?

For example:

$$\Delta W > 0$$

is, in itself, a result.

The statement

this indicates the preservation of a structural trace of a preceding influence
is an interpretation.

These two levels must remain distinct.

12.25. Comparison with Literature

The result must be related to previous studies:

$$\leftrightarrow R_{\text{literature}}$$

The comparison must address more than the names of methods; it must examine:

assumptions;

mechanisms;

variables;

types of dynamics;

the scope of application;

the nature of the result.

This is where SOURCE once again becomes an active research operator.

12.26. Alternative Explanations

HAAG requires consideration of alternative explanations.

For example, the observed effect may be caused by:

semantic mechanism,

but also by:

topology, parameter regime, initial conditions.

Contrarian and Falsifier must help formulate such alternatives.

Discussion must not act solely as an advocate for the author's hypothesis.

12.27. What the Model Does NOT Show

This item is especially important.

It must explicitly identify which conclusions the model does not permit.

For example:

The model demonstrates the possibility of structural memory emerging, but does not prove that this mechanism is dominant in real social systems.

This does not weaken the study. Rather, it defines the bounds of its scientific validity.

12.28. Limitations

Limitations must follow from the actual research design.

For example:

simplified topology;

synthetic parameters;

a limited range of regimes;

lack of causal identification in the real-data case;

dependence on the semantic-network reconstruction procedure.

Limitations must not be merely a formal paragraph at the end of the article.

They determine the

ALLOWED_STRENGTHALLOWED_STRENGTH

of the principal claims.

12.29. Conditions for Future Generalization

Instead of the abstract statement:

Further research is needed in the future,

HAAG proposes identifying the conditions that would justify strengthening the claims.

For example:

To move from a model-specific finding to a more general conclusion, the effect must be tested across different topologies, parameter regimes, and independent real-world datasets.

That is:

required evidence]future workgeneral claim.

12.30. CONCLUSIONS — Only What Results Support

Fundamental rule:

CONCLUSIONS:

only conclusions supported by RESULTS

Conclusions is not the place to introduce a new result.

The correct sequence is:

$\rightarrow INTERPRETATION \rightarrow CONCLUSION.$

The following is not permitted:

$\rightarrow desired\ conclusion.$

If a claim has no support in Results and the Claims Ledger, it must not suddenly appear in Conclusions.

12.31. IMRaD as a System for Checking Article Integrity

Before completion, HAAG must traverse the sequence in both directions.

In the forward direction:

$\rightarrow Objective \rightarrow Method \rightarrow Result \rightarrow Interpretation \rightarrow Conclusion.$

And in reverse:

$\rightarrow Result \rightarrow Method \rightarrow Objective \rightarrow Gap.$

For every conclusion, it must be possible to answer:

Which result supports it?

For every result:

By which method was it obtained?

For every method:

Which research question does it test?

For every question:

Which gap does it address?

This is HAAG's mandatory IMRaD logic.

13. Operationalization Rule for Real-World Data

13.1. Why the Role of a Real-World Case Must Be Defined

Theoretical and model-based papers often include an example drawn from real-world data. A problem arises when the mere use of a real corpus is automatically presented as “model validation.”

HAAG prohibits this automatic strengthening of a conclusion.

Before using a real-world case, its role must be explicitly specified:

ROLE:

ILLUSTRATION |

OPERATIONALIZATION |

EXTERNAL_VALIDATION |

CAUSAL_TEST

These roles differ in their epistemic strength.

13.2. ILLUSTRATION

ROLE:

ILLUSTRATION

means that a real-world example is used to demonstrate an idea.

It answers the question:

How can the proposed concept be illustrated using comprehensible real-world material?

An illustration does not prove that the model is correct.

13.3. OPERATIONALIZATION

ROLE:

OPERATIONALIZATION

has a stronger meaning.

Here, real-world data demonstrate that abstract objects in the model can be mapped onto observable data.

For example:

documents $\rightarrow$ entities $\rightarrow$ semantic relations $\rightarrow G_S(t)$.

Thus, the theoretical concept

$G_S(t)$

acquires an operational procedure for its construction.

This is an important result, but:

OPERATIONALIZATION $\neq$ CAUSAL VALIDATION.

13.4. EXTERNAL_VALIDATION

ROLE:

EXTERNAL_VALIDATION

means that data independent of the principal modeling experiment are used to test specific predictions or properties of the model.

In this case, it is necessary to specify clearly:

what exactly the model predicts

and

which observation in real-world data could support or refute it.

The mere visual resemblance of a figure to a model trajectory does not constitute external validation.

13.5. CAUSAL_TEST

The strongest role is:

ROLE:

CAUSAL_TEST

presupposes that the research design genuinely permits a causal claim to be tested.

The following sequence alone is insufficient:

$X(t_1), X(t_2), X(t_3)$.

Temporal succession does not, by itself, establish:

$X(t_1) \Rightarrow X(t_2)$.

Therefore, CAUSAL_TEST should be used only when the research design provides grounds for causal interpretation.

13.6. Prohibition on Implicitly Upgrading the Role

The central rule is:

OPERATIONALIZATION $\nRightarrow$ CAUSAL VALIDATION.

If a real-world case was introduced as operationalization, it must not be quietly transformed into evidence for a causal hypothesis in the Discussion or Conclusions.

For example, it is permissible to state:

Real-world data demonstrate the feasibility of reconstructing temporal semantic structures using the proposed method.

Without additional grounds, however, this must not be replaced by:

Real-world data confirm that an earlier information influence caused the observed restructuring of the semantic structure.

The second statement is considerably stronger.

13.7. Independent Reconstruction of Temporal Slices

For temporal data, HAAG recommends, where scientifically appropriate: reconstructing each temporal slice independently.

For example:

$\rightarrow G_1, D_2 \rightarrow G_2, D_3 \rightarrow G_3. D_3 \rightarrow G_3.$

Rather than:

$\rightarrow$ expectation regarding $G_2 \rightarrow G_2.$

This approach can be regarded as a form of blind reconstruction.

Its advantage is that the structure of the preceding slice does not force the algorithm or the researcher to search for the expected continuation in the next slice.

13.8. Publication Date and Event Date

For information flows, at least two temporal coordinates must be distinguished:

publication date, $t_e = \text{event/activitydate}$. $t_e = \text{event/activity date}$.

A document published on May 15 may describe an event that occurred on May 10.

If these dates are conflated, the reconstructed dynamics may be shifted.

Temporal analysis must therefore explicitly indicate which date is used.

13.9. Preserving Source IDs

For each document or other real-world data object, its identifier should be preserved whenever possible:

$\leftrightarrow \text{source_id}_i$.

This makes it possible to trace an aggregated result back to the primary evidence base:

$\rightarrow \text{Graph Element} \rightarrow \text{Document} \rightarrow \text{Source}$.

Without this, the traceability of a real-data case is substantially reduced.

13.10. Source Attribution and Model Inference

A clear distinction must be maintained between:

what the source states

and

what the model infers.

For example, a source may report:

Group X claimed responsibility for attack Y.

A model, by contrast, may infer:

X and Z belong to the same semantic cluster.

The second claim must not be attributed to the source.

In HAAG terminology:

SOURCE_ATTRIBUTION

and

MODEL_INFERENCE

must remain separate.

13.11. Noncausal Structural Differences

If the following is identified between temporal slices:

$$\neq G(t_2),$$

this constitutes an observed structural difference.

It does not, however, imply:

$$F(t_1) \Rightarrow G(t_2).$$

Therefore, if a causal mechanism has not been established, this should be stated explicitly:

The observed structural differences between temporal slices are noncausal within the scope of the analysis performed.

This protects the paper against overinterpretation.

13.12. General Workflow for Real-World Data

For a theoretical and model-based paper, HAAG effectively proposes the following sequence:

$\text{Real Data} \rightarrow \text{ROLE} \rightarrow \text{Operational Procedure} \rightarrow \text{Observed Structure} \rightarrow \text{Allowed Interpretation} \rightarrow \text{Claim.}$

Here, ROLE establishes the upper bound on the strength of permissible conclusions.

If:

ROLE = ILLUSTRATION,

a causal-test-level conclusion is not permitted.

If:

ROLE = OPERATIONALIZATION,

the feasibility of the operationalization procedure may be claimed, but the causal validity of the theory does not automatically follow.

14. SOURCE — Search and Verification of Scientific Sources

14.1. SOURCE as a Research Operator

In HAAG, working with the literature is not reduced to the final step of “adding 20–30 references.”

A separate primitive is introduced for this purpose:

SOURCE

It may be used throughout the entire research process:

$$\begin{aligned} &\rightarrow \text{Gap}, \text{SOURCE} \rightarrow \text{Method}, \text{SOURCE} \rightarrow \text{Method}, \text{SOURCE} \rightarrow \text{Novelty}, \text{SOURCE} \\ &\rightarrow \text{Novelty}, \text{SOURCE} \rightarrow \text{Discussion}. \text{SOURCE} \rightarrow \text{Discussion}. \end{aligned}$$

Thus, the literature does not merely decorate a completed article; it participates in shaping the research trajectory.

14.2. Three SOURCE Modes

The basic construct is:

SOURCE:

MODE:

VERIFY | DISCOVER | ENRICH

The three modes correspond to three different research tasks.

14.3. VERIFY — Checking a Specific Claim

MODE:

VERIFY

is used when a claim already exists, but it is necessary to establish whether it has appropriate scientific support.

For example:

SOURCE:

MODE:

VERIFY

QUERY:

Adaptive networks can retain structural changes induced by previous dynamics.

The system must not simply find a paper containing the words “adaptive network”; it must establish:

Does the source actually support this particular claim?

14.4. DISCOVER — Exploring the Relevant Research Landscape

MODE:

DISCOVER

is used when it is necessary to investigate which concepts, theories, or classes of methods already exist.

For example:

SOURCE:

MODE:

DISCOVER

QUERY:

models combining semantic-state evolution

with adaptive network topology

The result should not be a random list of publications, but a map of the closest research directions.

This mode is especially important for:
state of knowledge
state of knowledge
and
explicit gap.

14.5. ENRICH — Expanding an Established Context

MODE:

ENRICH

is used when the principal argument already exists, but it is necessary to find:
additional comparisons;
historical sources;
alternative approaches;
reviews;
independent corroboration;
sources for the Discussion.

In other words:

VERIFY

verifies,

DISCOVER

discovers,

while

ENRICH

expands the evidential context.

14.6. QUERY — What Exactly Are We Searching For?

Structure:

QUERY:

[current claim, concept, method,
gap, or competing theory]

A query need not be limited to the article's topic.

This is fundamental.

A poor query:

semantic string theory

may retrieve only studies using similar terminology.

It is much more productive to search for functional analogues:

adaptive semantic networks

network memory after external perturbation

state-dependent information diffusion

adaptive topology and contagion

higher-order contagion with memory

path-dependent diffusion in evolving networks

This makes it possible to find studies that describe the same or a related mechanism using different terminology.

14.7. Which Sources Should Be Preferred?

HAAG specifies:

PREFER:

primary scientific publications

peer-reviewed papers

authoritative scholarly books

official datasets

publisher records

Primary scientific publications take precedence when the objective is to establish the origin of a method or a particular result.

Review articles are useful for orientation in a field, but should not always replace the primary source.

For bibliographic details, publisher records or reliable bibliographic systems are preferable.

14.8. SCHOLARLY_SYSTEMS — Scholarly Search Systems

The framework provides for:

SCHOLARLY_SYSTEMS:

Google Scholar

Crossref

Semantic Scholar

discipline-specific databases

Scopus / Web of Science when available

publisher databases

These systems perform different functions.

Google Scholar is useful for broad discovery.

Crossref is useful for checking DOIs and metadata.

Semantic Scholar supports thematic searches and exploration of relationships among publications.

Scopus and Web of Science support structured bibliographic searches when available.

Publisher databases enable final verification against the publisher's own publication record.

14.9. What SOURCE Must Return

Result:

RETURN:

verified bibliographic candidates

support relationship

persistent identifier when available

Thus, SOURCE should return more than:

Smith et al., 2024.

A structured response is required.

For example:

SOURCE_RESULT:

BIBLIOGRAPHIC_CANDIDATE:

...

SUPPORT_RELATIONSHIP:

DIRECT_SUPPORT

PERSISTENT_IDENTIFIER:

DOI: ...

STATUS:

VERIFIED

14.10. Support Relationship — The Most Important Element

The mere existence of a publication does not mean that it supports our claim.

Therefore, between:

SOURCE_iSOURCE_i

and

CLAIM_jCLAIM_j

there must be an explicit relationship:

SOURCE_i→supportsCLAIM_j.SOURCE_i \xrightarrow{supports} CLAIM_j.

Other relationships are also possible:

supports,supports, partially supports,partially\ supports, contradicts,contradicts,
provides background,provides\ background, defines method,defines\ method,
nearest competing approach.nearest\ competing\ approach.

This is considerably more reliable than mechanically appending a citation to a sentence simply because similar terms appear in the source.

14.11. Persistent Identifier

Where possible, a source should have a persistent identifier:

DOI,DOI,

or another reliable bibliographic identifier.

This helps distinguish a real publication from an LLM-generated bibliographic hallucination.

For the final reference list, the following should be checked:

authors;

title;

year;

journal or publisher;

volume;

issue;

pages;

DOI.

14.12. SOURCE and the Claims Ledger

Literature research is directly related to Chapter 11.

For example:

CLAIM_ID:

C08

TYPE:

NOVELTY

CLAIM_TEXT:

...

SUPPORT:

SOURCE_12

SOURCE_17

SOURCE_23

comparison matrix

STATUS:

PARTIALLY_SUPPORTED

ALLOWED_STRENGTH:

limited

A literature source must therefore serve a specific function in the argument, rather than merely appearing in the References.

14.13. Verifying Novelty Is Not a Search for Identical Words

The most important rule in this section is:

Novelty must be tested against the nearest competing families of methods,
not only against papers using similar terminology.

This is especially important when introducing new terminology.

Suppose an author introduces the concept:

semantic string.

A search for only:

"semantic string"

may indicate that the term is rarely used.

But this does not demonstrate the novelty of the method.

An analogous mechanism may exist under names such as:

path-dependent diffusion;

adaptive semantic network;

temporal semantic pathway;

network memory;

higher-order contagion;

sequence-dependent propagation;

adaptive topology;

state-dependent diffusion.

Therefore:

Terminological Novelty $\neq$ Methodological Novelty.

14.14. Searching for the Closest Competing Families

For novelty analysis, HAAG must move from the name of a method to its functional properties.

For example, the proposed model has:

network diffusion,
semantic states,
adaptive edges,
memory,
path dependence.

It is then necessary to search the literature for each closely related family:

adaptive networks, $F_2 = \text{complex contagion}$, $F_2 = \text{complex contagion}$, F_3
 $= \text{higher-order contagion}$, $F_3 = \text{higher-order contagion}$, F_4
 $= \text{network memory}$, $F_4 = \text{network memory}$, F_5
 $= \text{semantic network dynamics}$, $F_5 = \text{semantic network dynamics}$, F_6
 $= \text{path-dependent diffusion}$. $F_6 = \text{path-dependent diffusion}$.

The subsequent comparison concerns properties, not names:

	F_1	F_2	F_3	<i>Proposed</i>
<i>Semantic state</i>	?	?	?	+
<i>Adaptive topology</i>	?	?	?	+
<i>Memory</i>	?	?	?	+
<i>Ordered pathways</i>	?	?	?	+

Only after this analysis is it appropriate to formulate a cautious claim about the paper's contribution.

14.15. SOURCE as an Iterative Process

Scholarly searching in HAAG is not a one-off action:

→ WRITE.

It is cyclical:

SOURCE → *CLAIM* → *COMPARE* → *NEW QUERY* → *SOURCE* → *REFINE CLAIM*.

For example, an initial search shows that adaptive networks already describe structural memory.

The initial novelty claim:

“network memory is introduced for the first time”

is then rejected.

A more precise question emerges, however:

Are there models in which an earlier influence simultaneously changes semantic state and topology, and this restructuring affects the response to a subsequent influence?

This gives rise to a new SOURCE operation.

Thus, literature searching does more than identify references: it refines the scientific novelty itself.

14.16. From SOURCE to CITE

After a source has been found, it should not automatically enter the bibliography.

The logic should be:

$$[SOURCE \rightarrow VERIFY \rightarrow CLAIM-SOURCE \text{ RELATION} \rightarrow CITE.]$$

A source is first discovered, then verified, then linked to the particular claim it supports, and only then cited.

This leads HAAG to the next mechanism: a working citation mode in which, during article development, a source may initially be represented by a verifiable search query and, following bibliographic verification, converted into an ordinary citation.

14.17. Shared Logic of Chapters 12–14

The three chapters form a single integrated framework.

IMRaD defines the logic of the article:

What role may real-world data legitimately play in this chain?

SOURCE determines:

On what scientific basis the Gap, comparisons, interpretations, and Novelty are established.

SOURCE determines:

On what scientific basis the Gap, comparisons, interpretations, and Novelty are established.

Taken together:

$$\begin{array}{l} SOURCE \rightarrow \textit{State of Knowledge} \\ \rightarrow \textit{Gap} \\ \rightarrow \textit{Objective} \\ \rightarrow \textit{Methods} \\ \rightarrow \textit{Results} \\ \rightarrow \textit{Discussion} \\ \rightarrow \textit{Conclusions,} \end{array}$$

For real-world data, a parallel constraint applies:

$$[Real \text{ Data} \rightarrow Declared \text{ Role} \rightarrow Allowed \text{ Evidence} \rightarrow Allowed \text{ Claim}.]$$

Thus, HAAG does not allow a persuasive narrative to be written first and then populated with literature, data, and results. The logic must run in the opposite direction: the state of scientific knowledge identifies the gap; the gap defines the objective; the objective determines the method; the method produces the result; the result defines the permissible conclusion; and sources provide verified support for specific claims.

15. Working Citation Mode — Number + Google Scholar Search Query

15.1. Why HAAG Needs a Separate Working Citation Mode

During the preparation of a scientific article, its bibliography develops gradually. At an early stage, a researcher often already understands which claim requires a reference but has not yet identified the definitive source.

This creates a particular risk for an LLM. The system may attempt to “complete” a missing bibliographic entry with plausible authors, article titles, years, journals, or DOIs. Such a reference may appear convincing even though some of its elements are fictitious.

HAAG adopts a different approach:

First, the need for a source and a search query; then, a verified bibliographic reference.

For this purpose, a transparent working format is introduced:

[number; "Google Scholar query"]

For example:

[5; "adaptive coevolutionary networks" Gross Blasius]

This means:

At this point in the manuscript, source no. 5 is required; it must be located and verified using the specified scholarly query.

This is not yet a final bibliographic citation.

15.2. A Working Citation as an Indicator of an Evidentiary Need

In a conventional manuscript, the notation

[5]

denotes a specific source in the reference list.

In HAAG's working mode:

query

has a different meaning.

It simultaneously preserves:

the future reference number;

the reason why this source is needed;

the search strategy for finding it;

its connection to a particular claim in the manuscript.

Thus, a working citation is not merely a bibliographic marker but part of research traceability.

15.3. Why a Complete Reference Should Not Be Generated Immediately

Suppose the following statement appears in a draft:

Adaptive networks can change their topology under the influence of node-state dynamics. HAAG should not automatically generate a complete bibliographic description from the LLM's memory.

Instead, it creates:

[5; "adaptive coevolutionary networks" Gross Blasius]

SOURCE must then identify an actual candidate publication, Source Auditor must verify it, and only after that is the final form produced:

[5]

Thus:

$[5; \textit{query}] \rightarrow \textit{SOURCE} \rightarrow \textit{VERIFY} \rightarrow [5].$

15.4. First Function — Showing Why a Source Is Being Sought

A working query must be sufficiently informative that the researcher can later understand: What precise scientific function is this reference supposed to perform?

For example:

[8; "higher-order contagion" simplicial complex social contagion]

immediately shows that a source is required for a claim about higher-order contagion.

And:

[12; "network memory" adaptive temporal networks]

indicates a search for studies of memory in network dynamics.

The search query is therefore a concise description of the bibliographic intention.

15.5. Second Function — Facilitating Manual Verification

HAAG remains a human-agent framework. The researcher must therefore be able to verify SOURCE's work easily.

If the text retains:

[17; "semantic network dynamics" evolving semantic networks]

the human can copy the query into Google Scholar and independently inspect the results.

This is much more transparent than receiving from an LLM a ready-made entry such as:

Smith J., Brown A. Semantic Network Dynamics... 2021...

and then attempting to establish whether that publication exists.

15.6. Third Function — Preventing Fabricated Bibliographic Completion

Before a source has been verified, HAAG must not invent:

the article title;

the author list;

the journal;

the volume;
the page range;
the year;
the DOI.

If only the direction of the search is known, that is precisely what should be recorded:
[21; "path dependent diffusion" adaptive networks memory]

This is preferable to a bibliographically complete but unverified citation.

The principle may be formulated as follows:

An Incomplete but Transparent Citation > A Complete but Unverified Citation.

15.7. Fourth Function — Linking a Source to a Specific Claim

A citation must appear adjacent to the claim it supports.

For example:

Adaptive networks allow node dynamics and network topology to coevolve [5; "adaptive coevolutionary networks" Gross Blasius].

The relationship is then explicit:

$\leftarrow REF_5.$

An undesirable practice is to place five citations at the end of a long paragraph without specifying which claim each source supports.

HAAG seeks the following structure:

CLAIM ↔ SOURCE ↔ CITATION.

15.8. Converting a Working Citation into a Final Citation

Once a candidate has been found, the following must be verified:

whether the publication exists;

whether the authors are correct;

whether the title is correct;

whether the year is correct;

whether the journal matches;

whether the DOI is correct;

most importantly, whether the publication actually supports the adjacent claim.

Only then is:

[5; "adaptive coevolutionary networks" Gross Blasius]

converted into:

[5]

and the complete verified bibliographic entry transferred to References.

15.9. REFERENCE_ENRICHMENT — Systematic Citation Enrichment

At a later stage of article preparation, a dedicated cycle is initiated:

REFERENCE_ENRICHMENT:

FOR each major CLAIM:

SOURCE: ENRICH

create concise Scholar query

verify candidate

CITE at adjacent supported claim

Thus, the system traverses the Claims Ledger rather than merely the text.

For each substantive claim, it asks:

Does this claim require an external source?

If so:

$\rightarrow \text{SOURCE}(\text{ENRICH}) \rightarrow \text{QUERY}_i \rightarrow \text{VERIFY} \rightarrow \text{CITE}_i.$

15.10. Not Every Claim Requires an External Citation

It is important not to turn REFERENCE_ENRICHMENT into a mechanical process of adding a citation after every sentence.

For example, an original mathematical result may be supported by:

SUPPORT=DERIVATION,SUPPORT=DERIVATION,

and a numerical result by:

SUPPORT=EXP01.SUPPORT=EXP01.

In such cases, an external source is not evidence for the newly obtained result itself.

Sources are, however, needed for:

the state of the art;

established methods;

accepted definitions;

comparisons;

claims about previous results;

novelty analysis.

15.11. Final Bibliographic Verification

Before completing the article, the following is performed:

VERIFY:

no orphan references

no uncited bibliography entries

no citation that does not support its adjacent claim

no duplicate references

These are four different types of error.

Orphan reference — a citation or working bibliographic marker exists, but its role in the text has been lost.

Uncited bibliography entry — a source appears in References but is not cited anywhere in the text.

Unsupported adjacent claim — the citation is genuine, but it does not support the claim beside which it appears.

Duplicate reference — the same publication has accidentally entered the bibliography several times under different numbers or in different formats.

15.12. Complete Citation Life Cycle

In HAAG, a bibliographic citation has its own life cycle:

<i>CLAIM → NEED FOR SOURCE → SCHOLAR QUERY → CANDIDATE → VERIFY → CLAIM-SOURCE RELATION → CITE → FINAL REFERENCE.</i>

Consequently, the bibliography is not assembled as a decorative list at the end of the work, but as a network of verified relationships between manuscript claims and scientific sources.

16. FIGURE as a Full-Fledged Research Artifact

16.1. A Figure in HAAG Is Not an Article Decoration

HAAG regards a scientific figure not as a formatting element added after an article has been written, but as a distinct research artifact.

The following primitive is therefore used:

FIGURE

with the formalized structure:

FIGURE:

ID

PURPOSE

CLAIMS_SHOWN

SOURCE_STATE

INPUT_VALUES

CAPTION

PRE_FIGURE_TEXT

POST_FIGURE_EXPLANATION

STATUS

This means that every significant figure must have traceable provenance.

16.2. ID — Figure Identifier

Each figure receives a working identifier:

FIGURE_ID:

FIG03

It need not coincide with the final number, such as Figure 3.

The working ID links the figure to:

CLAIMS,RESULTS,EXPERIMENTS,SOURCE_STATE.CLAIMS,\quad RESULTS,\quad
EXPERIMENTS,\quad SOURCE_STATE.

16.3. PURPOSE — Scientific Function of a Figure

For each figure, the following must be specified:

PURPOSE:

CONCEPT |

METHOD |

RESULT |

COMPARISON |

OPERATIONALIZATION

These are fundamentally different types of visualization.

16.4. CONCEPT — Conceptual Figure

PURPOSE:

CONCEPT

shows:

the structure of a concept;

principal entities;

logical relationships;

model architecture;

system levels.

For example:

$\rightarrow \textit{Activation} \leftrightarrow \textit{Meaning} \leftrightarrow \textit{Structure}.$

Such a figure explains the model but does not, by itself, establish causality.

16.5. METHOD — Methodological Figure

PURPOSE:

METHOD

shows a procedure:

$\rightarrow Processing \rightarrow Model \rightarrow Experiment \rightarrow Metrics.$

Such a figure must correspond to the actual sequence described in Methods.

A block that is not actually part of the method must not be added merely because it makes the diagram appear more visually complete.

16.6. RESULT — Results Figure

PURPOSE:

RESULT

shows results actually obtained:

plots;

time series;

comparisons;

networks;

distributions;

numerical characteristics.

An especially strict rule applies to such figures:

$$\boxed{INPUT_VALUES \subseteq VERIFIED_NUMERICALLY.}$$

Thus, a final results figure must not be based on provisional or approximate numbers.

16.7. COMPARISON — Comparative Figure

PURPOSE:

COMPARISON

may compare:

$$\leftrightarrow Model_B, Control \leftrightarrow Experiment, Control \leftrightarrow Experiment, Before \leftrightarrow After. Before \\ \leftrightarrow After.$$

Objects being compared must be presented on compatible scales and according to consistent rules.

Visual design must not artificially amplify one side of a comparison.

16.8. OPERATIONALIZATION — Operationalization Figure

PURPOSE:

OPERATIONALIZATION

shows how theoretical concepts are mapped onto real-world observable data.

For example:

$$\rightarrow Entities \rightarrow Relations \rightarrow Semantic Graph \rightarrow Semantic Strings.$$

Such a figure must not be presented as proof of the model's causal validity when the real-data case serves only the role of OPERATIONALIZATION.

16.9. CLAIMS_SHOWN — What Exactly the Figure Claims

For each figure, explicitly specify:

CLAIMS_SHOWN:

C12

C14

or describe the claims in words.

This is particularly important because a figure often conveys a stronger implicit impression than the text.

For example, an arrow

$$\rightarrow B$$

may be interpreted as a causal link even when the data show only an association.

All substantive elements of the figure must therefore be consistent with the Claims Ledger.

16.10. SOURCE_STATE — Research State from Which the Figure Was Created

The field

SOURCE_STATE

identifies the version of the Research State on which the figure is based.

For example:

SOURCE_STATE:

S_27

This is particularly important in dynamic research.

If the model changes after the figure was created:

$$\rightarrow M_4,$$

the old figure must not automatically remain in the article.

It must be checked against the current state.

16.11. INPUT_VALUES — Data Entering the Figure

For results figures, retain:

INPUT_VALUES

For example:

INPUT_VALUES:

EXP03

metric = P_D

control = C0
model = M3.2

This makes it possible to reconstruct:

$\rightarrow \textit{Values} \rightarrow \textit{Experiment} \rightarrow \textit{Model}.$

Such traceability is especially important when an article contains many plots.

16.12. CAPTION — Caption as a Separate Textual Artifact

HAAG explicitly separates:

IMAGEIMAGE

from

CAPTION.CAPTION.

The caption must not be embedded in the image itself.

It is prepared separately:

CAPTION:

...

and inserted using the article's typesetting tools.

This allows one to:

change numbering;

adapt to a journal's style;

edit the explanation;

avoid overcrowding the graphic.

16.13. PRE_FIGURE_TEXT

The field

PRE_FIGURE_TEXT

contains text introducing the figure to the reader.

For example:

To illustrate the interaction among the activation, semantic, and structural components of the model, Figure 2 presents its conceptual diagram.

This text explains:

Why does the reader need to look at this figure at this point?

16.14. POST_FIGURE_EXPLANATION

After the figure, the following is required:

POST_FIGURE_EXPLANATION

It answers a different question:

What exactly should the reader observe in the figure, and how does this relate to the article's argument?

Thus:

→ *FIGURE* → *CAPTION* → *POST_FIGURE_EXPLANATION*.

The figure becomes an integral part of the scientific exposition rather than an isolated illustration.

16.15. Figure STATUS

A figure should have a status, for example:

STATUS:

DRAFT

After verification:

STATUS:

VERIFIED

If the model or results have changed:

STATUS:

OUTDATED

This prevents an outdated figure from remaining in a revised article.

16.16. A Figure Must Not Introduce a New Claim

The fundamental rule is:

$$\boxed{Claims(Figure) \subseteq Claims(Verified\ Research\ State)}.$$

In other words:

A figure may visualize substantiated content, but must not independently introduce a new scientific fact.

This is especially important for generative graphics.

An image-generation system may attempt to “improve” the composition by adding:

an additional node;

a new arrow;

a causal relationship;

a process name;

a numerical value.

This may be acceptable in artistic graphics.

It is not acceptable in a scientific figure.

16.17. Special Rules for RESULT Figures

Four mandatory rules apply to results figures.

First:

only VERIFIED_NUMERICALLY values.

Second, the following must be specified:

axes;

units;

legend;

scale;

data series.

Third, the design must not cosmetically exaggerate differences.

For example, a change:

$\rightarrow 1.03$

must not be made to look like a fivefold increase merely by truncating the axis without adequate explanation.

Fourth, negative and null results must be retained.

A series must not be removed from a plot merely because it does not support the expected effect.

16.18. Special Rules for Conceptual Figures

A conceptual diagram poses a different risk: graphic language itself creates meaning.

For example:

A–BA–B

may indicate a relationship.

Whereas:

$\rightarrow B$

is often perceived as a direction of influence.

More strongly still:

$A \rightarrow \text{causes } B$

HAAG therefore requires a distinction between:

schematic relation

and

demonstrated causal relation.

If causality has not been established, the figure must not visually imply it.

16.19. A Figure as Part of Provenance

Ultimately, the following chain is established for a figure:

$\text{Research State} \rightarrow \text{Claims} \rightarrow \text{Verified Values} \rightarrow \text{FIGURE} \rightarrow \text{Figure Review} \rightarrow \text{Publication}.$

Accordingly, a figure in HAAG is a first-class research artifact, as traceable as an experiment, a claim, or a source.

17. FIGURE_PROMPT — Generation of Concise Scientific Figures

17.1. Why a Separate FIGURE_PROMPT Is Needed

HAAG separates the scientific content of a figure from its graphical realization.

For this purpose, it uses:

FIGURE_PROMPT

Thus, a formal FIGURE specification is prepared first, followed by a prompt for the image-generation system:

$\text{Verified Research State} \rightarrow \text{FIGURE} \rightarrow \text{FIGURE}_{\text{PROMPT}} \rightarrow \text{Image} \rightarrow \text{Figure Reviewer}.$

This distinction is important.

The image generator must not decide independently what scientific content should be shown.

17.2. Default Style

By default, HAAG specifies:

DEFAULT STYLE:

laconic scientific journal figure

That is, a figure should resemble the graphics of a scientific article rather than advertising infographics or a presentation poster.

The principal requirements are:

conciseness;

readability;

functionality;

minimal decorative elements;

unambiguous scientific notation.

17.3. White Background

By default:

white background
unless the publisher's requirements specify otherwise.
A white background:
integrates naturally into journal typesetting;
prints well;
provides high contrast;
facilitates subsequent scaling.

17.4. Primary Graphics in Black or Dark Gray

For text and principal lines:
black/dark-gray text and lines
Color must not be used merely for decoration.
If categories can be distinguished clearly by shape or structure, additional colors are unnecessary.

17.5. Accent Colors Only When Semantically Meaningful

Restrained accent colors are acceptable when they encode distinct scientific categories.
For example:
F1,F2,ControlF_1,\quad F_2,\quad Control
may be distinguished visually.
However, color must have a semantic function:
→ Scientific Category.

Not:
→ Decoration.

17.6. Large, Readable Typography

HAAG explicitly requires:
large readable typography
A figure must remain legible when reduced to journal-column width.
Therefore, avoid:
small labels;
long sentences inside blocks;
excessive notation;
dozens of secondary details.
Prefer:

Fewer Elements + Larger Font.

17.7. Minimal Decorative Elements

The following are discouraged:

stock-photo aesthetics

3D effects

gradients

shadows

glowing elements

ornamental icons

unnecessary textures

A scientific figure should explain a structure, not showcase the capabilities of a graphics editor.

17.8. Vector-Like Geometry

Preference is given to:

vector-like geometry

that is:

crisp lines;

simple shapes;

straight, clear arrows;

consistent blocks;

well-organized networks;

clean mathematical notation.

Such figures are easier to read, scale, and reproduce.

17.9. Alignment and Whitespace

HAAG requires:

clear spacing and alignment

and

balanced whitespace

Whitespace is not wasted space.

It helps separate logical groups and establish a visual hierarchy.

17.10. Figure Language

If the article is in English:

English labels

Otherwise:

PROFILE.LANGUAGE

This ensures that the figure matches the manuscript's language.

Exceptions may include established mathematical or technical notation.

17.11. What Must Not Be Placed Inside the Image

HAAG prohibits placing the following directly within the graphical area:

figure number;

figure title;

caption;

DOI;

author name;

journal title;

explanatory paragraph.

Thus, the image itself must not contain:

Figure 3.

Architecture of ...

These elements are provided separately in the manuscript.

Consequently:

Image $\neq$ Caption.

17.12. Accuracy of Mathematical Notation

If the Research State specifies:

ΔW ,

the generator must not transform these into:

ΔV .

even if the new notation appears visually “neater.”

A scientific figure must preserve the research's exact mathematical notation.

17.13. Prohibition of Graphical Hallucinations

An especially important rule is:

Never invent numbers, labels, nodes, relations, or causal arrows.

The generator must not independently add:

numerical values;

nodes;

agent names;

new edges;
algorithmic stages;
causal arrows;
mathematical symbols.

If an object is absent from FIGURE.CONTENT, it must not appear in the final image.

17.14. FIGURE_PROMPT Structure

HAAG proposes a standard template:

Create a laconic scientific figure
for a peer-reviewed article.

SCIENTIFIC PURPOSE:

[one sentence]

CONTENT TO SHOW:

[only verified objects, variables,
relations and values]

COMPOSITION:

[panels / left-to-right flow /
network / timeline / equations]

MANDATORY LABELS:

[list]

VISUAL RULES:

white background;
large readable sans-serif labels;
thin clean lines;
vector-like schematic;
balanced whitespace;
high contrast;
minimal colors;
no decorative imagery;
no 3D;
no gradients;
no shadows.

SCIENTIFIC CONSTRAINTS:

[what must NOT be implied;
causal/temporal/metric constraints]

DO NOT INCLUDE:

figure title;

figure number;

caption;

author names;

references;

unsupported annotations;

invented values.

OUTPUT:

clean article-ready image

with no surrounding frame

and enough margin for typesetting.

Each block has a distinct purpose.

17.15. SCIENTIFIC PURPOSE

SCIENTIFIC PURPOSE:

must contain one sentence.

For example:

Show the closed interaction between activation, semantic state, and adaptive network structure.

This answers the question:

Why does this figure exist in the article?

If this cannot be answered in one sentence, the figure may be overloaded and should perhaps be divided.

17.16. CONTENT TO SHOW

This block effectively serves as an allowlist of permissible content.

For example:

CONTENT TO SHOW:

External influence $F(t)$

Activation state $A(t)$

Semantic state $S(t)$

Network structure $W(t)$

Relations:

$F \rightarrow A$

$A \leftrightarrow S$

$S \leftrightarrow W$

$W \rightarrow A$

The generator must show only the specified objects and relationships.

17.17. COMPOSITION

This block defines the spatial arrangement.

For example:

COMPOSITION:

left-to-right flow

or:

COMPOSITION:

three-panel figure

or:

COMPOSITION:

central network with four surrounding

functional blocks

This prevents the compositional decision from being delegated entirely to the generative system.

17.18. MANDATORY LABELS

For example:

MANDATORY LABELS:

External influence $F(t)$

Activation $A(t)$

Semantic state $S(t)$

Structure $W(t)$

If these labels are essential to the scientific content, they must be explicitly specified.

This reduces the risk of arbitrary rewording.

17.19. SCIENTIFIC CONSTRAINTS

This is one of the most important blocks.

For example:

SCIENTIFIC CONSTRAINTS:

The figure is conceptual.

Do not imply empirically demonstrated causality.

Bidirectional arrows indicate model coupling,

not experimentally established causal influence.

Or:

SCIENTIFIC CONSTRAINTS:

Panel B represents independent temporal slices.

Do not draw arrows between the slices.

Thus, the prompt specifies not only what to draw but also which conclusion the figure must not suggest to the reader.

17.20. DO NOT INCLUDE

This block provides a negative specification:

DO NOT INCLUDE:

figure title

figure number

caption

author names

references

unsupported annotations

invented values

Negative constraints are particularly useful in generative graphics because a model may automatically add conventional elements of “scientific infographics.”

17.21. OUTPUT

The final requirement is:

OUTPUT:

clean article-ready image

with no surrounding frame

and enough margin for typesetting

That is, the result should be suitable for direct inclusion in the article after scientific verification.

17.22. Figure Reviewer — Figures Are Not Accepted Automatically

After generation, HAAG performs:

CALL Figure Reviewer

This is a fundamental requirement.

Successful image generation does not by itself mean that the figure is suitable for a scientific article.

17.23. Scientific Fidelity

The first check is:

VERIFY:

scientific fidelity

It must establish:

whether every depicted object exists in the Research State;

whether the relationships are shown correctly;

whether any new claims have been introduced;

whether the figure corresponds to the current model version.

17.24. Label Correctness

The second check is:

label correctness

The following are checked:

names;

mathematical symbols;

indices;

superscripts and subscripts;

units;

axis labels;

variable names.

In scientific graphics, an error in a single index can change the meaning of a figure.

17.25. Readability at Journal Scale

The third check is:

readability at journal column width

A figure must not be evaluated only in a large preview window.

If, after reduction to journal-column width:

the text becomes illegible;

nodes merge;

the legend cannot be read;

arrows cannot be distinguished,

the figure must be revised.

17.26. Checking for Hallucinated Text

The next requirement is:

absence of hallucinated text

This is particularly important for image-generation systems.

Every textual element must be checked.

If, instead of:

Semantic state

the figure contains:

Semantical statte

it must not be accepted merely because its composition is attractive.

17.27. No Figure Title or Caption Inside the Image

A separate check verifies:

absence of caption/title inside image

The image retains only labels that form part of the diagram itself:

block names;

variables;

axes;

legend;

essential notation.

The figure title and explanatory caption remain in the manuscript text.

17.28. REGENERATE — Correction Rather Than Silent Acceptance

If verification fails:

IF verification fails:

REGENERATE with explicit correction

Importantly, the revised prompt must state the specific reason for rejection.

For example:

CORRECTION:

Remove the causal arrow between

temporal slices t1 and t2.

These slices were reconstructed independently.

Keep all other elements unchanged.

or:

CORRECTION:

Increase all labels by approximately 30%.

Do not add or remove scientific content.

This establishes the cycle:

$FIGURE \rightarrow FIGURE_PROMPT \rightarrow GENERATE \rightarrow REVIEW \rightarrow \begin{cases} ACCEPT, \\ REGENERATE. \end{cases}$
--

17.29. Complete Life Cycle of a Scientific Figure

In HAAG, a figure undergoes a complete cycle:

<i>Verified Research State</i>	$\rightarrow FIGURE$ $\rightarrow PURPOSE$ $\rightarrow CLAIMS_SHOWN$ $\rightarrow INPUT_VALUES$ $\rightarrow FIGURE_PROMPT$ $\rightarrow Image$ $\rightarrow Figure\ Reviewer$ $\rightarrow VERIFIED\ FIGURE.$
--------------------------------	---

A textual package is prepared in parallel:

$PRE_FIGURE_TEXT + CAPTION + POST_FIGURE_EXPLANATION.$
--

Thus, graphics cease to be the final decorative stage of article preparation.

They become controlled and traceable artifacts, just like a mathematical model, experiment, numerical result, claim, or bibliographic source.

This yields the fundamental HAAG principle for scientific visualization:

$Research\ State \rightarrow Verified\ Scientific\ Content \rightarrow Figure\ Specification \rightarrow Image,$
--

and not the other way around.

First determine what science permits us to show; only then decide how to represent it graphically.

18. Standard FIGURE_PROMPT Templates

18.1. Why HAAG Needs Standard Figure Templates

The preceding chapter defined FIGURE_PROMPT as a formalized instruction for creating a scientific figure. Different figure types, however, serve different functions and therefore cannot be generated mechanically from the same composition.

A conceptual model, a classification figure, a plot of experimental results, and a real-data operationalization diagram have different scientific semantics.

HAAG therefore provides four basic patterns:

- A. CONCEPTUAL MODEL
- B. MEMORY / TAXONOMY
- C. EXPERIMENTAL RESULT
- D. REAL-DATA OPERATIONALIZATION

These are not rigid graphical templates. Rather, they specify the logic by which scientific content is represented visually.

Pattern selection can be expressed as:

PURPOSE → FIGURE PATTERN → FIGURE_PROMPT → IMAGE.

18.2. A. CONCEPTUAL MODEL — Conceptual Model

The pattern

CONCEPTUAL MODEL

is used to present the architecture of a model, system, or interaction mechanism among its main components concisely.

For example:

$\rightarrow A \rightarrow S \rightarrow W \rightarrow A.$

Such a diagram can be more informative than an extensive verbal description because it immediately displays the principal components and their relationships.

18.3. What the Arrows Mean

The most important requirement for a CONCEPTUAL MODEL is that FIGURE_PROMPT explicitly specify the semantics of the arrows.

The same graphical construction

$\rightarrow B$

may represent fundamentally different relationships.

For example:

Dynamic dependence

$\rightarrow B(t)$

means that the state of B depends on the state of A in the mathematical model.

Information flow

$\rightarrow B$

may denote information transfer from one system component to another.

Causal influence

$\rightarrow B$

may express the claim:

a change in A causes a change in B.

This is a substantially stronger interpretation.

HAAG therefore requires:

Arrow $\Rightarrow$ Explicit Semantics.

18.4. Example FIGURE_PROMPT for a Conceptual Model

FIGURE_PROMPT:

Create a laconic scientific conceptual diagram
for a peer-reviewed article.

SCIENTIFICPURPOSE:

Show the closed interaction between external
influence, activation, semantic state,
and adaptive network structure.

CONTENT TO SHOW:

$F(t)$ — external influence

$A(t)$ — activation

$S(t)$ — semantic state

$W(t)$ — network structure

Relations:

$F \rightarrow A$

$A \rightarrow S$

$S \rightarrow W$

$W \rightarrow A$

COMPOSITION:

compact left-to-right closed-loop architecture

MANDATORY LABELS:

$F(t)$

$A(t)$

$S(t)$

$W(t)$

SCIENTIFIC CONSTRAINTS:

Arrows indicate dynamical dependencies

defined by the mathematical model.

They must not be interpreted as independently
established empirical causal relations.

DO NOT INCLUDE:

figure title

figure number

caption

author names

references

additional nodes

additional relations

invented equations or values

Thus, the generator receives not merely an instruction to “draw a cycle,” but a precise specification of what the cycle means scientifically.

18.5. A Conceptual Diagram Is Not Evidence of a Mechanism

It is particularly important not to conflate:

model architecture

with

empirically demonstrated mechanism.

If the model equations specify:

$$\rightarrow A \rightarrow S \rightarrow W,$$

the figure may represent that structure as part of the model.

But the figure itself does not demonstrate that such a causal chain exists in the system being studied.

It is therefore advisable to explicitly specify for a conceptual figure:

RELATION_TYPE:

MODEL_DEPENDENCE

or:

RELATION_TYPE:

INFORMATION_FLOW

or, only where genuinely justified:

RELATION_TYPE:

CAUSAL

18.6. B. MEMORY / TAXONOMY — Memory and Taxonomy

The second pattern:

MEMORY / TAXONOMY

is intended for cases in which several conceptually distinct categories, rather than a sequence of causal events, must be shown.

For example:

MA | MS | MF.M_A\;|\;M_S\;|\;M_F.

These may be different types of memory, mechanisms, regimes, classes, or categories.

18.7. Why an Artificial Causal Chain Is Unnecessary Here

A generative system often tends to turn a collection of objects into an attractive sequence:

$$\rightarrow M_S \rightarrow M_F.$$

Yet such a diagram would assert a particular ordering or transition between categories.
If the scientific model does not imply this, a parallel structure is preferable:

$$\boxed{M_A \quad M_S \quad M_F.}$$

That is:

COMPOSITION:

three parallel blocks

rather than:

COMPOSITION:

three-stage causal flow

18.8. Highlighting a Verified Regime

If one regime has been experimentally confirmed, it may be highlighted separately.

For example:

$$\boxed{M_S} \quad M_F.$$

However, visual emphasis must reflect the actual evidential status:

VERIFIED_REGIME:

M_S

A category must not be visually highlighted merely because it is central to the author's hypothesis.

That is:

$$\boxed{\textit{Visual Emphasis} \leftarrow \textit{Verified Research State.}}$$

18.9. Example FIGURE_PROMPT for a Taxonomy

FIGURE_PROMPT:

Create a laconic scientific taxonomy diagram.

SCIENTIFICPURPOSE:

Show three conceptually distinct forms
of memory considered in the model.

CONTENT TO SHOW:

M_A — activation memory

M_S — semantic memory

M_F — structural memory

COMPOSITION:

three equal parallel blocks

SCIENTIFICCONSTRAINTS:

The three categories are conceptually distinct.

Do not draw causal arrows between them.

Do not imply a temporal sequence.

Highlight M_S only if its verified status

is explicitly supplied in the research state.

DO NOT INCLUDE:

additional categories

causal arrows

invented hierarchy

figure title

caption

references

18.10. C. EXPERIMENTAL RESULT — Experimental Result

The third pattern:

EXPERIMENTAL RESULT

has the strictest connection to numerical audit.

It is governed by the rule:

$$\boxed{\textit{Figure Values} \subseteq \textit{VERIFIED_NUMERICALLY}.}$$

Such a figure should display only the results needed to test or explain a particular claim in the article.

18.11. One or Two Information-Rich Panels

HAAG recommends:

Prefer 1–2 panels.

For example:

$\boxed{A}$ $\boxed{B}$.

Panel A may display the dynamics:

$a(t)$,

and Panel B the summary metric:

PD.

There is no need to show five or six plots merely because many variables were computed during the experiment.

The principle is:

$$\boxed{\textit{Figure Content} = \textit{Minimum Evidence Needed for Claim}.}$$

18.12. A Figure Must Support a Specific Claim

Suppose CLAIM C21 states:

Following a prior influence, the system's response to F_2 differs from its response without that prior influence.

This does not require showing every internal parameter of the model.

What should be shown are precisely the quantities that enable readers to assess:

$R(F_2|F_1)R(F_2|F_1)$

and

$R(F_2|\emptyset).R(F_2|\text{\textbackslash varnothing})$.

Thus:

CLAIMS_SHOWN:

C21

determines which metrics belong in the figure.

18.13. A Null Effect Must Also Be Shown Honestly

An especially important rule is:

If an effect is null, visualize it honestly rather than amplifying it.

Suppose:

$R_0=1.00, R_1=1.02$.

One must not select an axis scale that makes a scientifically insignificant difference appear enormous.

The visual representation must not transform:

$$\Delta = 0.02$$

into the visual impression:

$$\Delta \gg 1.$$

If the effect is null or small, that is what the figure should convey.

18.14. Negative Results Must Not Be Omitted from a Figure

If three experiments are compared:

$EXP_1, EXP_2, EXP_3 \quad EXP_1, \quad \text{\textbackslash quad} \quad EXP_2, \quad \text{\textbackslash quad} \quad EXP_3$

and only one supports the expected effect, it is unacceptable to show only that experiment when the scientific claim concerns the entire series.

The figure must retain:

positive, negative, null positive, $\text{\textbackslash quad}$ negative, $\text{\textbackslash quad}$ null

results in accordance with the actual research state.

18.15. Example FIGURE_PROMPT for an Experimental Result

FIGURE_PROMPT:

Create a two-panel scientific result figure.

SCIENTIFICPURPOSE:

Compare the verified response to influence F2
with and without prior influence F1.

CONTENT TO SHOW:

only VERIFIED_NUMERICALLY values
from EXP03 and its registered control.

Panel A:

response trajectories

Panel B:

primary metric P_D

COMPOSITION:

two horizontally aligned panels

MANDATORY LABELS:

time

response

P_D

F2 | F1

F2 | no prior influence

SCIENTIFIC CONSTRAINTS:

Use identical scales for comparable quantities.

Do not visually exaggerate small differences.

Preserve null or negative results.

Do not add trend lines that were not calculated.

DO NOT INCLUDE:

unverified values

expected values

decorative annotations

figure title

caption

references

18.16. D. REAL-DATA OPERATIONALIZATION — Operationalization Using Real Data

The fourth pattern:

REAL-DATA OPERATIONALIZATION

is used to show how a theoretical method is applied to actual information materials.

For example:

$$\rightarrow Entities_t \rightarrow Relations_t \rightarrow G_S(t).$$

If several temporal slices exist, each should be presented as a separate pipeline when independent reconstruction forms part of the methodology.

18.17. Independent Temporal Slices — Independent Pipelines

For example:

$$\rightarrow E_1 \rightarrow G_1, D_2 \rightarrow E_2 \rightarrow G_2, D_3 \rightarrow E_3 \rightarrow G_3. D_3 \rightarrow E_3 \rightarrow G_3.$$

Graphically, these are best shown as three parallel trajectories:

D1 → Processing → G1

D2 → Processing → G2

D3 → Processing → G3

rather than:

D1 → G1 → D2 → G2 → D3 → G3

The latter may imply a causal or algorithmic dependence between temporal slices that was not actually present.

18.18. Prohibition of Causal Arrows Between Independent Slices

If the study has established only:

$$\neq G_2, G_2 \neq G_3, G_2 \neq G_3,$$

one must not automatically draw:

$$\rightarrow G_2 \rightarrow G_3.$$

Such arrows may be interpreted as:

state G_1 caused G_2 , and G_2 caused G_3 .

FIGURE_PROMPT should therefore state explicitly:

SCIENTIFIC CONSTRAINTS:

The temporal slices were reconstructed independently.

Do not draw causal arrows between G_1 , G_2 and G_3 .

Spatial alignment indicates temporal ordering only,

not causal dependence.

18.19. Example FIGURE_PROMPT for Real-Data Operationalization

FIGURE_PROMPT:

Create a laconic scientific operationalization figure.

SCIENTIFIC PURPOSE:

Show how three independent temporal document collections are transformed into semantic-network representations.

CONTENT TO SHOW:

D1 → Entity/Relation Extraction → G1

D2 → Entity/Relation Extraction → G2

D3 → Entity/Relation Extraction → G3

COMPOSITION:

three parallel horizontal pipelines

MANDATORY LABELS:

D1

D2

D3

Entity extraction

Relation extraction

G1

G2

G3

SCIENTIFIC CONSTRAINTS:

Each temporal slice is processed independently.

Temporal ordering does not imply causal dependence.

Do not draw arrows between G1, G2 and G3.

DO NOT INCLUDE:

causal arrows between slices

invented source documents

invented entities

invented numerical values

figure title

caption

references

18.20. General Principle of Standard Patterns

The four patterns can be summarized as follows:

CONCEPTUAL MODEL → Structure and Relationships
MEMORY / TAXONOMY → Categories and Modes
EXPERIMENTAL RESULT → Verified Results
REAL-DATA OPERATIONALIZATION → Mapping from Data to Model Entities

The composition must follow from the scientific function of the figure, not the other way around.

19. Figure Text Package

19.1. Why an Image Alone Is Not Enough

In HAAG, the final outcome of work on a figure is not an isolated graphics file.

For each figure, a coordinated package comprising four components is prepared:

1. PRE_FIGURE_TEXT
2. CAPTION
3. POST_FIGURE_EXPLANATION
4. FIGURE_PROMPT

Thus:

$$\text{Figure Package} = \text{PreText} + \text{Caption} + \text{PostText} + \text{GenerationPrompt}.$$

The entire package constitutes the research artifact.

19.2. PRE_FIGURE_TEXT — Text Before the Figure

The first component:

PRE_FIGURE_TEXT

is a short passage in the manuscript that integrates the figure into the article's argument.

It should explain:

Why does this figure appear at precisely this point?

For example:

The interaction among the activation, semantic, and structural components of the proposed model is summarized in Figure 2.

This text should not retell the entire figure.

Its function is:

→ *Figure.*

19.3. A Figure Reference Must Be Substantive

A weak formulation:

Figure 2 is shown below.

A better formulation:

The closed-loop interaction among external influence, activation, semantic state, and network structure is shown in Figure 2.

In the second case, the reader understands what to look for in the figure.

19.4. CAPTION — Figure Title and Caption

The second component:

CAPTION

is stored outside the image.

For example:

Figure 2. Conceptual diagram of the interaction among the activation, semantic, and structural components of the model.

The image itself should not contain:

Figure 2

or the figure title.

This is a fundamental HAAG rule:

<i>Image ≠ Caption.</i>

19.5. Requirements for CAPTION

The caption should be:

concise;

unambiguous;

informative;

consistent with the terminology of the article;

sufficient to identify the figure's function.

There is no need to turn the caption into a lengthy Discussion paragraph.

If a figure requires explanation, that explanation belongs in the next component.

19.6. POST_FIGURE_EXPLANATION — Explanation After the Figure

The third component:

POST_FIGURE_EXPLANATION

contains a brief interpretation.

It answers the question:

What exactly does the figure demonstrate in the context of the study?

For example:

The diagram depicts the closed-loop nature of the model interaction: an external influence changes the activation state; activation is associated with a change in semantic state; and semantic dynamics are associated with a change in network structure, which in turn affects subsequent activation.

However, if the figure is conceptual, one must not add:

This proves that such a mechanism exists in real-world information systems.

The figure does not demonstrate that.

19.7. The Explanation Is Limited to What Is Actually Shown

The following rule applies to POST_FIGURE_EXPLANATION:

$$\boxed{Interpretation(Figure) \subseteq Claims_Shown(Figure).}$$

That is, the explanation following the figure must not introduce a new claim absent from the verified research state itself.

This is especially important for expressions such as:

“proves”;

“confirms causality”;

“demonstrates universality”;

“shows effectiveness.”

Such expressions may be used only when the relevant interpretation is genuinely supported by the result.

19.8. FIGURE_PROMPT — Precise Specification for Creating a Figure

The fourth component:

FIGURE_PROMPT

is the exact prompt from which the figure was, or can be, created or redrawn.

Preserving FIGURE_PROMPT is important for reproducibility.

If, a year later, the figure needs to be regenerated at a higher resolution or adapted to another journal's requirements, the saved prompt makes it possible to reproduce its content.

19.9. All Four Components Must Use the Same Language

The most important rule of the Figure Text Package:

VERIFY:

the four artifacts use identical terminology, symbols and values.

For example, the following inconsistency is unacceptable:

PRE_FIGURE_TEXT:

semantic state $S(t)$

CAPTION:

meaning state $M(t)$

FIGURE:

Semantic Layer S

POST_FIGURE_EXPLANATION:

cognitive state

Even if all these terms seem closely related, they may denote different things.

The correct approach is:

$S(t)$

must remain

$S(t)$

across all four artifacts.

19.10. Consistency of Numerical Values

Numerical values must likewise agree.

If the result is:

$P_D = 1.18$,

the following discrepancy is not acceptable:

FIGURE:

1.18

CAPTION:

1.2

POST_FIGURE_TEXT:

approximately 1.3

without a specific explanation of rounding.

For quantitative figures, all values must originate from the same VERIFIED_NUMERICALLY state.

19.11. Semantic Consistency

Not only words and numbers, but also the strength of their meaning must be consistent.

For example, if FIGURE_PROMPT states:

arrows indicate model dependencies,

not empirical causality

then POST_FIGURE_EXPLANATION cannot state:

The figure confirms a causal effect of A on S.

Consequently:

Terminology + Notation + Values + Epistemic Meaning

must be consistent throughout the package.

19.12. Figure Package as a Single Artifact

For each figure, HAAG can in practice store:

FIGURE_PACKAGE:

ID:

FIG03

PRE_FIGURE_TEXT:

...

CAPTION:

...

POST_FIGURE_EXPLANATION:

...

FIGURE_PROMPT:

...

SOURCE_STATE:

S_31

STATUS:

VERIFIED

This is particularly convenient for subsequently transferring a figure:

to an article;

to a monograph;

to a presentation;

to GitHub;

to supplementary materials.

19.13. Life Cycle of the Figure Text Package

The complete cycle is:

Research State → Figure Specification → Figure Prompt → Image → Figure Review → Figure Text Package

Thus, HAAG preserves not only the final PNG, SVG, or PDF, but also the context of its scientific provenance.

20. Reproducibility and the GitHub Package

20.1. From an Article to a Reproducible Research Package

A scientific publication is a concise representation of a much larger research process.

The article itself cannot accommodate in full:

all experiment specifications;

all prompts;

negative results;

detailed parameters;

complete figure descriptions;

the history of source searches;

alternative research branches.

Therefore, for the profiles:

ARTICLE

and

SHORT_PAPER

HAAG generates an additional prompt-native supplementary repository when appropriate.

Its purpose is to enable another researcher to understand not only the final text, but also precisely how the result was obtained.

20.2. Prompt-Native Reproducibility

Traditionally, reproducibility is often associated with software code:

Repository = Code + Data.

For HAAG, this is insufficient.

In human-agent research, important artifacts also include:

prompts;

agent roles;

specifications;

mathematical models;

HUMAN decisions;

preregistration;

experimental protocols;

figure prompts;

bibliographic search queries.

Therefore:

Reproducibility ≠ Code Only.

In HAAG, reproducibility has a broader meaning:

*Reproducibility = Model + Specifications + Prompts + Experiments +
Results + Provenance.*

20.3. Basic Repository Structure

HAAG proposes:

PACKAGE:

README.md

MODEL.md

REPRODUCIBILITY.md

prompts/

experiments/

figures/

references/

real_case/ when applicable

In file-system form, this may look as follows:

project/

|

├── README.md

├── MODEL.md

├── REPRODUCIBILITY.md

|

├── prompts/

|

├── experiments/

|

├── figures/

|

├── references/

|

└── real_case/

Each component has its own function.

20.4. README.md — Research Map

The file:

README.md

should provide the researcher with a concise answer to the question:

What is this research about, and how are its materials organized?

It is advisable to include:

TITLE

OBJECTIVE

RESEARCH QUESTION

MAIN HYPOTHESES

MODEL

EXPERIMENTS

MAIN VERIFIED RESULTS

REPOSITORY STRUCTURE

REPRODUCIBILITY INSTRUCTIONS

The README should not duplicate the entire article.

It serves as a navigation map of the repository.

20.5. MODEL.md — Formal Model

MODEL.md

contains a stable description of the model:

definitions;

variables;

parameters;

equations;

assumptions;

initial conditions;

model versions;

limitations.

For example:

MODEL_VERSION:

M3.1

STATE VARIABLES:

$A(t)$

$S(t)$

W(t)

EXTERNAL INPUT:

F(t)

PARAMETERS:

...

This separates the mathematical model itself from a particular experiment.

20.6. Model Versioning

If the model has changed:

$$\rightarrow M_2 \rightarrow M_3,$$

this history should preferably be preserved.

For example:

M1.0:

initial formulation

M2.0:

added adaptive structural dynamics

M3.0:

modified relaxation term

It is then possible to trace:

$$\rightarrow MODEL_VERSION_j.$$

This is especially important for Numerical Audit.

20.7. REPRODUCIBILITY.md

The file:

REPRODUCIBILITY.md

explains how to reproduce the research.

For a no-code study, this may be a sequence of steps:

*Load Model → Apply Frozen Specification → Run → Compute Metrics →
Audit → Compare.*

It should describe:

required input materials;

the sequence of actions;

specification versions;

success criteria;

the method of verifying results.

20.8. The prompts/ Directory

The directory:

prompts/

stores prompts that were consequential for the research.

For example:

prompts/

└─ hypothesis_generation.md

└─ falsification.md

└─ source_audit.md

└─ experiment_design.md

└─ claims_audit.md

└─ figure_generation.md

In HAAG, a prompt is a reproducible research artifact if it influenced the result obtained.

20.9. Not Every Utterance Needs to Be Preserved

Prompt-native reproducibility does not mean that the entire dialogue must be archived indiscriminately.

The prompts and HUMAN decisions that affect the following must be preserved:

H, M, EXP, R, C, V.

That is, hypotheses, the model, experiments, results, claims, and figures.

20.10. The experiments/ Directory

Every significant experiment should have its own directory.

For example:

experiments/

|

└─ EXP00/

| └─ README.md

| └─ specification.md

| └─ results.md

|

└─ EXP01/

| └─ README.md

| └─ specification.md

| └─ results.md

|

```
└─ EXP02/  
  └─ README.md  
  └─ specification.md  
  └─ results.md
```

20.11. README.md for an Individual Experiment

For each experiment:

README.md

should briefly explain:

the objective;

the hypothesis;

the model version;

the experiment type;

the principal metrics;

the status.

For example:

EXPERIMENT_ID:

EXP02

PURPOSE:

Test whether prior influence F1 changes

the response to F2.

MODEL_VERSION:

M3.1

STATUS:

VERIFIED

20.12. Frozen Specification

The experiment directory must contain the specification frozen before RUN.

For example:

EXPERIMENT_ID

HYPOTHESIS

MODEL_VERSION

INITIAL_STATE

PARAMETERS

INPUT

CONTROLS

PRIMARY_METRICS
SECONDARY_METRICS
STOP_CONDITIONS
DECISION_THRESHOLDS
RANDOM_SEED
FAILURE_CRITERIA

This is, in effect, the materialized result of:

→ *FREEZE*.

20.13. results.md

The file:

results.md

should contain:

the result obtained;

metrics;

diagnostics;

warnings;

numerical audit;

the scientific status of the result.

For example:

RESULT:

...

METRICS:

...

DIAGNOSTICS:

...

AUDIT_STATUS:

VERIFIED_NUMERICALLY

INTERPRETATION:

...

It is important to distinguish:

RESULT

from

INTERPRETATION.

20.14. Experiment Status

In the repository, every experiment must have an explicit status:

status:

VERIFIED

or:

status:

PROVISIONAL

This prevents a preliminary experiment from inadvertently being treated as final.

For unsuccessful experiments, the corresponding failure state may be preserved, with an explanation of the cause.

20.15. The figures/ Directory

The directory:

figures/

stores more than just finished images.

For each figure, it is advisable to retain:

FIGURE_ID

PURPOSE

SOURCE_STATE

CLAIMS_SHOWN

INPUT_VALUES

FIGURE_PROMPT

CAPTION

PRE_FIGURE_TEXT

POST_FIGURE_EXPLANATION

STATUS

For example:

figures/

└─ FIG01/

 └─ figure.png

 └─ specification.md

 └─ prompt.md

└─ FIG02/

 └─ figure.svg

 └─ specification.md

└─prompt.md

The figure thereby becomes a reproducible artifact.

20.16. The references/ Directory

The directory:

references/

may contain:

a verified reference list;

working Scholar queries;

source audit results;

claim–source mapping.

For example:

references/

└─verified_references.md

└─scholar_queries.md

└─claim_source_map.md

This makes it possible to reproduce not only the bibliography, but also the logic by which it was assembled.

20.17. Preserving Working Scholar Queries

For example:

CLAIM_ID:

C07

QUERY:

"adaptive coevolutionary networks" Gross Blasius

FINAL_REFERENCE:

REF05

STATUS:

VERIFIED

Alternatively:

CLAIM_ID:

C12

QUERY:

"path dependent diffusion" network memory

STATUS:

UNRESOLVED

The second case is also useful: it shows that a particular bibliographic requirement remains unresolved.

20.18. The real_case/ Directory

If the article contains a real-world case, the following directory is created:

real_case/

It may contain:

a description of sources;

selection rules;

time intervals;

source IDs;

the reconstruction procedure;

results for temporal slices;

the role of the case.

For example:

ROLE:

OPERATIONALIZATION

It is especially important to preserve this status so that, in later versions of the work, a real-world case is not silently reclassified as a CAUSAL_TEST.

20.19. NO_CODE Means Executable Code Is Not Required

HAAG is a prompt-native, no-code framework.

Therefore, if the profile specifies:

NO_CODE

the repository must not artificially generate Python, R, or other code merely to appear “reproducible.”

Reproducible artifacts may include:

*Prompts + Mathematical Specifications + Natural-Language Algorithms +
Frozen Experiments.*

This is an important feature of HAAG.

20.20. A Mathematical Specification as a Reproducibility Artifact

For example, the system of equations:

$$\dot{x} = f(x, \theta, u)$$

together with:

$$x(0) = x_0,$$

termination conditions, metrics, and thresholds may constitute a complete experiment specification even without program code.

Another researcher can implement it using any suitable tool.

20.21. An Algorithm in Natural Language Can Also Be an Artifact

For example:

1. Initialize the semantic network.
2. Apply influence F1 for T1 time units.
3. Allow the system to relax until criterion R is reached.
4. Apply influence F2.
5. Measure response metric P_D.
6. Compare with the registered control.

If this procedure is unambiguous and contains all required parameters, it is a valid reproducibility artifact in a no-code profile.

20.22. Provisional and Verified Values Must Be Kept Separate

The repository must not mix:

PROVISIONAL

and

VERIFIED

numbers.

For example:

value:

1.24

status:

PROVISIONAL

must not inadvertently appear in the final table alongside:

value:

1.18

status:

VERIFIED_NUMERICALLY

without an explicit distinction.

20.23. Negative Experiments Must Also Be Preserved

HAAG explicitly requires the inclusion of:

negative experiments

and:

failed controls

This means that the repository must show more than the successful final trajectory.

For example:

experiments/

└─ EXP01_VERIFIED/

└─ EXP02_NULL/

└─ EXP03_FAILED_CONTROL/

└─ EXP04_VERIFIED/

This makes it possible to understand how the research evolved.

20.24. Preserving the Seed

For stochastic experiments, the following must be retained:

RANDOM_SEED

For example:

seed = 42.

Without the seed, exact repetition of a particular realization of a stochastic experiment may be impossible.

If multiple seeds were used, the full set must also be recorded.

20.25. Preserving Thresholds

The following must likewise be retained:

DECISION_THRESHOLDS

For example:

$P_D \geq 1.20$.

This is fundamental because changing the threshold after obtaining a result may completely alter the experiment's interpretation.

Therefore:

∈ Frozen Specification.

20.26. Preserving Figure Specifications

The repository must preserve:

figure specifications

and:

FIGURE_PROMPT

This makes it possible to verify:

which data were used to construct the figure;

which claims it represents;

what the arrows mean;
which values were used;
which constraints were given to the generator.

20.27. Preserving Scholar Queries

Working search queries are also part of provenance:

[5; "adaptive coevolutionary networks" Gross Blasius]

They therefore need not disappear after being converted into [5].

The article retains:

[5]

while the supplementary repository may preserve:

REF05

WORKING_QUERY:

"adaptive coevolutionary networks" Gross Blasius

VERIFICATION_STATUS:

VERIFIED

This makes the process of compiling the bibliography more transparent.

20.28. Restrictions on Third-Party Documents

HAAG establishes an important rule:

Do not distribute third-party source documents without appropriate permission.

Using articles, news materials, commercial databases, or other documents in research does not automatically authorize their inclusion in a GitHub repository.

Instead, one may preserve:

bibliographic records;

DOIs;

source IDs;

permitted URLs;

a description of the data acquisition procedure;

original derived data, where permitted by the applicable terms of use.

Thus:

Reproducibility ≠ Unauthorized Redistribution.

20.29. The Repository as a Representation of Research State

The structure of the GitHub package can be mapped to the HAAG state:

$S_{\square} = (T, H, Q, E, M, R, P, B, A, D, C, X, V, G).$

For example:

$\rightarrow MODEL.md, R \rightarrow experiments/, R \rightarrow experiments/, V \rightarrow figures/, V \rightarrow figures/, E$
 $\rightarrow references/, E \rightarrow references/, X \rightarrow provenancematerials, X$
 $\rightarrow provenance materials, G \rightarrow repositorystate. G \rightarrow repository state.$

The repository is therefore not an arbitrary collection of files, but a materialized part of the HAAG Research State.

20.30. The Repository Must Not Change the Science

An important principle:

$\boxed{Research \rightarrow Repository,}$

rather than:

$\rightarrow Research.$

The model must not be changed merely because a different structure is easier to present on GitHub.

Code must not be invented if the research was no-code.

Failed experiments must not be concealed to make the repository appear neater.

A complex research process must not be replaced by a simplified README if that would compromise reproducibility.

20.31. Minimum HAAG GitHub Package

In compact form:

```
HAAG-RESEARCH-PACKAGE/  
|  
├── README.md  
├── MODEL.md  
├── REPRODUCIBILITY.md  
|  
├── prompts/  
|   └── ...  
|  
├── experiments/  
|   ├── EXP01/  
|   |   ├── README.md  
|   |   ├── specification.md  
|   |   └── results.md  
|   └── ...
```

```

|
|└─ figures/
|  |└─ FIG01/
|  |  |└─ specification.md
|  |  |└─ prompt.md
|  |  └─ figure.*
|  └─ ...
|
|└─ references/
|  |└─ verified_references.md
|  |└─ scholar_queries.md
|  └─ claim_source_map.md
|
└─ real_case/
  └─ ...

```

The `real_case/` directory is added only when it is genuinely part of the research.

20.32. The Complete HAAG Reproducibility Principle

Under the traditional approach, publishing code is sometimes considered sufficient for reproducibility.

In HAAG, this is insufficient, because one must also know:

Which hypothesis was tested?

Which model version was used?

Which parameters were frozen before execution?

Which metrics were primary?

Which thresholds were defined before the result was obtained?

Which experiments were unsuccessful?

Which numbers were merely provisional?

Which HUMAN decisions changed the trajectory?

Which queries were used to search the literature?

Which data and specifications were used to create the figures?

Therefore, reproducibility in HAAG can be represented as:

$ \begin{aligned} \text{Reproducibility} = & \text{Model} \\ & + \text{Preregistration} \\ & + \text{Frozen Specifications} \\ & + \text{Prompts} \\ & + \text{Human Decisions} \\ & + \text{Experiments} \\ & + \text{Numerical Audits} \\ & + \text{Negative Results} \\ & + \text{Claims Provenance} \\ & + \text{Source Queries} \\ & + \text{Figure Specifications.} \end{aligned} $

20.33. Reproducibility of the Trajectory, Not Only the Result

The most significant feature of this approach is that HAAG seeks to preserve not only:

How to reproduce the final calculation,

but also, insofar as necessary for scientific transparency:

How the research arrived at that calculation.

The supplementary repository may therefore be regarded as a reproducible trace of the research trajectory:

$ \begin{aligned} & \text{Question} \rightarrow \text{Hypothesis} \rightarrow \text{Model} \rightarrow \text{Preregister} \rightarrow \text{Run} \rightarrow \text{Audit} \rightarrow \\ & \text{Result} \rightarrow \text{Claim} \rightarrow \text{Figure} \rightarrow \text{Publication.} \end{aligned} $

Unsuccessful transitions also remain visible:

$\rightarrow \text{Failure} \rightarrow \text{Lesson} \rightarrow \text{New Preregistration.}$

This is why the GitHub package in HAAG is not merely an appendix to the article. It serves as the structured memory of the research, preserving models, specifications, prompts, verified and negative results, sources, figures, and consequential decisions needed to understand and reproduce the scientific result.

21. Separation of Research, Writing, and Artifact Generation

21.1. Why HAAG Separates These Processes

In conventional article preparation, research, writing, figure creation, and preparation of supplementary materials often proceed simultaneously. This is natural for a human researcher, but in a human-agent environment, conflating these activities creates a risk: well-written prose may begin to shape the scientific interpretation, an attractive figure may suggest causality that the study has not established, and the convenience of repository organization may encourage an unjustified simplification of the model.

HAAG therefore explicitly distinguishes three interconnected but separate layers:

RESEARCH LAYER ↔ WRITING LAYER ↔ ARTIFACT LAYER

These layers interact, but have different functions and epistemic statuses.

The central rule is that the scientific state takes precedence: text and artifacts must reflect that state rather than be tailored to a desired outcome.

21.2. RESEARCH LAYER — The Research Layer

The first layer is:

RESEARCH LAYER:

hypotheses

models

evidence

experiments

falsification

This is where scientific investigation itself takes place.

At this layer, HAAG operates on:

H,Q,E,M,R,C.

That is, hypotheses, research questions, evidence, models, results, and claims.

21.3. Hypotheses Belong to Research, Not to the Text

A hypothesis must arise from a scientific problem:

$\rightarrow \textit{Question} \rightarrow \textit{Hypothesis}.$

The reverse process is undesirable:

Convenient Article Structure $\rightarrow$ Required Hypothesis.

For example, a third hypothesis should not be invented merely because three experiments would make the Results section appear more symmetrical.

The research determines the number and content of hypotheses.

21.4. The Model Also Belongs to the Research Layer

A mathematical model should change only for scientific reasons.

For example:

$\rightarrow M_2$

may be justified if:

a logical contradiction has been identified;

the model omits a necessary mechanism;
an experiment has revealed a structural defect;
new evidence has emerged;
HUMAN has made a reasoned decision to revise the problem formulation.
However, the model must not be changed merely because:
a new equation is easier to explain in the article.
Or because:
a different model produces a more attractive graph.

21.5. Evidence — The Evidence Base

The Research Layer also includes sources of evidence:

{derivations,experiments,sources,datasets,observations}.

These determine which claims may receive the status:

SUPPORTED

or:

PARTIALLY_SUPPORTED.

The Writing Layer cannot independently upgrade the strength of evidence.

21.6. Experiment and Falsification

In HAAG, an experiment is intended not to obtain the “desired” result, but to test a research claim.

Therefore, the Research Layer includes not only:

Experiment,Experiment,

but also:

Falsification.Falsification.

The correct logic is:

→ Test → Result → Accept/Reject/Revise.

The inadmissible logic is:

→ Desired Result → Parameter Adjustment → Confirmation.

21.7. WRITING LAYER — The Scientific Writing Layer

The second layer is:

WRITING LAYER:

IMRaD prose generated only

from accepted research state.

Its task is fundamentally different.

The Writing Layer does not discover a scientific result. It transforms an accepted research state into comprehensible scientific prose.

Formally:

$$\boxed{S_t^{accepted} \xrightarrow{WRITE} P_t.}$$

Thus, the text is a projection of the verified research state.

21.8. Direction of Dependence Between Research and Text

HAAG requires:

$$\boxed{Research \rightarrow Writing.}$$

Not:

$$\rightarrow Research.$$

This is one of the central ideas of the entire framework.

If the Results section has already been written convincingly, but Numerical Audit reveals an error, the text must be revised.

The result must not be adjusted to fit an already written paragraph.

21.9. IMRaD as a Projection of the Research State

The Writing Layer maps different parts of the Research State onto the corresponding sections of the article:

$$\begin{aligned} &\rightarrow Introduction, M, E, EXP \rightarrow Methods, M, E, EXP \rightarrow Methods, R \rightarrow Results, R \\ &\quad \rightarrow Results, C, alternatives, limitations \\ &\quad \rightarrow Discussion, C, alternatives, limitations \rightarrow Discussion, R + C \\ &\quad \rightarrow Conclusions. R + C \rightarrow Conclusions. \end{aligned}$$

Thus, IMRaD does not exist independently of the research.

It is a structured textual representation of the research findings.

21.10. REWRITE After a Change in Research State

Suppose the initial version contained:

positive result.

After the audit:

$$\rightarrow FAILED_REPRODUCTION.$$

Then:

$P_i \rightarrow \text{REWRITE} \rightarrow P_{i+1}$.

The Results, Discussion, Conclusions, and possibly the Abstract must be revised.
It is unacceptable to retain the old wording merely because it sounds better.

21.11. ARTIFACT LAYER — The Research Artifact Layer

The third layer is:

ARTIFACT LAYER:

figures

prompts

tables

supplementary repository

provenance files

This layer materializes the research in forms that complement the main text.

It includes:

figures,

tables, prompts, experimental specifications, the GitHub package, provenance files, and other supplementary materials.

21.12. A Figure Does Not Determine Interpretation

A fundamental rule applies to the Artifact Layer:

<i>Visual attractiveness $\Rightarrow$ / Scientific interpretation.</i>
--

For example, the closed loop:

$$\rightarrow A \rightarrow S \rightarrow W \rightarrow A$$

may appear persuasive as a causal mechanism.

However, if the Research Layer defines the arrows merely as dependencies within a mathematical model, the Artifact Layer must not turn them into empirically established causal relationships.

21.13. A Table Must Not Create a New Result Either

The same principle applies to tables.

If an experiment produced:

R_1, R_2, R_3 ,

a table may organize these results.

But it must not add a fourth “expected” result merely to achieve symmetry.

Therefore:

$\subseteq$ *Verified Research State*.

21.14. The Repository Must Not Dictate the Model

Another rule is:

Repository convenience $\Rightarrow$ / Model modification.

For example, if a mathematical model contains five interacting components, it should not be reduced to three simply because such a structure looks simpler in the README.

The GitHub package must be adapted to the research.

The research must not be adapted to the GitHub package.

21.15. Interaction Among the Three Layers

The layers are not isolated.

The correct interaction loop is:

Research $\rightarrow$ Writing $\rightarrow$ Artifacts

with feedback checks.

For example, Figure Reviewer may identify a relationship in a figure that lacks supporting evidence. This signals the need to return to the Research State and verify the claim.

However, the figure cannot itself “legitimize” that relationship.

21.16. General Rule for Separating the Layers

In compact form:

Research Layer: What have we established?

Writing Layer: How can we describe it accurately?

Artifact Layer: How can we reproduce, demonstrate, and preserve it?

This separation protects HAAG against one of the greatest risks of generative systems: the possibility that the persuasiveness of a presentation will displace scientific justification.

22. Final Audits

22.1. Why Separate Final Control Is Necessary

Even if individual components of a study have already passed VERIFY, HAAG conducts a comprehensive audit before final synthesis.

The reason is straightforward: locally correct components do not yet guarantee the correctness of the work as a whole.

For example:

a model may be mathematically correct but fail to address the stated objective;

numerical values may be correct but belong to an earlier model version;

sources may exist but fail to support the claims beside which they are cited;

a figure may be visually appealing yet contain a causal arrow absent from the Results.

Therefore, seven audits are performed before FINAL_SYNTHESIS:

$$A_S, A_N, A_C, A_I, A_R, A_F, A_G.$$

22.2. A. SCIENTIFIC AUDIT — Scientific Audit

The following are examined:

hypotheses

model consistency

alternative explanations

negative results

limitations

This audit assesses the scientific logic of the entire work.

22.3. Hypothesis Verification

For each hypothesis, it is necessary to establish:

$$\rightarrow Test_i \rightarrow Result_i \rightarrow Status_i.$$

No hypothesis should remain in the Introduction without having actually been investigated.

Likewise, Results should not include experiments whose connection to the research question is unclear.

22.4. Model Consistency

The internal consistency of the model is examined:

variables;

equations;

parameters;

assumptions;

initial conditions;

model versions;

interpretation of components.

Particular attention is paid to whether different sections of the article use different versions of the model.

22.5. Alternative Explanations

The final audit must ensure that the author's hypothesis is not treated as the only possible explanation without sufficient justification.

Here, the logic of the Contrarian is activated again:

$$\rightarrow \{Explanation_1, Explanation_2, \dots\}.$$

If alternatives have not been ruled out, this must be reflected in the Discussion.

22.6. Negative Results

The audit checks whether any of the following have been lost:

null results;

unsuccessful controls;

rejected hypotheses;

failed branches.

The final article need not describe every technical failure, but scientifically significant negative results must not disappear merely because they complicate the author's narrative.

22.7. Limitations

The audit examines whether the stated limitations reflect the actual boundaries of the study.

In particular:

$$\rightarrow ALLOWED_STRENGTH.$$

If the research design supports only a limited inference, the Conclusions must not present it as universal.

22.8. B. NUMERICAL AUDIT — Numerical Audit

The following are examined:

all headline values recomputed

all equations/metrics consistent

no provisional value presented as final

This is the final verification of all scientifically significant numerical values.

22.9. Headline Values

Headline values are the numerical quantities on which the principal conclusions directly depend:

$$\Delta W, \quad R, \quad accuracy, \quad p, \quad CI,$$

or other principal metrics of the study.

They must be recomputed from the frozen specification.

Not from the manuscript.

Not from the agent's memory.

Not from an earlier draft.

22.10. Consistency of Equations and Metrics

If Methods defines:

$$\frac{R(F_2|F_1)}{R(F_2|\emptyset)},$$

then Results must not silently use:

$$PD=R(F_2|F_1)-R(F_2|\emptyset).P_D=R($$

The following are also checked:

normalization;

indices;

units;

denominators;

time intervals;

aggregation rules.

22.11. PROVISIONAL Must Not Silently Become FINAL

The audit checks:

PROVISIONAL $\Rightarrow$ FINAL.PROVISIONAL $\not\Rightarrow$ FINAL.

If a numerical value was previously entered manually or estimated approximately, it must not enter the final manuscript until it receives the status:

VERIFIED_NUMERICALLY

22.12. C. CLAIMS AUDIT — Audit of Claims

The following are examined:

each major claim has provenance

claim strength matches support

Every substantive claim must have a traceable path:

$\rightarrow$ *Support*.

For example:

$\rightarrow$ EXP03,

or:

$\rightarrow$ Eq. 7,

or:

→ REF08.

22.13. Claim Strength Must Match the Evidence

If the supporting evidence permits only:

ALLOWED_STRENGTH:

limited

the text cannot state:

the proposed mechanism universally determines...

A permissible formulation might be:

the findings demonstrate the possibility of...

Thus:

$$\boxed{Strength(Claim) \leq Strength(Verified\ Support).}$$

22.14. D. IMRaD AUDIT

The entire chain is examined:

$$\boxed{Gap \rightarrow Objective \rightarrow Method \rightarrow Result \rightarrow Interpretation \rightarrow Conclusion.}$$

HAAG must traverse it both forward and backward.

For each Conclusion:

Which Result supports it?

For each Result:

Which Method produced it?

For each Method:

Which Objective does it implement?

For each Objective:

Which Gap does it address?

22.15. E. SOURCE/CITE AUDIT

The following are checked:

metadata verified

claim-source relation verified

working Scholar queries resolved

references and in-text citations synchronized

Thus, confirming that a publication exists is not sufficient.

It is also necessary to establish:

Source_i → supports Claim_j. Source_i \xrightarrow{supports} Claim_j.

22.16. Working Scholar Queries Must Be Resolved

The final manuscript must not retain:

[12; "network memory adaptive topology"]

unless it is a specifically prepared working version.

Before final synthesis, these must be converted into:

→ [number].

If no suitable candidate is found, the corresponding claim must be:

weakened;

removed;

marked as unverified;

or subjected to further source searching.

22.17. F. FIGURE AUDIT

For each figure, the following are examined:

scientific purpose

verified values

caption outside image

no hallucinated labels

no unsupported causal arrows

readability at publication scale

It is particularly important to verify consistency with:

↔ *Current Research State*.

An old figure based on an earlier model version must not automatically remain in the manuscript.

22.18. G. REPOSITORY AUDIT

The following are checked:

supplementary files agree with manuscript

experiment versions and statuses explicit

negative branches preserved

The article and repository must represent the same research.

It is unacceptable for:

Manuscript: M4, Manuscript: M_4,

while:

Repository:M3.Repository: M_3.

22.19. Final Audit Matrix

In compact form:

Final synthesis is permitted only after this entire set of checks has been passed.

23. Improved Minimal Executable HAAG Program

23.1. Why an Executable Program Is Needed

The preceding chapters defined the primitives, agents, rules for experiments, sources, figures, reproducibility, and audits.

These elements must now be integrated into a single procedure.

This is the purpose of:

Improved Minimal Executable HAAG Program

It can be regarded as a minimal program for a research process that an LLM system can interpret and execute interactively.

It does not predetermine the outcome.

It defines the rules for progressing toward an outcome.

23.2. START — Initiating the Research

The program begins with:

LABEL: START

The PROFILE is then defined.

PROFILE:

resolve publication type,

language,

venue,

figure policy,

citation working mode,

reproducibility requirements.

At this stage, HAAG must establish precisely what kind of scientific output is required.

23.3. Initial Exploration of the Problem Space

Next:

CALL: explorer

The Explorer does not write the Introduction.

It explores the space of possible research problems:

$\rightarrow \{Problem_1, \dots, Problem_n\}.$

For each candidate problem, the following are invoked:

CALL: theorist

CALL: contrarian

CALL: falsifier

SOURCE: DISCOVER nearest competing literature

Thus, each direction undergoes not only generation but also initial critical assessment.

23.4. Loop Over Candidate Problems

Formally:

FOR each candidate research problem:

CALL: theorist

CALL: contrarian

CALL: falsifier

SOURCE: DISCOVER

nearest competing literature

END FOR

The outcome is not merely a list of appealing topics but a set of directions with preliminary evaluations:

$\rightarrow \{H_i, Alternatives_i, Objections_i, Literature_i\}.$

23.5. When Several Fundamentally Different Directions Survive

If several substantively different trajectories remain after initial criticism:

IF multiple consequential directions survive:

FORK

HUMAN:

select/modify/combine/create direction

END IF

Automatic selection is undesirable at this point.

Choosing among fundamentally different scientific directions is a decision for the researcher.

23.6. HUMAN Can Do More Than Select

The human researcher can:

SELECT, MODIFY, COMBINE, CREATE. SELECT, MODIFY, COMBINE, CREATE.

For example, rather than selecting:

or B

the researcher may propose:

$C=A+B'$, $C=A+B'$,

that is, create a new hybrid branch.

This is one reason why HUMAN in HAAG is not an external approver but an operator capable of changing the research trajectory.

23.7. HYPOTHESIS — Formulating a Hypothesis

After a direction has been selected:

LABEL: HYPOTHESIS

CALL: theorist

CALL: falsifier

SOURCE: VERIFY novelty and background

The Theorist formalizes the hypothesis.

The Falsifier determines how it could be refuted.

SOURCE examines its relationship to previous research.

23.8. If the Hypothesis Fails the Initial Test

IF hypothesis fails:

 preserve rejected branch

 BACKTRACK

END IF

Both operations are important here.

First:

preserve rejected branch, preserve \ rejected \ branch,

Then:

BACKTRACK.BACKTRACK.

Thus, an unsuccessful hypothesis is not erased.

23.9. Designing the Experiment

If the hypothesis remains viable:

CALL: experimentalist

The Experimentalist determines which test can genuinely answer the research question.

If a numerical or empirical experiment is required:

PREREGISTER experiment

23.10. HUMAN Before the Experiment — Only When Necessary

Next:

HUMAN only if unresolved

consequential choices remain

This is an important restriction.

HAAG should not ask:

Which numerical solver should be chosen?

if that choice does not affect the scientific substance.

However, it should ask, for example:

Which of two fundamentally different controls should be considered the primary one?

if the answer changes the interpretation of the result.

23.11. FREEZE → RUN → AUDIT

After consequential uncertainties have been resolved:

→ *FREEZE* → *RUN* → *Numerical Auditor*.

In the program:

FREEZE experiment

RUN experiment

CALL: numerical_auditor

This protects the experiment against post-hoc adjustment.

23.12. If the Result Is Negative or Null

The program explicitly provides for:

IF failure or null result occurs:

record it as RESULT

CALL: falsifier

decide whether the original question

is answered

do not tune silently

END IF

That is:

≠ *Experiment Disappears*.

Sometimes a null result already answers the research question.

There is then no need to “rescue” the hypothesis.

23.13. Real-World Case

If a real-world case is scientifically useful:

IF real-world case is useful:

its role must first be specified:

ILLUSTRATION

OPERATIONALIZATION

VALIDATION

CAUSAL_TEST

This determines the strength of subsequent interpretation.

23.14. Real Data Must Not Be Silently Replaced with Synthetic Data

If real sources are required:

acquire real sources

or:

ask HUMAN for data source choice

but:

never substitute synthetic data silently

If real data are unavailable, the system must say so.

A synthetic experiment may be useful, but its status must remain explicitly synthetic.

23.15. Reviewer Before Article Writing

After the main research cycle:

CALL: reviewer

Then:

VERIFY:

methods

results

claims

IMRAD_READINESS

This is fundamental: the Reviewer acts before the final article has been written in full, not merely afterward.

23.16. If a Critical Problem Remains

IF critical research problem remains:

BACKTRACK to first inconsistent state

The phrase “first inconsistent state” is especially important.

The process must return not merely “a little way back,” but to the earliest state in which the error arose.

For example:

$$\leftarrow Result \leftarrow Metric \leftarrow Method.$$

If the problem lies in the metric, rewriting only the Conclusion is insufficient.

The process must return to Methods.

23.17. WRITE Begins After Research Readiness

If no critical problems remain:

WRITE CURRENT_DRAFT section by section

That is:

$$\rightarrow CURRENT_DRAFT.$$

The article is produced section by section in accordance with IMRaD.

23.18. Figure Creation Loop

For each planned figure:

FOR each planned figure:

FIGURE:

define scientific purpose and claims

FIGURE_PROMPT:

generate laconic article-image prompt

create PRE_FIGURE_TEXT,

CAPTION,

POST_FIGURE_EXPLANATION

VERIFY figure package

END FOR

Thus, each figure undergoes its own mini-cycle:

$$\rightarrow Claims \rightarrow Prompt \rightarrow Image \rightarrow Text Package \rightarrow Verify.$$

23.19. REFERENCE_ENRICHMENT

Once the main text exists, the following is initiated:

LABEL: REFERENCE_ENRICHMENT

For every substantive claim:

FOR each major claim:

SOURCE: ENRICH

create [number; "Scholar query"]

verify bibliographic candidate

CITE at correct manuscript position

END FOR

References are therefore attached to specific claims rather than distributed mechanically throughout the text.

23.20. PACKAGE — Preparing Supplementary Materials

Next:

PACKAGE:

prepare supplementary/GitHub materials

By this stage, the following should already exist:

model;

frozen experiments;

results;

figure packages;

source mappings;

provenance.

The GitHub package is therefore assembled from actual research artifacts rather than constructed retrospectively.

23.21. Final Council of Agents

Before the final decision, the following are invoked again:

CALL: reviewer

CALL: numerical_auditor

CALL: source_auditor

Their functions differ:

→ *scientific coherence, NumericalAuditor*

→ *quantitative correctness, Numerical Auditor*

→ *quantitative correctness, SourceAuditor*

→ *bibliographic and evidential correctness. Source Auditor*

→ *bibliographic and evidential correctness.*

23.22. Comprehensive VERIFY

After that:

VERIFY:

SCIENTIFIC_AUDIT

NUMERICAL_AUDIT

CLAIMS_AUDIT
FINAL_IMRAD_AUDIT
SOURCE_CITE_AUDIT
FIGURE_AUDIT
REPOSITORY_AUDIT

This constitutes the final verification gate:

→ *FINAL AUDIT* → *Publication*.

23.23. The Final Decision Belongs to HUMAN

After all audits:

HUMAN:

Authorize final synthesis
or select a state for revision.

This is one of the most important HUMAN decision points in the entire HAAG framework.

The agents may report:

all checks have been passed.

But the researcher alone decides:

AUTHORIZEAUTHORIZE

or:

REVISE.REVISE.

23.24. If Final Synthesis Is Authorized

IF AUTHORIZED:

WRITE FINAL_PUBLICATION

EXPORT final manuscript

+ figure package

+ reproducibility package

STOP

Thus, FINAL_PUBLICATION appears near the end of HAAG rather than at its beginning.

This is a fundamental architectural feature.

23.25. If the Researcher Does Not Authorize Completion

ELSE:

BACKTRACK human_selected_state

ENDIF

The researcher may say, for example:

Let us return to the formulation of H2.

or:

We need to reconsider real-data operationalization.

or:

The novelty claim is insufficiently convincing — let us return to SOURCE.

HAAG does not then restart the entire process.

It returns to a specific state.

23.26. Overall Logic of the Executable Program

In condensed form:

<i>Topic</i>	→ <i>Profile</i>
	→ <i>Explore</i>
	→ <i>Hypothesize</i>
	→ <i>Challenge</i>
	→ <i>Source</i>
	→ <i>Human Decision</i>
	→ <i>Preregister</i>
	→ <i>Run</i>
	→ <i>Audit</i>
	→ <i>Review</i>
	→ <i>Write</i>
	→ <i>Figure</i>
	→ <i>Cite</i>
	→ <i>Package</i>
	→ <i>Final Audits</i>
	→ <i>Human Authorization</i>
	→ <i>Publication.</i>

However, this sequence is not rigidly linear.

At any scientifically justified point, the following are possible:

FORK, REJECT, BACKTRACK, HUMAN, REWRITE. FORK, \quad REJECT, \quad BACKTRACK, \quad HUMAN, \quad REWRITE.

HAAG is therefore better understood not as a pipeline but as a controlled graph of transitions between research states.

24. Rules for Human Interaction

24.1. HAAG's Internal Language Must Not Become the Language of Dialogue

HAAG uses formal primitives:

CALL

VERIFY

FORK
BACKTRACK
PREREGISTER
FREEZE
SOURCE
FIGURE
HUMAN

These are needed to organize the process internally.

However, the researcher should not constantly be shown:

HUMAN CONTEXT / OPTIONS / ALLOW / RETURN.

By default, HAAG communicates with the human in concise, natural scientific language.

24.2. The Formal Structure Remains Internal

Internally, the system may use:

HUMAN:

CONTEXT:

...

PROBLEM:

...

OPTIONS:

A ...

B ...

C ...

QUESTION:

...

ALLOW:

FREE_TEXT

This may instead be presented naturally to the researcher:

We have obtained two substantively different model variants. The first retains a fixed topology and makes it possible to isolate semantic dynamics. The second introduces adaptive weights and enables the study of structural memory, but requires additional parameters and experimental controls. Which direction should we pursue?

The scientific content is the same, but the interaction is natural.

24.3. What Should Be Presented at a HUMAN Decision Point

When a decision genuinely requires the researcher's involvement, HAAG should briefly communicate five things:

the current scientific state;
why the decision matters;
two to four substantively different alternatives, where available;
the expected scientific consequences of each alternative;
one clear question.

In addition, the researcher must always retain the option of providing a free-form answer.

24.4. Current Scientific State

The human should understand exactly where the research stands.

For example:

H1 has passed conceptual verification, but two different structural adaptation models could be used to test it experimentally.

That is sufficient.

There is no need to display the entire:

St=(T,H,Q,E,M,R,P,B,A,D,C,X,V,G).

24.5. Why the Decision Matters

The consequence of the decision must then be explained.

For example:

The choice will determine whether we test only for the presence of semantic memory or also examine the reconfiguration of network topology.

Thus, the researcher should understand not only the options but also the scientific cost of the choice.

24.6. Alternatives Must Be Genuinely Different

HAAG proposes:

2–4 genuinely different alternatives

when available

The word genuinely is essential here.

It should not ask:

Option A: call the section “Methods.”

Option B: call it “Methodology.”

if this does not change the scientific content.

A meaningful choice would instead be:

A — a fixed network;

B — an adaptive network;

C — two parallel branches followed by comparison.

These are genuinely different research trajectories.

24.7. Expected Scientific Consequences

The consequences of each alternative should preferably be explained briefly.

For example:

A. Fixed topology. The model is simpler and isolates the semantic effect more effectively, but cannot be used to investigate structural memory.

B. Adaptive topology. This enables the investigation of structural memory, but increases the number of parameters and requires additional controls.

C. Both models. This allows a stronger comparison but substantially expands the experimental component.

This helps the human make a scientific decision rather than an interface-level choice.

24.8. One Clear Question

After explaining the alternatives, HAAG should ask one question.

For example:

Which model should we adopt as the primary one: fixed, adaptive, or both as parallel branches?

It is undesirable to ask simultaneously:

Which model shall we choose, which parameters shall we use, how many experiments shall we conduct, and what shall we call the article?

These are several distinct HUMAN decisions.

24.9. FREE_TEXT — The Human Is Not Restricted to the Proposed Options

Even if HAAG proposes:

A,B,C,A,\quad B,\quad C,

the human must not be limited to these options.

They may:

SELECT,REJECT,MODIFY,COMBINE,EXPAND,NARROW,REFORMULATE,RETURN,CREATE_BRANCH,ABANDON_BRANCH,NEW_DIRECTION.SELECT, REJECT, MODIFY, COMBINE, EXPAND, NARROW, REFORMULATE, RETURN, CREATE_BRANCH, ABANDON_BRANCH, NEW_DIRECTION.

For example:

None of these options. Keep the adaptive topology, but remove semantic relaxation.

This is a fully valid HUMAN decision.

24.10. Do Not Ask the Human About Inconsequential Matters

Fundamental rule:

Do not ask the human to choose between options that are merely formatting variants.

HAAG must not turn research into an endless dialogue:

Which font?

In what order should two equivalent sentences appear?

Should the block be round or square?

Should the file be named figure1.png or fig1.png?

The system should make such decisions autonomously in accordance with PROFILE.

24.11. The Principle of Adaptive Autonomy

The interaction rule can be generalized as follows:

The less a decision affects the scientific outcome, the greater HAAG's autonomy.

And conversely:

The more significantly a decision changes the research trajectory, the more important HUMAN involvement becomes.

Thus, HAAG seeks neither complete autonomy nor constant confirmation of every step.

It seeks an adaptive distribution of decisions between the agentic system and the researcher.

24.12. HUMAN as an Interface for Scientific Governance

As a result, HUMAN should be regarded not merely as a command:

ask user

but as a dedicated transition mechanism:

$$S_t \xrightarrow{\text{HUMAN decision}} S_{t+1}.$$

The decision is recorded in:

DD

and in the history:

X.X.

It is therefore possible to determine later:

Why did the research move from model M2 to M3?

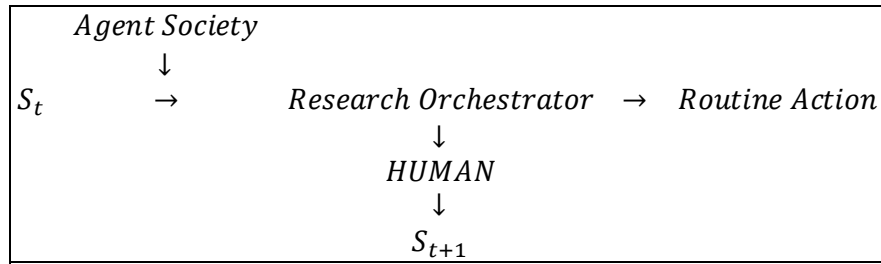
The answer may be:

Because of the researcher's decision after comparing two scientifically distinct branches.

This is part of provenance.

24.13. General Model of Human–HAAG Interaction

Architecturally, the process can be represented as follows:



If a decision is routine, the Research Orchestrator makes it autonomously.

If a decision changes the scientific trajectory, HUMAN is activated.

24.14. The Human Should Not Manage Microsteps but Retains Scientific Control

This is where the meaning of the name Human-Agentic Article Genesis becomes apparent.

HAAG does not envisage:

→ Performs every operation manually.

Nor does it envisage:

→ Independently determines the scientific objective.

Instead:

Agents explore Human shapes trajectory.

Agents can search, formalize, criticize, verify, compute, structure, and prepare artifacts.

The human intervenes when a choice determines the meaning, boundaries, or direction of the research.

24.15. Concluding Logic of Chapters 21–24

These four chapters effectively complete the architectural framework of HAAG.

Chapter 21 defines three layers:

→ *Writing* → *Artifacts*.

Chapter 22 establishes final control:

Scientific+Numerical+Claims+IMRaD+Sources+Figures+Repository.Scientific + Numerical + Claims + IMRaD + Sources + Figures + Repository.

Chapter 23 integrates primitives and agents into an executable research program:

→ *Challenge* → *Verify* → *Experiment* → *Audit* → *Write* → *Package*.

Chapter 24 then defines the human's place within this program.

Ultimately, HAAG can be represented as a closed human-agent cycle:

<i>Research State</i>	→ <i>Agentic Exploration</i>
	→ <i>Verification</i>
	→ <i>HUMAN Decision when consequential</i>
	→ <i>Experiment/Audit</i>
	→ <i>Accepted Research State</i>
	→ <i>Writing</i>
	→ <i>Scientific Artifacts</i>
	→ <i>Final Audits</i>
	→ <i>HUMAN Authorization</i>
	→ <i>Publication.</i>

Thus, HAAG does not automate the author or reduce the researcher to an operator who merely clicks “approve.” Its architecture leaves control of the scientific trajectory with the human, while delegating a substantial part of searching, critique, verification, computation, and artifact preparation to the agentic system.

25. Final Output Package of HAAG v1.1

25.1. The Publication as One Result, Not the Only Result, of Research

In a conventional scenario involving a large language model, the final outcome is often considered to be a text:

→ *Text.*

For HAAG, this is fundamentally insufficient. A scientific article is only one outcome of a much broader human–agent research trajectory.

The process generates hypotheses, alternative models, experimental results, rejected branches, researcher decisions, verified and unsupported claims, bibliographic queries, figures, specifications, and other artifacts.

Therefore, the final HAAG outcome takes the form:

<i>Final Output = Publication + Research Trace + Scientific Artifacts + Reproducibility Package.</i>
--

The final publication remains the central outcome, but it is accompanied by materials that make it possible to understand how the research arrived at the conclusions formulated therein.

25.2. Contents of the Final Output Package

In HAAG v1.1, the final package may include:

FINAL_PUBLICATION

FIGURE_LIST

FIGURE_PROMPTS

FIGURE_CAPTIONS

PRE_POST_FIGURE_TEXT
RESEARCH_GRAPH
HYPOTHESIS_TREE
AGENT_HISTORY
HUMAN_DECISION_TRACE
REJECTED_BRANCHES
FAILED_EXPERIMENTS
EXPERIMENT_REGISTRY
CLAIM_PROVENANCE
SOURCE_PROVENANCE
WORKING_SCHOLAR_QUERIES
NUMERICAL_AUDIT
REPRODUCIBILITY_PACKAGE
UNRESOLVED_PROBLEMS
FUTURE_RESEARCH_BRANCHES

Not every element must necessarily be published with the article. Some may remain internal research materials, some may be supplementary materials, and others may form part of an open repository.

What matters is that HAAG can generate and preserve them as components of the research process.

25.3. FINAL_PUBLICATION — The Final Scientific Publication

FINAL_PUBLICATION

is the final text that has passed the checks prescribed by HAAG.

For an article, it must follow the logic:

→ Objective → Method → Result → Interpretation → Conclusion.

The final publication is generated only from an accepted research state.

Therefore:

$$FINAL_PUBLICATION = Projection(Verified\ Research\ State).$$

It must not contain unverified numerical values, working bibliographic placeholders, inadequately supported claims, or results with PROVISIONAL status presented as final.

25.4. FIGURE_LIST — Figure Register

FIGURE_LIST

contains the list of all figures in the final work.

For each figure, it is advisable to retain at least:

FIGURE_ID

PURPOSE

CLAIMS_SHOWN

SOURCE_STATE

STATUS

For example:

FIGURE_ID:

FIG03

PURPOSE:

RESULT

CLAIMS_SHOWN:

C12, C13

SOURCE_STATE:

S_27

STATUS:

VERIFIED

This makes it possible to identify not only the figure number but also its scientific purpose and provenance.

25.5. FIGURE_PROMPTS — Prompts Used to Create Figures

FIGURE_PROMPTS

contains the precise instructions used to create or redraw scientific figures.

This is an important component of reproducibility.

If it subsequently becomes necessary to:

change the format;

increase the resolution;

translate labels;

adapt a figure to the requirements of another journal;

restore it after the graphic file has been lost,

the preserved FIGURE_PROMPT makes it possible to reproduce the scientific content of the figure without reconstructing it from memory.

25.6. FIGURE_CAPTIONS — Separate Caption Register

FIGURE_CAPTIONS

contains the final figure titles and captions.

The HAAG principle remains unchanged:

Caption $\notin$ Image.

The caption belongs to the manuscript, not to the graphic image.

This simplifies editing, translation, renumbering, and adaptation of the article to different publication formats.

25.7. PRE_POST_FIGURE_TEXT — Text Surrounding Figures

For each figure, the following are retained:

PRE_FIGURE_TEXT

and

POST_FIGURE_EXPLANATION.

The first component explains why the figure is introduced at that point in the article.

The second explains what follows from it.

The result is:

$\rightarrow \textit{Figure} \rightarrow \textit{PostText}.$

Together with CAPTION and FIGURE_PROMPT, these form a complete Figure Text Package.

25.8. RESEARCH_GRAPH — Graph of the Research Trajectory

RESEARCH_GRAPH

represents the structure of the research itself rather than the structure of the article.

Its nodes may include:

problems;

questions;

hypotheses;

models;

experiments;

results;

HUMAN decisions;

verification procedures;

claims.

For example:

$\rightarrow H_1 \rightarrow M_1 \rightarrow EXP_1 \rightarrow R_1 \rightarrow C_1.$

If an alternative branch emerges:

$\rightarrow \begin{cases} M_1 \rightarrow EXP_1, \\ M_2 \rightarrow EXP_2. \end{cases}$

The Research Graph makes it possible to view research as a graph of transitions rather than as a linear text.

25.9. HYPOTHESIS_TREE — Hypothesis Tree

HYPOTHESIS_TREE

records the evolution of hypotheses.

For example:

$$\rightarrow \{H_1, H_2, H_3\}.$$

After verification:

$$\rightarrow SUPPORTED, H_2 \rightarrow REJECTED, H_2 \rightarrow REJECTED, H_3 \rightarrow REFORMULATED \rightarrow H_{3a}. H_3 \\ \rightarrow REFORMULATED \rightarrow H_{3a}.$$

Such a tree shows not only the final hypothesis but also the intellectual history of its formation.

25.10. AGENT_HISTORY — History of Agent Activities

AGENT_HISTORY

records the substantive actions of the agent society.

For example:

STATE S12:

Explorer $\rightarrow$ proposed Q3

STATE S13:

Theorist $\rightarrow$ formulated H2

STATE S14:

Contrarian $\rightarrow$ proposed alternative H2a

STATE S15:

Falsifier $\rightarrow$ identified falsification criterion

STATE S16:

HUMAN $\rightarrow$ selected H2 with modification

The purpose is not to archive every utterance of an LLM, but to preserve those agent actions that changed the research state.

25.11. HUMAN_DECISION_TRACE — Trace of Researcher Decisions

The following is stored separately:

HUMAN_DECISION_TRACE.

For each consequential decision, one may record:

DECISION_ID

STATE_BEFORE

OPTIONS

DECISION

RATIONALE when supplied

CONSEQUENCE

STATE_AFTER

For example:

DECISION_ID:

D17

DECISION_BY:

HUMAN

DECISION:

Use the real-world case only as operationalization.

CONSEQUENCE:

Do not make causal validation claims.

This makes it possible to distinguish human decisions from the autonomous actions of the agentic system.

25.12. REJECTED_BRANCHES — Rejected Branches

REJECTED_BRANCHES

preserves scientifically significant trajectories that were abandoned during the research.

For example:

$$\begin{aligned} &\rightarrow ACCEPTED, B_2 \rightarrow REJECTED, B_2 \rightarrow REJECTED, B_3 \rightarrow ABANDONED. B_3 \\ &\quad \rightarrow ABANDONED. \end{aligned}$$

A rejected branch is not “waste.”

It may reveal:

which alternatives were considered;

why a particular model was not used;

which approach proved unsuitable;

where potential directions for further research lie.

25.13. FAILED_EXPERIMENTS — Failed Experiments

The following are stored separately:

FAILED_EXPERIMENTS.

These may include experiments in which:

the relaxation criterion was not met;

instability occurred;
the metric was confounded;
the control failed;
the specification was ambiguous;
the result could not be reproduced.

For such experiments, it is important to preserve:

FAILURE_TYPE

CAUSE

WHAT_WAS_LEARNED

INVALIDATED_CLAIMS

ALLOWED_NEXT_TRANSITIONS

That is:

→ Knowledge,

rather than:

→ Deletion.

25.14. EXPERIMENT_REGISTRY — Register of All Experiments

EXPERIMENT_REGISTRY

provides a consolidated representation of the experimental component.

For example:

The register shows which results correspond to which model version and hypothesis.

25.15. CLAIM_PROVENANCE — Provenance of Claims

CLAIM_PROVENANCE

links the article's final claims to their supporting evidence.

For example:

$\rightarrow EXP04, C_{21} \rightarrow Eq. (8), C_{21} \rightarrow Eq. (8), C_{25} \rightarrow REF12 + REF18. C_{25} \rightarrow REF12 + REF18.$

For each claim, the following are stored:

CLAIM_ID

CLAIM_TEXT

TYPE

SUPPORT

STATUS

ALLOWED_STRENGTH

This makes it possible to answer the question:

On what grounds did this sentence appear in the article?

25.16. SOURCE_PROVENANCE — Provenance of Sources

SOURCE_PROVENANCE

records the relationship:

↔ *Source*.

The following may be stored:

search query;

retrieved candidate;

verified metadata;

DOI or another persistent identifier;

the claim supported by the source;

verification status.

This is a substantially stronger model of bibliography than a simple References list.

25.17. WORKING_SCHOLAR_QUERIES — History of Bibliographic Searching

WORKING_SCHOLAR_QUERIES

preserves working constructs such as:

[5; "adaptive coevolutionary networks" Gross Blasius]

After verification, the final article retains:

[5]

but the original query may remain in the provenance package.

This makes it possible to reproduce the literature-search logic and recheck the bibliography.

25.18. NUMERICAL_AUDIT — Final Numerical Audit

NUMERICAL_AUDIT

contains the results of checking all substantive numerical claims.

For each headline result, it is advisable to know:

RESULT_ID

MODEL_VERSION

EXPERIMENT_ID

RECOMPUTED_VALUE

METRIC_DEFINITION

UNITS

THRESHOLD

SENSITIVITY

AUDIT_STATUS

Only results with the following status may be used in final quantitative claims:

VERIFIED_NUMERICALLY.

25.19. REPRODUCIBILITY_PACKAGE

REPRODUCIBILITY_PACKAGE

brings together the materials needed to reproduce the research.

It may include:

MODEL.md, REPRODUCIBILITY.md,

prompts/, experiments/, figures/, references/, real_case/, and other materials specified by the research profile.

For the NO_CODE profile, executable program code is not mandatory.

Reproducibility may be ensured through mathematical specifications, frozen experiments, natural-language algorithms, and precise prompts.

25.20. UNRESOLVED_PROBLEMS — Unresolved Problems

HAAG must not create the impression that completion of an article means that the scientific problem has been fully resolved.

Therefore, the following is generated separately:

UNRESOLVED_PROBLEMS.

For example:

UP01:

H3 was not tested on real data.

UP02:

Parameter sensitivity remains incomplete.

UP03:

The operationalization case does not establish causality.

This is particularly important for properly developing the Discussion and Limitations sections.

25.21. FUTURE_RESEARCH_BRANCHES — Future Research Branches

On the basis of:

unresolved problems;

rejected branches;

null results;

alternative hypotheses;

limitations;

HAAG may generate:

FUTURE_RESEARCH_BRANCHES.

This is not a generic statement:

Future research is planned.

Instead, each branch must have a substantive provenance.

For example:

$$\begin{aligned} &\leftarrow UNRESOLVED_PROBLEM_2, B_{future,2} \leftarrow REJECTED_BRANCH3, B_{future,2} \\ &\leftarrow REJECTED_BRANCH3, B_{future,3} \leftarrow LIMITATION1, B_{future,3} \\ &\leftarrow LIMITATION_1. \end{aligned}$$

Thus, Future Work follows naturally from the research trajectory.

25.22. Levels of Openness of the Final Output Package

Not all elements of the package must necessarily be public.

In practice, one can distinguish:

Internal $\rightarrow$ *Supplementary* $\rightarrow$ *Public*.

AGENT_HISTORY or a detailed HUMAN_DECISION_TRACE may remain internal.

EXPERIMENT_REGISTRY, FIGURE_PROMPTS, and REPRODUCIBILITY_PACKAGE may serve as supplementary materials.

FINAL_PUBLICATION, figures, and permitted repository materials may be made public.

Thus, HAAG ensures traceability without requiring automatic disclosure of the entire internal research process.

25.23. Final Structure of the Outcome

In summary:

<i>HAAG Final Output =</i>	<i>Publication</i>
	<i>+Figures</i>
	<i>+Research Graph</i>
	<i>+Experiment History</i>
	<i>+Human Decision Trace</i>
	<i>+Claim Provenance</i>
	<i>+Source Provenance</i>
	<i>+Numerical Audit</i>
	<i>+Reproducibility</i>
	<i>+Open Problems</i>
	<i>+Future Branches.</i>

The scientific publication is therefore accompanied by a traceable research process and, where appropriate, a prompt-native reproducibility repository.

26. Fundamental HAAG Rule

26.1. The System's Primary Question

The shortest formulation of how HAAG differs from an ordinary scientific-text generator can be expressed through the question the system asks at each step.

HAAG should not primarily ask:

“What text should be generated next?”

Its primary question is:

“What scientifically justified research transition should occur next?”

This is a fundamental change in logic.

26.2. From Text Generation to Research Transition

A conventional generative scheme often looks like this:

→ Outline → Introduction → Methods → Results → Conclusions.

The danger is obvious: when the system reaches Results, it may experience a structural “need” to write something there, regardless of whether a result actually exists.

HAAG reverses the order:

→ Research → Verified Results → Publication.

Thus, the textual structure does not produce the scientific result.

26.3. A Scientifically Justified Transition

Let the current state be:

$St.S_t$.

The system must then determine an admissible next operation:

$St \rightarrow utSt+1.S_t \rightarrow S_{t+1}$.

A transition may be motivated by:

an agent's output;

a verified source;

a mathematical derivation;

an experimental result;

an audit;

a HUMAN decision.

However, the transition must have a justification.

In conceptual form:

$\text{Reason}(St \rightarrow St+1) \neq \emptyset$.

26.4. THINK — Formulate a Scientific Possibility

The extended HAAG v1.1 cycle begins with:

THINK.THINK.

This is not simply “text generation.”

At this stage, the system may:

formulate a problem;

propose a hypothesis;

define a model;

identify a possible contradiction;

propose an experiment;

identify a research gap.

The outcome of THINK is a candidate for further investigation, not an automatically accepted claim.

26.5. CHALLENGE — Subject the Idea to Criticism

After an idea has been generated:

→ *CHALLENGE*.

HAAG deliberately does not proceed directly to writing.

The idea is passed to the Contrarian, Falsifier, Reviewer, or another critical mechanism.

The question becomes:

What might be wrong with this assumption?

This implements the principle:

Generation≠Verification.

26.6. VERIFY — Perform Verification

Next:

CHALLENGE→VERIFY.

Verification depends on the type of object.

For a hypothesis, this may involve a logical check.

For a mathematical claim, it may involve a derivation.

For a bibliographic claim, a SOURCE audit.

For a number, a Numerical Audit.

For a figure, a Figure Review.

For a claim, a Claims Audit.

Thus, VERIFY is a universal primitive, but the specific verification procedure depends on the object.

26.7. DECIDE — Make a Decision About the Research Trajectory

After verification, a decision is required:

ACCEPT?ACCEPT? REJECT?REJECT? MODIFY?MODIFY? FORK?FORK?
BACKTRACK?BACKTRACK?

If the decision has significant scientific consequences:

DECIDE→HUMAN.

If it is routine, it may be made by the Research Orchestrator.

26.8. PREREGISTER — Specify the Experiment Before Obtaining the Result

If the next transition requires an experiment:

→ *PREREGISTER*.

The following are specified:

Hypothesis,Model,Initial State,Parameters,Controls,Metrics,Thresholds,Stop
Conditions,Seed.

This separates:

what we decided to test before obtaining the result

from:

how we explained the result after obtaining it.

26.9. RUN — Execute the Experiment

After FREEZE:

→ *FREEZE* → *RUN*.

RUN produces:

RAW_RESULT

METRICS

DIAGNOSTICS

WARNINGS

However, merely obtaining a number does not yet make it a scientific result.

26.10. AUDIT — Examine the Obtained Result

Therefore, the next stage is:

→ *AUDIT*.

The audit checks whether:

the experiment followed the frozen specification;

the numbers are reproducible;

the correct model version was used;
the metrics were calculated correctly;
there is no ambiguity;
the result was not interpreted in favor of the hypothesis without justification.
Only then may the result receive the status:
VERIFIED_NUMERICALLY.

26.11. TRANSITION — Update the Research State

After the audit, the result must be integrated into the research state:

$$\rightarrow S_{t+1}.$$

For example:

$$\rightarrow PARTIALLY_SUPPORTED.$$

Or:

$$\rightarrow REJECTED.$$

Or:

$$\rightarrow FAILED.$$

This is precisely what TRANSITION means.

26.12. WRITE — Only After the Transition

Only after the research state has been updated:

$$\rightarrow WRITE.$$

The text describes the already accepted state:

$$\rightarrow P_{t+1}.$$

Therefore:

$$\boxed{\text{Research Transition precedes Text Generation.}}$$

26.13. FIGURE — Visualize the Verified State

Figures may be created after, or in parallel with, the Writing Layer:

$$\rightarrow FIGURE.$$

However, the scientific source of a figure remains the Research State, not the text.

More precisely:

$$\rightarrow \begin{cases} \text{Text}, \\ \text{Figure}. \end{cases}$$

The text and the figure are two representations of the same verified state.

26.14. VERIFY SOURCES — Check the Scientific Basis

Once claims have been formulated, their relationship to the literature must be verified:

→ *SOURCE* → *VERIFY*.

This is especially important for:

the state of the art;

novelty;

comparisons of methods;

established properties;

claims about previous findings.

26.15. CITE — Only After Verification

Only after verification:

→ *CITE*.

In other words, HAAG separates:

FindFind

and

Cite.Cite.

A retrieved source does not automatically become bibliographic support.

26.16. PACKAGE — Assemble the Results

Next:

→ *PACKAGE*.

The package includes:

manuscript;

figures;

references;

experiments;

prompts;

provenance;

reproducibility materials.

At this stage, the research becomes an integrated scientific product.

26.17. FINAL AUDIT

Before completion:

→ *FINAL AUDIT*.

The following are checked:

Scientific, Numerical, Claims, IMRaD, Sources, Figures, Repository.

Only after this may HUMAN authorize final synthesis.

26.18. The Complete HAAG v1.1 Cycle

Thus, the fundamental cycle is:

THINK → CHALLENGE → VERIFY → DECIDE → PREREGISTER when needed → RUN →
AUDIT → TRANSITION → WRITE → FIGURE → VERIFY SOURCES → CITE → PACKAGE →
FINAL AUDIT.

It is not necessarily executed only once.

After AUDIT, the following is possible:

BACKTRACK.BACKTRACK.

After VERIFY SOURCES, a new problem may emerge:

→ *NEW GAP* → *THINK*.

After FINAL AUDIT, HUMAN may return the research to an earlier state.

This is therefore not an assembly line but an iterative, governed research cycle.

26.19. Publication Emerges from the Trajectory

The main consequence of the fundamental HAAG rule can be written as:

Publication ≠ DirectGeneration(Topic).

Instead:

Publication = Synthesis(Verified Research Trajectory).

This is why HAAG differs fundamentally from one-shot prompting:

Topic → LLM Article. Topic $\xrightarrow{\text{LLM}}$ Article.

In HAAG:

Topic → Research Program → Research Trajectory → Verified Results → Publication.

27. HAAG as a Prompt-Native Domain-Specific Language

27.1. HAAG as a Prompt-Native Domain-Specific Language

At the conceptual level, HAAG can be interpreted as:

a prompt-native, domain-specific, no-code language for interactive scientific research and the production of scientific publications.

Each component of this definition has its own meaning.

Prompt-native means that the natural form of an executable specification is a structured prompt.

Domain-specific means that the language is designed not for arbitrary programming but specifically for organizing scientific research.

No-code means that a conventional programming language is not required to describe the research logic.

Interactive means that the human researcher can change the trajectory at scientifically consequential decision points.

27.2. Natural Language Defines Semantics

In HAAG, natural language is used to describe:

problems;

hypotheses;

roles;

tasks;

models;

criteria;

results;

decisions;

interpretations.

Thus:

→ Semantics.

This is why HAAG remains comprehensible to humans.

A researcher can read a HAAG program without knowing Python, Java, or another programming language.

27.3. Logical Primitives Provide Control

Natural language alone is insufficient to organize a complex research process precisely.

HAAG therefore introduces a system of primitives:

IF, ELSE, FOR, IF, \ ELSE, \ FOR, CALL, RETURN, CALL, \ RETURN,
FORK, MERGE, BACKTRACK, FORK, \ MERGE, \ BACKTRACK,
VERIFY, ACCEPT, REJECT, VERIFY, \ ACCEPT, \ REJECT,
PREREGISTER, RUN, AUDIT, PREREGISTER, \ RUN, \ AUDIT, HUMAN.HUMAN.

Thus:

Logical Primitives → Control.

As a result, natural language acquires a controlled, executable structure.

27.4. LLM Agents Provide Distributed Reasoning

Individual research functions are assigned to specialized agent roles:

Explorer, Theorist, Contrarian, Falsifier, Experimentalist, Explorer, Theorist, Contrarian, Falsifier, Experimentalist, Numerical Auditor, Source Auditor, Figure Designer, Reviewer, Synthesizer, Repository Curator, Orchestrator. Numerical\ Auditor, Source\ Auditor, Figure\ Designer, Reviewer, Synthesizer, Repository\ Curator, Orchestrator.

Together, these roles implement:

LLM Agents → Distributed Reasoning.

Importantly, distribution here primarily means functional separation of roles, not necessarily the use of multiple physically independent models.

27.5. External Scholarly Systems Provide Grounding

An LLM should not be the sole source of scientific facts.

HAAG therefore provides for interaction with external scholarly systems:

Google Scholar, Crossref, Semantic Scholar, Scopus, Web of Science, Google\ Scholar, Crossref, Semantic\ Scholar, Scopus, Web\ of\ Science, disciplinary databases, publishers' platforms, and official datasets.

Their function is:

External Scholarly Systems → Verifiable Grounding.

27.6. Numerical Tools Provide Verifiable Calculations

Large language models can work with mathematical expressions, but final numerical results must not rely solely on language-based reasoning.

Therefore:

→ *Auditable Calculations.*

This is especially important for:

modeling;

statistics;

optimization;

numerical integration;

network analysis;

processing experimental data.

The resulting output undergoes Numerical Audit.

27.7. Image Generation Systems Create Controlled Scientific Artifacts

In HAAG, generative graphics systems have a separate function:

→ *Scientific Visual Artifacts*.

However, they must not independently determine the scientific content.

That content is specified by:

→ *FIGURE* → *FIGURE_PROMPT* → *Image*.

Figure Review is then performed.

27.8. The Human Defines Scientific Intent

The most important component of the architecture is:

→ *Scientific Intent + Trajectory Control*.

The human determines what is scientifically interesting, which research boundaries are acceptable, which fundamentally different directions should be pursued, and when the research is ready for final synthesis.

This is why the framework is called:

Human-Agent Article Genesis.

27.9. Functional Distribution of HAAG Components

In compact form:

Natural Language→Semantics,Logical Primitives→Control,LLM Agents→Distributed Reasoning,Scholarly Systems→Verifiable Grounding,Numerical Tools→Auditable Calculations,Image Systems→Controlled Scientific Visuals,Human→Scientific Intent and Trajectory

It is the combination of these components that constitutes HAAG.

27.10. HAAG Is Not Merely a Prompt

Although HAAG can be stored in a single document and loaded into an LLM as a prompt framework, conceptually it is broader than an ordinary prompt.

An ordinary prompt describes:

What the model should answer.

HAAG describes:

How the research process should evolve.

Therefore, a more accurate formulation is:

$[HAAG = \textit{Prompt-Native Research Program}.]$

27.11. A Compact HAAG Model

In the original framework, the entire architecture is reduced to a very compact sequence:

$\boxed{\text{Topic} + \text{Profile}}$ $\Downarrow$ ResearchProgram $\boxed{\text{Research Program}}$ $\Downarrow$
 $\Downarrow$ $\text{Human} - \text{AgentResearchTrajectory}$ $\boxed{\text{Human-Agent Research Trajectory}}$ $\Downarrow$
 $\Downarrow$ VerifiedResults $\boxed{\text{Verified Results}}$ $\Downarrow$
 $\Downarrow$ $\text{VerifiedScientificPublication}$
 $+ \text{ReproducibilityPackage.}$ $\boxed{\text{Verified Scientific Publication} + \text{Reproducibility Package.}}$

This diagram is, in effect, a concise conceptual description of the entire HAAG v1.1.

27.12. From a Topic to a Research Program

The user may begin with a minimal request:

“Investigate string informatics.”

A PROFILE is added to it.

HAAG then performs conceptual compilation:

$\boxed{\text{HAAG} + \text{Topic} + \text{Profile} \rightarrow \text{Topic-Specific Research Program.}}$

The result is no longer a general framework, but a specific research program for a particular topic.

27.13. From a Research Program to a Human–Agent Trajectory

The program is then executed not as a rigid script, but as a sequence of transitions:

$S_0 \rightarrow S_1 \rightarrow S_2 \rightarrow \dots \rightarrow S_n$.

The transitions involve:

Agents, Tools, Sources, HUMAN.

Thus, there emerges:

-Agent Research Trajectory.

This trajectory, rather than any individual LLM response, is the central process in HAAG.

27.14. From the Trajectory to Verified Results

Dozens of candidate results may emerge during the research:

$R_1, R_2, \dots, R_n$.

After verification, however, some may have the status:

VERIFIED,

some:

PROVISIONAL,

some:

REJECTED,

and some:

FAILED.

Only results with an appropriate level of evidential support enter the final scientific argument.

27.15. From Verified Results to Publication

Only after this:

→ *Verified Claims* → *Scientific Publication*.

In parallel, the following is formed:

Reproducibility Package.

The final product therefore has two interconnected components:

Scientific Communication + Scientific Traceability.

The first is embodied in the article.

The second is embodied in the research and reproducibility package.

27.16. Conceptual Formula of HAAG

In a more expanded form, the entire concept can be represented as follows:

$\begin{aligned} HAAG = & \text{Natural Language Semantics} \\ & + \text{Logical Research Primitives} \\ & + \text{Agent Society} \\ & + \text{Research State} \\ & + \text{HUMAN Operator} \\ & + \text{Verification} \\ & + \text{Preregistration} \\ & + \text{External Scholarly Grounding} \\ & + \text{Numerical Audit} \\ & + \text{Scientific Visualization} \\ & + \text{Provenance} \\ & + \text{Reproducibility.} \end{aligned}$

This is no longer a model in which “an LLM writes an article,” but a model of interactive human-agent research followed by the preparation of a publication.

27.17. Attribution of the Concept

The Human-Agent Article Genesis (HAAG) concept, its prompt-native architecture, system of logical primitives, HUMAN state-transition operator, research-trajectory model, prompt-compilation approach, publication-profile mechanism, IMRaD control, SOURCE/CITE verification cycle, and the extensions included in version 1.1 of this document are attributed as the conceptual development of Professor Dmytro Lande.

This may also be recorded explicitly in the framework itself:

Concept and framework idea:

27.18. Version Identifier

To unambiguously refer to a specific state of the framework, the full identifier is used:

Human-Agent Article Genesis (HAAG) Framework v1.1, September 2026.

Versioning has substantive significance here.

v1.1 designates a particular revision of the architecture that includes, among other features, expanded mechanisms for preregistration, numerical audit, provenance, handling negative results, scientific figures, and the reproducibility package.

Further changes to the framework should be released as new versions, while the previous version must remain identifiable.

27.19. Concluding Definition of HAAG v1.1

Human-Agent Article Genesis (HAAG) is a human-agent, prompt-native, no-code framework for conducting interactive scientific research. Specialized LLM agents explore the space of problems, hypotheses, models, and evidence; logical primitives govern research transitions; external scholarly and computational tools enable verification of sources and results; the human defines scientific intent and governs consequential transitions in the research trajectory; and the scientific publication, together with reproducibility materials, emerges from a traceable and verified research process.

This formulation captures the central idea of HAAG v1.1: the system should not generate an article directly from a topic, but instead generate a controlled and verifiable research trajectory from which a scientific publication subsequently emerges.

Content

Abstract.....	1
Principle of Dual Interpretation.....	1
Editorial Note	1
1. Purpose and Concept of HAAG	2
1.1. From Text Generation to Conducting Research	2
1.2. Core Principle: Agents Explore, Humans Determine the Trajectory	3
1.3. HAAG as a Human-Agent System.....	3
1.4. The Article Is an Outcome, Not the Starting Point.....	4
1.5. Research Trajectories Rather Than Linear Generation	4
1.6. A Negative Result Is Still a Result.....	5

1.7. Scientific Artifacts as Independent Research Outputs	5
1.8. HAAG as a Prompt-Native, No-Code Framework	6
1.9. Traceability as a Central Requirement	7
1.10. Overall Purpose of HAAG	7
2. User Interaction with HAAG	7
2.1. Minimal Entry Requirements	7
2.2. Extended Initial Request	8
2.3. HAAG Should Not Ask Unnecessary Questions	9
2.4. The Principle of Adaptive Human Involvement	10
2.5. Example of Initiating a Real Research Project	10
2.6. Why the Initial Prompt Can Remain Short	11
3. Scientific Publication Profile	11
3.1. Why HAAG Needs a Publication Profile	11
3.2. Type of Intended Publication	13
3.3. Subject Area	13
3.4. Language of the Work	14
3.5. Target Length	14
3.6. Target Venue	15
3.7. Citation Style	15
3.8. Adaptive Human Interaction	15
3.9. Novelty Level	16
3.10. Publication Structure	16
3.11. Prohibition of Fabricated Results	17
3.12. Figure Policy	18
3.13. Reproducibility Package	18
3.14. Working Reference Search Mode	19
3.15. The Profile as a Research Contract	19
4. HAAG Architecture and Research State	20
4.1. From a Universal Framework to a Specific Study	20
4.2. Two Phases: COMPILE and EXECUTE	20
4.3. Research as a Sequence of States	21
4.4. T: Topic and Scope	22
4.5. H: Hypotheses	22

4.6. Q: Research Questions.....	22
4.7. E: Evidence and Sources.....	23
4.8. M: Methods and Models.....	23
4.9. R: Results	24
4.10. P: Current Publication Draft	24
4.11. B: Research Branches	25
4.12. A: Active Agents.....	25
4.13. D: Decisions.....	25
4.14. C: Claims and Their Provenance	26
4.15. X: Research History	26
4.16. V: Figures and Visual Materials	27
4.17. G: Reproducibility Package Status	27
4.18. Transitions Between States	28
4.19. Not Every Transition Is Permitted.....	28
4.20. Example of Research State Evolution.....	28
4.21. Architectural Meaning of the HAAG State	29
5. HAAG Primitive System.....	30
5.1. Purpose of the Primitive System	30
5.2. PROFILE — Publication and Research Profile.....	32
5.3. STRUCTURE — Structure of the Scholarly Product.....	33
Agents and Tasks	34
5.4. AGENT — Definition of a Specialized Agent.....	34
5.5. ROLE — Persistent Function of an Agent	34
5.6. TASK — Specific Assignment for an Agent	35
State and Traceability	35
5.7. STATE — Current Research State	35
5.8. TRACE — History of State Provenance.....	36
5.9. FUNCTION — Defining a Research Procedure.....	37
5.10. CALL — Invoking a Function or Agent.....	38
5.11. RETURN — Result of a Procedure	38
5.12. IF — Conditional Transition	39
5.13. ELSE — Alternative Transition	39
5.14. FOR — Repeating an Operation	40

5.15. LABEL — Naming a Stage	40
5.16. GOTO — Transition to a Defined Stage.....	40
5.17. STOP — Terminating a Process or Branch.....	41
5.18. HUMAN — A Scientifically Consequential Human Decision.....	41
5.19. FORK — Creating Alternative Branches.....	42
5.20. MERGE — Combining Compatible Branches.....	43
5.21. BACKTRACK — Returning to an Earlier State	43
5.22. SPAWN — Creating a New Agent or Process	43
5.23. FREEZE — Fixing a State or Parameters.....	44
5.24. RESUME — Resuming a Suspended Process.....	44
5.25. KILL — Terminating a Branch or Agent	45
5.26. VERIFY — Explicit Verification Procedure	45
5.27. ACCEPT — Acceptance Following Verification.....	46
5.28. REJECT — Rejection Following Verification.....	46
Generating Scientific Text.....	47
5.29. WRITE — Generating Text from a Verified State	47
5.30. REWRITE — Updating Text After a Research Change	47
Sources and Citations.....	48
5.31. SOURCE — Finding and Verifying a Source	48
5.32. CITE — Linking a Verified Source to a Claim	48
Experimental Cycle	49
5.33. PREREGISTER — Freezing an Experiment Specification Before Execution.....	49
5.34. RUN — Actual Execution.....	50
5.35. AUDIT — Independent Verification of a Result	50
Scientific Figures.....	51
5.36. FIGURE — Specification of a Scientific Figure.....	51
5.37. FIGURE_PROMPT — Prompt for Generating Scientific Graphics.....	52
Assembling the Final Product	54
5.38. PACKAGE — Assembling the Research Package	54
5.39. EXPORT — Creating External Representations	55
5.40. Relationships Among the Primitives.....	55
5.41. Functional Classification of the Primitives.....	57
5.42. HAAG as a Domain-Specific Language for Research	58

6. HAAG Agent Society	58
6.1. From a General-Purpose LLM to Functionally Distributed Research	58
6.2. Explorer — Identifying Research Problems.....	58
6.3. Theorist — Formulating Concepts, Hypotheses, and Models.....	59
6.4. Contrarian — Seeking Competing Interpretations	59
6.5. Falsifier — Actively Attempting to Reject a Hypothesis	59
6.6. Experimentalist — Designing Experiments	60
6.7. Numerical Auditor — Numerical Verification.....	60
6.8. Source Auditor — Verifying the Evidence Base.....	61
6.9. Figure Designer — Designing Scientific Figures	61
6.10. Reviewer — Internal Scientific Peer Review	62
6.11. Synthesizer — Maintaining a Consistent Research State	62
6.12. Repository Curator — Preparing Reproducibility Materials	63
6.13. Research Orchestrator — Research Coordinator.....	63
6.14. Agent Society as a System of Mutual Checks	64
6.15. Dynamic Formation of Agent Groups	64
6.16. HUMAN Is Not an Agent of Agent Society	65
6.17. General Functional Structure of Agent Society	65
6.18. General Execution Scheme	65
7. HUMAN as a Full-Fledged Operator of Scientific Research	66
7.1. The Role of the Human in the HAAG Architecture.....	66
7.2. Why a Fully Autonomous Trajectory Is Undesirable.....	67
7.3. Standard Structure of HUMAN	67
7.4. CONTEXT — Minimal Scientific Context.....	68
7.5. PROBLEM — Why a Human Decision Is Needed	68
7.6. OPTIONS — Substantive Alternatives.....	69
7.7. QUESTION — One Clear Scientific Question	69
7.8. FREE_TEXT — The Human Is Not Limited to Proposed Alternatives.....	70
7.9. Permitted HUMAN Actions	70
7.10. A HUMAN Decision Must Change STATE	72
7.11. When HUMAN Is Mandatory	72
7.12. When HUMAN Need Not Be Invoked.....	74
7.13. Adaptive Frequency of HUMAN.....	75

8. Experiment Preregistration and Freezing	75
8.1. Why HAAG Needs a Preregistration Mechanism.....	75
8.2. PREREGISTER — Specification Before Obtaining Results	75
8.3. EXPERIMENT_ID — Experiment Identifier	76
8.4. HYPOTHESIS — What Exactly Is Being Tested	76
8.5. MODEL_VERSION — Which Model Is Being Tested.....	77
8.6. INITIAL_STATE — Initial State.....	77
8.7. PARAMETERS — Model Parameters.....	77
8.8. INPUT / INFLUENCE — Experimental Influence.....	78
8.9. CONTROLS — Control Scenarios.....	78
8.10. PRIMARY_METRICS — Primary Metrics.....	79
8.11. SECONDARY_METRICS — Secondary Characteristics	79
8.12. RELAXATION / STOP CONDITIONS — Termination Conditions.....	79
8.13. DECISION_THRESHOLDS — Interpretation Thresholds.....	80
8.14. RANDOM_SEED — Controlling Randomness	80
8.15. FAILURE_CRITERIA — When an Experiment Is Invalid	81
8.16. FREEZE — Freezing the Specification.....	81
8.17. The Main Rule After RUN	82
8.18. What to Do If a Model Defect Is Discovered After Execution Begins	82
8.19. Distinguishing Error Correction from Result Tuning	83
8.20. A Negative Result Is Not an Error	83
8.21. A Negative Result Can Change the Theory	84
8.22. AUDIT After Execution.....	84
8.23. Complete Experiment Life Cycle.....	85
8.24. Relationship Between PREREGISTER and HUMAN.....	86
8.25. Two Modes: Confirmatory and Exploratory.....	86
8.26. Role of TRACE in Preregistration	87
8.27. Preregistration as a Mechanism of HAAG's Scientific Memory.....	88
8.28. Shared Logic of HUMAN and PREREGISTRATION.....	88
9. RUN and Numerical Audit	89
9.1. From an Experiment Specification to an Actual Result	89
9.2. RUN Must Execute the Frozen Specification.....	90
9.3. Types of RUN Objects.....	90

9.4. What RUN Must Return	91
9.5. RAW_RESULT — Primary Result	92
9.6. METRICS — Computed Metrics	92
9.7. DIAGNOSTICS — Execution Diagnostics.....	93
9.8. WARNINGS — Warnings.....	93
9.9. Why a Separate AUDIT Is Needed After RUN.....	94
9.10. Recomputing Principal Results	94
9.11. Checking Units, Indexing, and Normalization	95
9.12. Checking the Model Version	95
9.13. Provisional Numbers and Tool-Computed Numbers	96
9.14. Numerical Sensitivity Testing	96
9.15. Recording Implementation Ambiguities	97
9.16. The Auditor Must Not Help the Hypothesis	97
9.17. Numerical Audit Statuses.....	98
9.18. VERIFIED_NUMERICALLY	98
9.19. PROVISIONAL.....	98
9.20. FAILED_REPRODUCTION.....	99
9.21. AMBIGUOUS_SPECIFICATION.....	99
9.22. The Fundamental Rule for Numerical Claims	99
10. Failure as a Legitimate Research State	100
10.1. Why Failed Experiments Must Not Be Deleted	100
10.2. FAILURE Does Not Always Mean the Same Thing	100
10.3. Failure to Reach Relaxation	100
10.4. Model Instability	101
10.5. Confounded Metric	101
10.6. Real-World Data Do Not Support a Causal Interpretation	101
10.7. Null Result for a Hypothesis	102
10.8. Structure of a Failure Record	103
10.9. FAILURE_TYPE	103
10.10. CAUSE.....	103
10.11. WHAT_WAS_LEARNED	104
10.12. WHICH_CLAIMS_ARE_INVALIDATED	104
10.13. WHICH_NEXT_TRANSITIONS_ARE_ALLOWED	104

10.14. A Failed Control and a New Experiment	105
10.15. FAILURE as a Mechanism of Research Evolution	105
11. Claims Ledger and Claim Provenance	105
11.1. Why a Claims Ledger Is Necessary	105
11.2. Basic Record Structure	106
11.3. CLAIM_ID — Unique Identifier	107
11.4. CLAIM_TEXT — The Claim Itself	107
11.5. Claim Types	107
11.6. DEFINITION — Definition	107
11.7. DERIVATION — Derived Result	108
11.8. NUMERICAL_RESULT — Numerical Result	108
11.9. EMPIRICAL_OBSERVATION — Empirical Observation	108
11.10. INTERPRETATION — Interpretation	109
11.11. NOVELTY — Novelty Claim	109
11.12. LIMITATION — Statement of Limitations	110
11.13. SUPPORT — The Basis of a Claim	110
11.14. Claim Statuses	111
11.15. ALLOWED_STRENGTH — How Strongly May a Claim Be Worded?	112
11.16. Descriptive — Descriptive Level	112
11.17. Limited — Restricted Claim	112
11.18. General — General Claim	112
11.19. Fundamental Rule of the Claims Ledger	113
11.20. CLAIMS_AUDIT Before Final Writing	113
11.21. Claims Ledger as a Knowledge-Provenance Graph	114
11.22. Relationship Between RUN, FAILURE, and the CLAIMS LEDGER	115
Which CLAIMS Are No Longer Permitted	
and	
Which Transitions Are Allowed Next.	115
11.23. General Logic of the Three Chapters	115
12. Mandatory IMRaD Logic	116
12.1. IMRaD in HAAG Is More Than an Article Format	116

12.2. INTRODUCTION — Formulating the Scientific Problem	117
12.3. Context — Research Context	117
12.4. State of Knowledge — What Is Already Known	117
12.5. Explicit Gap — An Explicitly Formulated Research Gap	118
12.6. Objective — Research Objective.....	118
12.7. Research Question / Hypothesis	119
12.8. Contribution Boundaries — Limits of the Contribution	119
12.9. METHODS — How the Result Was Obtained	119
12.10. Definitions and Assumptions.....	120
12.11. Data / Materials	120
12.12. Mathematical Model	120
12.13. Algorithms and Procedures.....	121
12.14. Preregistered Experiments	121
12.15. Metrics and Thresholds.....	121
12.16. Reproducibility Conditions.....	122
12.17. RESULTS — Only What Was Actually Obtained.....	122
12.18. Only Actually Obtained Results	122
12.19. Negative and Null Results	122
12.20. Diagnostics and Failed Controls	123
12.21. Tables and Figures	123
12.22. Validation	123
12.23. DISCUSSION — What the Results Mean.....	123
12.24. Interpretation	124
12.25. Comparison with Literature.....	124
12.26. Alternative Explanations	124
12.27. What the Model Does NOT Show	125
12.28. Limitations	125
12.29. Conditions for Future Generalization	125
12.30. CONCLUSIONS — Only What Results Support.....	125
12.31. IMRaD as a System for Checking Article Integrity	126
13. Operationalization Rule for Real-World Data.....	126
13.1. Why the Role of a Real-World Case Must Be Defined.....	126
13.2. ILLUSTRATION.....	127

13.3. OPERATIONALIZATION.....	127
13.4. EXTERNAL_VALIDATION.....	127
13.5. CAUSAL_TEST.....	128
13.6. Prohibition on Implicitly Upgrading the Role	128
13.7. Independent Reconstruction of Temporal Slices	128
13.8. Publication Date and Event Date	129
13.9. Preserving Source IDs	129
13.10. Source Attribution and Model Inference	129
13.11. Noncausal Structural Differences.....	130
13.12. General Workflow for Real-World Data.....	130
14. SOURCE — Search and Verification of Scientific Sources	130
14.1. SOURCE as a Research Operator	130
14.2. Three SOURCE Modes.....	131
14.3. VERIFY — Checking a Specific Claim.....	131
14.4. DISCOVER — Exploring the Relevant Research Landscape	131
14.5. ENRICH — Expanding an Established Context.....	132
14.6. QUERY — What Exactly Are We Searching For?	132
14.7. Which Sources Should Be Preferred?	133
14.8. SCHOLARLY_SYSTEMS — Scholarly Search Systems	133
14.9. What SOURCE Must Return.....	134
14.10. Support Relationship — The Most Important Element.....	134
14.11. Persistent Identifier	135
14.12. SOURCE and the Claims Ledger	135
14.13. Verifying Novelty Is Not a Search for Identical Words.....	136
14.14. Searching for the Closest Competing Families	136
14.15. SOURCE as an Iterative Process	137
14.16. From SOURCE to CITE.....	138
14.17. Shared Logic of Chapters 12–14.....	138
15. Working Citation Mode — Number + Google Scholar Search Query	139
15.1. Why HAAG Needs a Separate Working Citation Mode	139
15.2. A Working Citation as an Indicator of an Evidentiary Need	139
15.3. Why a Complete Reference Should Not Be Generated Immediately	139
15.4. First Function — Showing Why a Source Is Being Sought	140

15.5. Second Function — Facilitating Manual Verification	140
15.6. Third Function — Preventing Fabricated Bibliographic Completion	140
15.7. Fourth Function — Linking a Source to a Specific Claim.....	141
15.8. Converting a Working Citation into a Final Citation	141
15.9. REFERENCE_ENRICHMENT — Systematic Citation Enrichment.....	142
15.10. Not Every Claim Requires an External Citation	142
15.11. Final Bibliographic Verification.....	142
15.12. Complete Citation Life Cycle	143
16. FIGURE as a Full-Fledged Research Artifact.....	143
16.1. A Figure in HAAG Is Not an Article Decoration	143
16.2. ID — Figure Identifier	144
16.3. PURPOSE — Scientific Function of a Figure	144
16.4. CONCEPT — Conceptual Figure.....	144
16.5. METHOD — Methodological Figure.....	144
16.6. RESULT — Results Figure.....	145
16.7. COMPARISON — Comparative Figure.....	145
16.8. OPERATIONALIZATION — Operationalization Figure.....	145
16.9. CLAIMS_SHOWN — What Exactly the Figure Claims	146
16.10. SOURCE_STATE — Research State from Which the Figure Was Created	146
16.11. INPUT_VALUES — Data Entering the Figure	146
16.12. CAPTION — Caption as a Separate Textual Artifact.....	147
16.13. PRE_FIGURE_TEXT	147
16.14. POST_FIGURE_EXPLANATION.....	147
16.15. Figure STATUS	148
16.16. A Figure Must Not Introduce a New Claim	148
16.17. Special Rules for RESULT Figures	149
16.18. Special Rules for Conceptual Figures	149
16.19. A Figure as Part of Provenance.....	150
17. FIGURE_PROMPT — Generation of Concise Scientific Figures.....	150
17.1. Why a Separate FIGURE_PROMPT Is Needed	150
17.2. Default Style.....	150
17.3. White Background	150
17.4. Primary Graphics in Black or Dark Gray	151

17.5. Accent Colors Only When Semantically Meaningful	151
17.6. Large, Readable Typography.....	151
17.7. Minimal Decorative Elements.....	152
17.8. Vector-Like Geometry	152
17.9. Alignment and Whitespace	152
17.10. Figure Language	152
17.11. What Must Not Be Placed Inside the Image	153
17.12. Accuracy of Mathematical Notation.....	153
17.13. Prohibition of Graphical Hallucinations	153
17.14. FIGURE_PROMPT Structure.....	154
17.15. SCIENTIFIC PURPOSE.....	155
17.16. CONTENT TO SHOW.....	155
17.17. COMPOSITION.....	156
17.18. MANDATORY LABELS.....	156
17.19. SCIENTIFIC CONSTRAINTS	156
17.20. DO NOT INCLUDE	157
17.21. OUTPUT.....	157
17.22. Figure Reviewer — Figures Are Not Accepted Automatically	157
17.23. Scientific Fidelity.....	158
17.24. Label Correctness	158
17.25. Readability at Journal Scale.....	158
17.26. Checking for Hallucinated Text.....	158
17.27. No Figure Title or Caption Inside the Image	159
17.28. REGENERATE — Correction Rather Than Silent Acceptance.....	159
17.29. Complete Life Cycle of a Scientific Figure	160
18. Standard FIGURE_PROMPT Templates	160
18.1. Why HAAG Needs Standard Figure Templates.....	160
18.2. A. CONCEPTUAL MODEL — Conceptual Model	161
18.3. What the Arrows Mean	161
18.4. Example FIGURE_PROMPT for a Conceptual Model	162
18.5. A Conceptual Diagram Is Not Evidence of a Mechanism.....	163
18.6. B. MEMORY / TAXONOMY — Memory and Taxonomy	163
18.7. Why an Artificial Causal Chain Is Unnecessary Here	163

18.8. Highlighting a Verified Regime.....	164
18.9. Example FIGURE_PROMPT for a Taxonomy.....	164
18.10. C. EXPERIMENTAL RESULT — Experimental Result.....	165
18.11. One or Two Information-Rich Panels	165
18.12. A Figure Must Support a Specific Claim	166
18.13. A Null Effect Must Also Be Shown Honestly	166
18.14. Negative Results Must Not Be Omitted from a Figure	166
18.15. Example FIGURE_PROMPT for an Experimental Result	167
18.16. D. REAL-DATA OPERATIONALIZATION — Operationalization Using Real Data ..	167
18.17. Independent Temporal Slices — Independent Pipelines.....	168
18.18. Prohibition of Causal Arrows Between Independent Slices	168
18.19. Example FIGURE_PROMPT for Real-Data Operationalization.....	168
18.20. General Principle of Standard Patterns.....	169
19. Figure Text Package	170
19.1. Why an Image Alone Is Not Enough	170
19.2. PRE_FIGURE_TEXT — Text Before the Figure.....	170
19.3. A Figure Reference Must Be Substantive	170
19.4. CAPTION — Figure Title and Caption	171
19.5. Requirements for CAPTION.....	171
19.6. POST_FIGURE_EXPLANATION — Explanation After the Figure.....	171
19.7. The Explanation Is Limited to What Is Actually Shown	172
19.8. FIGURE_PROMPT — Precise Specification for Creating a Figure	172
19.9. All Four Components Must Use the Same Language.....	172
19.10. Consistency of Numerical Values.....	173
19.11. Semantic Consistency	173
19.12. Figure Package as a Single Artifact	174
19.13. Life Cycle of the Figure Text Package.....	174
20. Reproducibility and the GitHub Package.....	175
20.1. From an Article to a Reproducible Research Package.....	175
20.2. Prompt-Native Reproducibility.....	175
20.3. Basic Repository Structure.....	176
20.4. README.md — Research Map	177
20.5. MODEL.md — Formal Model.....	177

20.6. Model Versioning	178
20.7. REPRODUCIBILITY.md.....	178
20.8. The prompts/ Directory	179
20.9. Not Every Utterance Needs to Be Preserved	179
20.10. The experiments/ Directory	179
20.11. README.md for an Individual Experiment	180
20.12. Frozen Specification	180
20.13. results.md.....	181
20.14. Experiment Status	182
20.15. The figures/ Directory.....	182
20.16. The references/ Directory	183
20.17. Preserving Working Scholar Queries	183
20.18. The real_case/ Directory	184
20.19. NO_CODE Means Executable Code Is Not Required.....	184
20.20. A Mathematical Specification as a Reproducibility Artifact.....	184
20.21. An Algorithm in Natural Language Can Also Be an Artifact	185
20.22. Provisional and Verified Values Must Be Kept Separate	185
20.23. Negative Experiments Must Also Be Preserved	185
20.24. Preserving the Seed	186
20.25. Preserving Thresholds	186
20.26. Preserving Figure Specifications.....	186
20.27. Preserving Scholar Queries	187
20.28. Restrictions on Third-Party Documents.....	187
20.29. The Repository as a Representation of Research State	187
20.30. The Repository Must Not Change the Science.....	188
20.31. Minimum HAAG GitHub Package.....	188
20.32. The Complete HAAG Reproducibility Principle.....	189
20.33. Reproducibility of the Trajectory, Not Only the Result.....	190
21. Separation of Research, Writing, and Artifact Generation	190
21.1. Why HAAG Separates These Processes	190
21.2. RESEARCH LAYER — The Research Layer.....	191
21.3. Hypotheses Belong to Research, Not to the Text.....	191
21.4. The Model Also Belongs to the Research Layer	191

21.5. Evidence — The Evidence Base	192
21.6. Experiment and Falsification	192
21.7. WRITING LAYER — The Scientific Writing Layer.....	192
21.8. Direction of Dependence Between Research and Text	193
21.9. IMRaD as a Projection of the Research State	193
21.10. REWRITE After a Change in Research State	193
21.11. ARTIFACT LAYER — The Research Artifact Layer	194
21.12. A Figure Does Not Determine Interpretation.....	194
21.13. A Table Must Not Create a New Result Either	194
21.14. The Repository Must Not Dictate the Model	195
21.15. Interaction Among the Three Layers	195
21.16. General Rule for Separating the Layers	195
22. Final Audits	195
22.1. Why Separate Final Control Is Necessary	195
22.2. A. SCIENTIFIC AUDIT — Scientific Audit	196
22.3. Hypothesis Verification	196
22.4. Model Consistency	196
22.5. Alternative Explanations.....	197
22.6. Negative Results.....	197
22.7. Limitations.....	197
22.8. B. NUMERICAL AUDIT — Numerical Audit.....	197
22.9. Headline Values.....	197
22.10. Consistency of Equations and Metrics.....	198
22.11. PROVISIONAL Must Not Silently Become FINAL	198
22.12. C. CLAIMS AUDIT — Audit of Claims.....	198
22.13. Claim Strength Must Match the Evidence	199
22.14. D. IMRaD AUDIT	199
22.15. E. SOURCE/CITE AUDIT.....	199
22.16. Working Scholar Queries Must Be Resolved.....	200
22.17. F. FIGURE AUDIT.....	200
22.18. G. REPOSITORY AUDIT.....	200
22.19. Final Audit Matrix	201
23. Improved Minimal Executable HAAG Program	201

23.1. Why an Executable Program Is Needed	201
23.2. START — Initiating the Research.....	201
23.3. Initial Exploration of the Problem Space	201
23.4. Loop Over Candidate Problems.....	202
23.5. When Several Fundamentally Different Directions Survive.....	202
23.6. HUMAN Can Do More Than Select	202
23.7. HYPOTHESIS — Formulating a Hypothesis	203
23.8. If the Hypothesis Fails the Initial Test.....	203
23.9. Designing the Experiment	203
23.10. HUMAN Before the Experiment — Only When Necessary	204
23.11. FREEZE → RUN → AUDIT.....	204
23.12. If the Result Is Negative or Null	204
23.13. Real-World Case	205
23.14. Real Data Must Not Be Silently Replaced with Synthetic Data	205
23.15. Reviewer Before Article Writing.....	205
23.16. If a Critical Problem Remains	205
23.17. WRITE Begins After Research Readiness.....	206
23.18. Figure Creation Loop.....	206
23.19. REFERENCE_ENRICHMENT.....	206
23.20. PACKAGE — Preparing Supplementary Materials.....	207
23.21. Final Council of Agents	207
23.22. Comprehensive VERIFY	207
23.23. The Final Decision Belongs to HUMAN	208
23.24. If Final Synthesis Is Authorized	208
23.25. If the Researcher Does Not Authorize Completion	208
23.26. Overall Logic of the Executable Program.....	209
24. Rules for Human Interaction.....	209
24.1. HAAG's Internal Language Must Not Become the Language of Dialogue	209
24.2. The Formal Structure Remains Internal.....	210
24.3. What Should Be Presented at a HUMAN Decision Point.....	210
24.4. Current Scientific State	211
24.5. Why the Decision Matters.....	211
24.6. Alternatives Must Be Genuinely Different.....	211

24.7. Expected Scientific Consequences	212
24.8. One Clear Question	212
24.9. FREE_TEXT — The Human Is Not Restricted to the Proposed Options	212
24.10. Do Not Ask the Human About Inconsequential Matters.....	212
24.11. The Principle of Adaptive Autonomy	213
24.12. HUMAN as an Interface for Scientific Governance.....	213
24.13. General Model of Human–HAAG Interaction.....	213
24.14. The Human Should Not Manage Microsteps but Retains Scientific Control	214
24.15. Concluding Logic of Chapters 21–24.....	214
25. Final Output Package of HAAG v1.1	215
25.1. The Publication as One Result, Not the Only Result, of Research	215
25.2. Contents of the Final Output Package.....	215
25.3. FINAL_PUBLICATION — The Final Scientific Publication.....	216
25.4. FIGURE_LIST — Figure Register	216
25.5. FIGURE_PROMPTS — Prompts Used to Create Figures	217
25.6. FIGURE_CAPTIONS — Separate Caption Register	217
25.7. PRE_POST_FIGURE_TEXT — Text Surrounding Figures	218
25.8. RESEARCH_GRAPH — Graph of the Research Trajectory	218
25.9. HYPOTHESIS_TREE — Hypothesis Tree	219
25.10. AGENT_HISTORY — History of Agent Activities.....	219
25.11. HUMAN_DECISION_TRACE — Trace of Researcher Decisions.....	219
25.12. REJECTED_BRANCHES — Rejected Branches.....	220
25.13. FAILED_EXPERIMENTS — Failed Experiments	220
25.14. EXPERIMENT_REGISTRY — Register of All Experiments	221
25.15. CLAIM_PROVENANCE — Provenance of Claims.....	221
25.16. SOURCE_PROVENANCE — Provenance of Sources.....	222
25.17. WORKING_SCHOLAR_QUERIES — History of Bibliographic Searching	222
25.18. NUMERICAL_AUDIT — Final Numerical Audit.....	222
25.19. REPRODUCIBILITY_PACKAGE	223
25.20. UNRESOLVED_PROBLEMS — Unresolved Problems.....	223
25.21. FUTURE_RESEARCH_BRANCHES — Future Research Branches	223
25.22. Levels of Openness of the Final Output Package.....	224
25.23. Final Structure of the Outcome.....	224

26. Fundamental HAAG Rule	225
26.1. The System's Primary Question	225
26.2. From Text Generation to Research Transition	225
26.3. A Scientifically Justified Transition.....	225
26.4. THINK — Formulate a Scientific Possibility.....	226
26.5. CHALLENGE — Subject the Idea to Criticism.....	226
26.6. VERIFY — Perform Verification.....	226
26.7. DECIDE — Make a Decision About the Research Trajectory	227
26.8. PREREGISTER — Specify the Experiment Before Obtaining the Result	227
26.9. RUN — Execute the Experiment	227
26.10. AUDIT — Examine the Obtained Result	227
26.11. TRANSITION — Update the Research State.....	228
26.12. WRITE — Only After the Transition.....	228
26.13. FIGURE — Visualize the Verified State.....	228
26.14. VERIFY SOURCES — Check the Scientific Basis	229
26.15. CITE — Only After Verification.....	229
26.16. PACKAGE — Assemble the Results	229
26.17. FINAL AUDIT.....	229
26.18. The Complete HAAG v1.1 Cycle.....	230
26.19. Publication Emerges from the Trajectory	230
27.1. HAAG as a Prompt-Native Domain-Specific Language.....	230
27.2. Natural Language Defines Semantics	231
27.3. Logical Primitives Provide Control.....	231
27.4. LLM Agents Provide Distributed Reasoning.....	232
27.5. External Scholarly Systems Provide Grounding	232
27.6. Numerical Tools Provide Verifiable Calculations.....	232
27.7. Image Generation Systems Create Controlled Scientific Artifacts.....	232
27.8. The Human Defines Scientific Intent	233
27.9. Functional Distribution of HAAG Components.....	233
27.10. HAAG Is Not Merely a Prompt.....	233
27.11. A Compact HAAG Model.....	233
27.12. From a Topic to a Research Program.....	234
27.13. From a Research Program to a Human-Agent Trajectory.....	234

27.14. From the Trajectory to Verified Results	234
27.15. From Verified Results to Publication	235
27.16. Conceptual Formula of HAAG	235
27.17. Attribution of the Concept	235
27.18. Version Identifier	236
27.19. Concluding Definition of HAAG v1.1	236
Content	236

