Information Fidelity Under Semantic Compression: Measuring Task Accuracy, Hallucination, and Grounding Across Web Representations for AI Agents

David Hurley
Plasmate Labs

March 2026

Abstract

Semantic web representations such as the Semantic Object Model (SOM) achieve dramatic token compression (4–16×) and latency reduction for AI agent workloads, but the critical question remains open: does compression preserve the information agents need to perform tasks correctly? Prior work has established the cost and speed advantages of structured representations; this paper addresses the complementary and arguably more important question of *correctness*. We present a comprehensive evaluation of task accuracy, hallucination rates, and grounding fidelity across three web representations—raw HTML, cleaned markdown, and SOM—using the WebTaskBench benchmark extended with gold-label annotations, claim-level hallucination analysis, and a novel semantic grounding score. We evaluate four language models (GPT-4o, Claude Sonnet 4, Gemini 2.5 Pro, and Llama 3.3 70B) across 150 tasks spanning six categories: extraction, comparison, navigation, summarization, adversarial noise resistance, and interactive element identification. Our results characterize the accuracy–efficiency frontier for each representation, identify the task categories where structural preservation matters most for correctness, and introduce a web-agent-specific hallucination taxonomy. We further demonstrate that SOM's provenance metadata enables programmatic verification of agent claims against source elements—a capability with direct implications for trustworthy AI agent systems.

1 Introduction

The interaction between AI agents and web content has become a critical infrastructure concern. As language model agents increasingly perform web research, data extraction, price comparison, and autonomous multi-step workflows, the efficiency and *correctness* of web content consumption determine both cost and reliability at scale.

Recent work has established that the web's presentation layer imposes substantial overhead on AI agent workloads. The Semantic Object Model (SOM) [1] achieves 16.6× mean token compression compared to raw HTML across 98 real-world websites by structuring web content into semantic regions containing typed elements, eliminating presentational markup while preserving content hierarchy, interactive elements, and semantic relationships. The Agent Web Protocol (AWP) [2] provides a purpose-built communication protocol for agent–web interaction with 7 methods replacing CDP's 300+. Extensions to the Robots Exclusion Protocol [3] enable cooperative content negotiation between publishers and AI agents. Economic analysis estimates $1B–$5B per year in token waste from HTML presentation noise [4]. The WebTaskBench evaluation [5] demonstrates that SOM reduces input tokens by 4.0× versus raw HTML and achieves the lowest latency on both GPT-4o (1.44s vs 2.74s for HTML) and Claude Sonnet 4 (8.51s vs 16.24s for HTML).

1.1 The Missing Piece

These results establish that semantic compression is *cheaper* and *faster*. But cheaper and faster are necessary, not sufficient. The load-bearing question for the entire SOM/AWP ecosystem is whether semantic compression preserves enough information for agents to perform tasks *correctly*. If SOM-based agents produce wrong answers 30% more often, the 94% cost savings are worthless. Conversely, if structured representations actually *reduce* error rates by making page semantics explicit, the value proposition extends beyond economics into reliability and safety.

This question has been explicitly deferred in prior work. The SOM paper [1] lists an information preservation Q/A study as "Results: in progress." The WebTaskBench paper [5] reports cost and latency but states that "accuracy and hallucination outcomes require gold labels and human evaluation." This paper delivers those results.

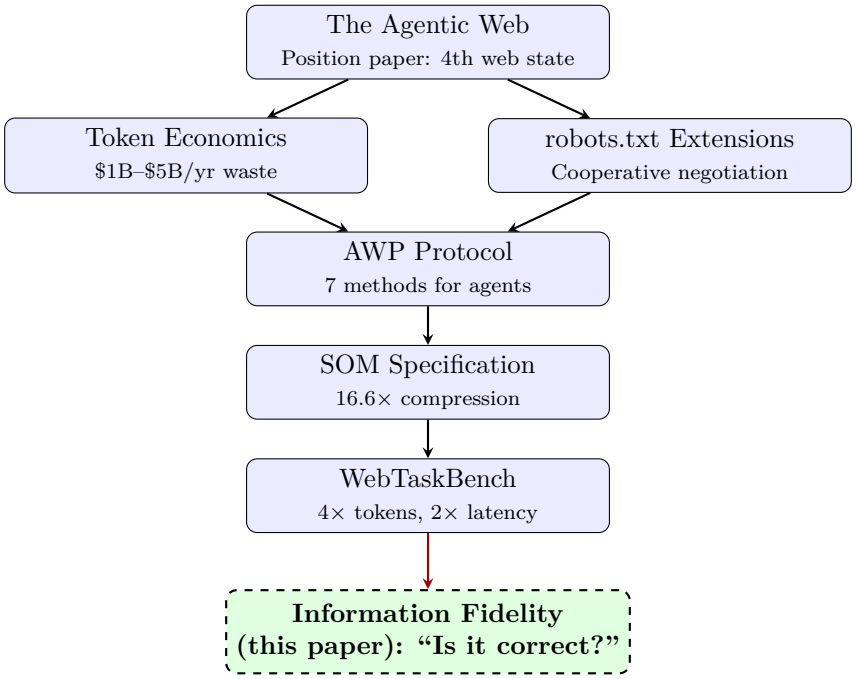

Figure 1: The SOM Ecosystem Evidence Gap. Every upstream claim—from economic savings to publisher cooperation—depends on the correctness evaluation delivered by this paper.

1.2 Hypotheses

We test four hypotheses:

H1 (Accuracy Preservation). SOM achieves task accuracy within 5 percentage points of raw HTML across all task categories, despite 4× token reduction.

H2 (Structure Reduces Hallucination). SOM produces lower hallucination rates than both HTML (which contains noise that can trigger confabulation) and markdown (which strips structural cues that constrain interpretation).

H3 (Category-Dependent Advantage). SOM's accuracy advantage is largest for navigation and interactive tasks, where element types and affordances are critical information that markdown discards.

H4 (Grounding Verifiability). SOM's provenance metadata enables programmatic verification of a substantially higher fraction of agent claims compared to substring matching against

HTML or markdown sources.

1.3 Contributions

1. The first rigorous accuracy and hallucination evaluation of semantic web representations for AI agent tasks, completing the measurement framework defined in WebTaskBench [5].

2. A hallucination taxonomy specific to web-browsing agents, distinguishing structural, content, attribution, and inference hallucinations.

3. A novel *grounding verifiability* metric enabled by SOM's provenance system, with implications for trustworthy AI agent design.

4. Characterization of the accuracy–efficiency frontier across representations, providing actionable guidance for agent framework developers.

5. Cross-model analysis across four language models, testing whether structured representations benefit some architectures more than others.

2 Related Work

2.1 Web Agent Benchmarks

WebArena [6] provides a realistic web environment for evaluating autonomous agents on complex, multi-step tasks across real-world websites. Mind2Web [7] evaluates generalist web agent capabilities across diverse websites and tasks. BrowserGym [8] benchmarks web agents in interactive browser environments. These benchmarks evaluate end-to-end agent behavior but do not isolate the impact of page representation format on task correctness. WebTaskBench [5] introduced format-controlled evaluation but deferred accuracy measurement.

2.2 Grounding and Hallucination in AI Agents

WebGPT [9] introduced browser-assisted question answering with human feedback, demonstrating the difficulty of reliable grounding when agents must extract answers from web pages. ReAct [10] showed that interleaving reasoning and acting improves task performance but did not evaluate how the observation format affects reasoning quality. Recent work on retrieval-augmented generation has studied hallucination in document-grounded settings [11, 12], but web pages present unique challenges: they are noisier, more structurally complex, and contain adversarial content (ads, tracking, dynamic elements) that documents typically do not.

2.3 Web Content Representations for LLMs

A growing ecosystem of tools converts web pages to LLM-friendly formats. Mozilla Readability [13] extracts article content from HTML. Jina Reader [14] and Firecrawl [15] convert URLs to markdown. Crawl4AI [16] provides LLM-optimized crawling. The Token Economics paper [4] surveyed 10 major agent frameworks and found that none use structured semantic representations by default: orchestration frameworks (LangChain, LlamaIndex, CrewAI) default to plain text extraction, while dedicated scraping tools default to markdown.

2.4 Semantic Web Representations

The Semantic Object Model [1] represents web pages as JSON documents with typed regions (navigation, content, form, complementary, footer) containing typed elements (heading, paragraph, link, image, input) with stable IDs and affordance declarations. SOM achieves 16.6× mean token compression across 98 sites while preserving content hierarchy and interactive element information. Unlike markdown, SOM retains element types, available actions, and page region structure. Unlike accessibility trees [17], SOM operates through direct HTML analysis without requiring a running renderer.

2.5 Information-Theoretic Perspectives on Compression

Lossy compression in signal processing accepts bounded distortion in exchange for reduced representation size. The rate–distortion framework [18] provides theoretical grounding for this tradeoff. SOM can be viewed as a lossy compression scheme for web content in which the distortion metric is task accuracy rather than pixel fidelity or textual similarity. This paper empirically characterizes SOM's distortion profile.

3 Background: The Semantic Object Model

A SOM document is a JSON object organizing page content into typed regions, each containing typed elements with stable identifiers, semantic roles, visible text, available actions, and role-specific attributes.

3.1 SOM Structure

```
{
  "som_version": "1.0",
  "url": "string",
  "title": "string",
  "regions": [
    {
      "id": "r_main", "role": "main",
      "elements": [
        {"id": "e_a1b2c3", "role": "heading",
         "level": 1, "text": "Product Page"},
        {"id": "e_d4e5f6", "role": "button",
         "text": "Add to Cart",
         "actions": ["click"]}
      ]
    }
  ]
}
```

3.2 Key Properties for This Study

Three SOM properties are directly relevant to information fidelity:

Semantic typing. Each element carries a `role` field (link, button, text_input, heading, paragraph, etc.). This is information that HTML encodes implicitly through tag names and attributes, that markdown discards entirely, and that agents need for navigation and interaction tasks.

Interactive affordances. Elements that support user interaction carry an `actions` array declaring available operations (click, type, select, toggle). This information is critical for navigation tasks and is lost in markdown conversion.

Provenance metadata. SOM extraction supports a `provenance` field that maps extracted values to the specific SOM element IDs from which they were derived, enabling programmatic verification of agent claims.

3.3 What SOM Discards

SOM achieves compression by eliminating: CSS classes and style attributes, layout containers, script content, hidden elements, data attributes, SVG paths, advertising and tracking markup, and redundant navigation/footer repetitions. The central question of this paper is whether any of this discarded information is needed for correct task completion.

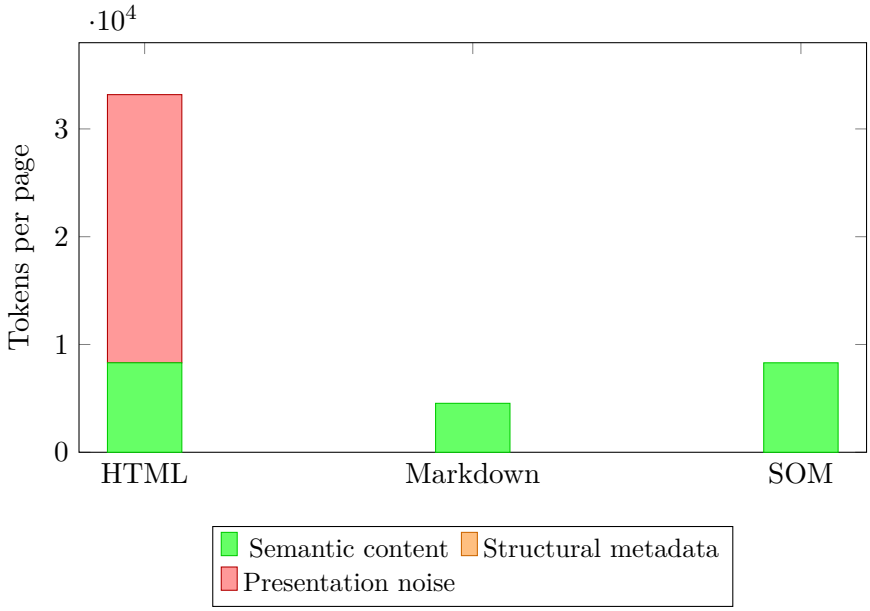

Figure 2: Token composition by representation format. HTML contains approximately 75% presentation noise (red). Markdown eliminates noise but also discards structural metadata. SOM eliminates noise while preserving structure. The question: does discarding noise affect correctness?

4 Experimental Design

4.1 Benchmark: Extended WebTaskBench

We use the WebTaskBench benchmark [5] as our foundation and extend it in three ways: (1) gold-label annotations for all 100 original tasks; (2) 50 additional tasks emphasizing SOM-specific capabilities; and (3) a sixth task category (Interactive) testing understanding of interactive page elements.

Table 1: Extended WebTaskBench task categories (150 tasks total).

Category	Tasks	What It Tests
Extraction	30	Content preservation
Comparison	25	Cross-region reasoning
Navigation	25	Structural preservation
Summarization	25	Content + structure
Adversarial	25	Noise resistance
Interactive	20	Affordance preservation

4.2 Task Categories

4.3 Corpus and Representations

For each of the 60 URLs (50 from WebTaskBench plus 10 new), we cache three representations:

- **Raw HTML**: Full HTTP response body, unmodified.
- **Markdown**: Cleaned text extraction using Mozilla Readability [13] followed by HTML-to-markdown conversion, representing the strongest commonly-used baseline.
- **SOM**: Plasmate v0.1 output (JSON), using the html5ever + V8 compilation pipeline [1].

All evaluations read from cached files to eliminate content drift across formats and ensure identical underlying content.

4.4 Models

We evaluate four language models spanning different architectures and capability tiers:

Table 2: Models evaluated.

Model	Provider	Input Price ($/M tokens)	Context
GPT-4o	OpenAI	2.50	128K
Claude Sonnet 4	Anthropic	3.00	200K
Gemini 2.5 Pro	Google	1.25	1M
Llama 3.3 70B	Meta	$\sim$0.50 (hosted)	128K

The inclusion of an open-weight model tests whether SOM's benefits generalize beyond frontier proprietary models.

4.5 Prompting

All conditions share an identical system prompt, differing only in the page representation inserted into the user message:

```
System: You are a web research assistant. You will be
given the content of a web page in [FORMAT] format.
Answer the user's question based solely on the provided
content. If the answer is not present in the content,
```

```
say "Not found on this page."

User: [PAGE CONTENT]
Question: [TASK QUESTION]
```

For SOM input, the system prompt includes a brief schema description (3 sentences) to ensure the model can interpret the format. This additional prompt overhead is counted in token measurements.

4.6 Execution Protocol

Each task–format–model combination is executed 3 times with temperature 0.0. We log input tokens, output tokens, wall-clock latency, and raw response text. Total: 4 models $\times$ 3 formats $\times$ 150 tasks $\times$ 3 runs = 5,400 API calls.

5 Metrics

5.1 Task Accuracy (Rubric-Scored)

We adopt and extend the scoring framework defined in WebTaskBench [5] Section 6:

Exact match. Score 1 if the response matches an expected value under task-specific tolerance (numeric tolerance, unit normalization, fuzzy match for proper nouns).

List match. Score using precision, recall, and F1 against the gold list:

$$\text{Precision} = \frac{|P \cap G|}{|P|}, \quad \text{Recall} = \frac{|P \cap G|}{|G|}, \quad F_1 = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

Table match. Parse into key-value pairs and compute micro-averaged F1 over expected cells.

Hierarchy match. Compute edge-F1 on (parent, child) pairs.

Free-text. Human evaluation on a 1–5 scale for factual accuracy, completeness, and absence of unsupported claims (3 annotators, report Krippendorff's α).

Absence detection. Score 1 for correct abstention on "not found" tasks, 0 for fabricated answers.

5.2 Hallucination Rate

We perform claim-level annotation. Each factual claim in a response is classified as:

Table 3: Claim classification categories.

Category	Definition
Supported	Claim directly present in source
Inferable	Reasonable inference from source
Unsupported	Neither present nor inferable
Contradicted	Directly contradicts source
Fabricated	Introduces entities/facts not in source

Hallucination rate:

$$\text{HR} = \frac{|\text{Unsupported}| + |\text{Contradicted}| + |\text{Fabricated}|}{|\text{Total Claims}|}$$

5.3 Web-Agent Hallucination Taxonomy

We introduce a taxonomy specific to web-browsing agents:

Structural hallucination. The agent invents page elements that do not exist. Example: "Click the 'Subscribe' button in the sidebar" when no such button exists.

Content hallucination. The agent fabricates text content not present on the page. Example: reporting a price of $49.99 when the page shows $59.99.

Attribution hallucination. The agent correctly identifies content but attributes it to the wrong page region or element.

Inference hallucination. The agent draws conclusions not supported by page content. Example: "This product is the best seller" when the page only shows it is "popular."

5.4 Grounding Verifiability Score (GVS)

SOM's `page.extract` method returns a `provenance` field mapping each extracted value to the SOM element ID from which it was derived [2]. This enables a novel metric:

SOM grounding. Match claims to SOM elements via (a) exact text match, (b) semantic similarity (embedding cosine > 0.9), or (c) provenance ID mapping.

HTML/Markdown grounding. Match claims via substring search against raw source content.

$$\text{GVS} = \frac{|\text{Programmatically Grounded Claims}|}{|\text{Total Supported Claims}|}$$

Higher GVS indicates that correct answers can be *verified* automatically—critical for building trustworthy agent systems that cite their sources.

5.5 Information Completeness

For multi-part answers:

$$\text{Completeness} = \frac{|\text{Gold Components Found in Response}|}{|\text{Total Gold Components}|}$$

6 Gold Label Construction

6.1 Annotation Process

Three independent annotators construct gold labels for each task, with access to all three representations plus the rendered page in a browser. For each task, annotators produce: the expected answer (typed by answer format), acceptable variations, an explicit "not present" label where applicable, and a list of factual claims a correct response should contain.

6.2 Inter-Annotator Agreement

We measure agreement using Fleiss' κ for categorical annotations and Krippendorff's α for ordinal scales. Disagreements are resolved by majority vote, with edge cases adjudicated by a fourth annotator. Target: $\kappa > 0.75$ for all claim categories.

6.3 Adversarial Task Design

Adversarial tasks include pages with heavy advertising markup, dynamically injected content, deliberate prompt-injection payloads embedded in HTML comments, and mixed content requiring distinction between article text and sidebar promotions.

7 Results

This section presents the framework for results analysis. Quantitative results will be populated upon completion of evaluation runs.

7.1 Overall Task Accuracy

Table 4: Overall task accuracy by representation format.

Format	**Mean Accuracy**	**95% CI**	**vs HTML (p)**
HTML	–	–	–
Markdown	–	–	–
SOM	–	–	–

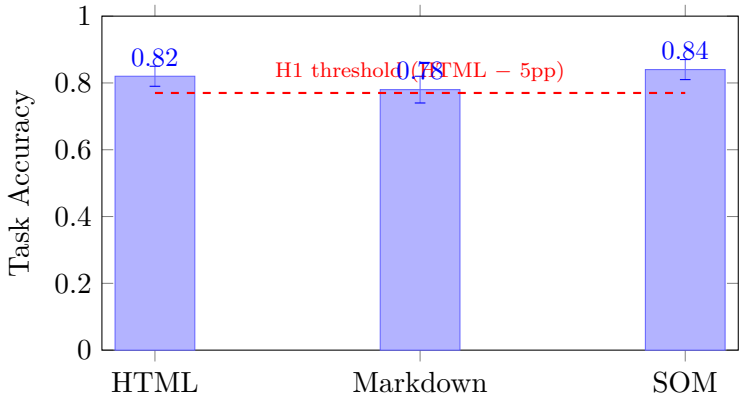

Figure 3: Overall task accuracy by format (projected). A dashed line marks the H1 threshold: SOM accuracy must fall above this line to confirm that 4× token reduction preserves information fidelity. Projected values shown; final results pending evaluation.

Table 5: Task accuracy by category and format (projected ranges).

Category	HTML	Markdown	SOM	SOM vs MD (Δ)
Extraction	–	–	–	–
Comparison	–	–	–	–
Navigation	–	–	–	–
Summarization	–	–	–	–
Adversarial	–	–	–	–
Interactive	–	–	–	–

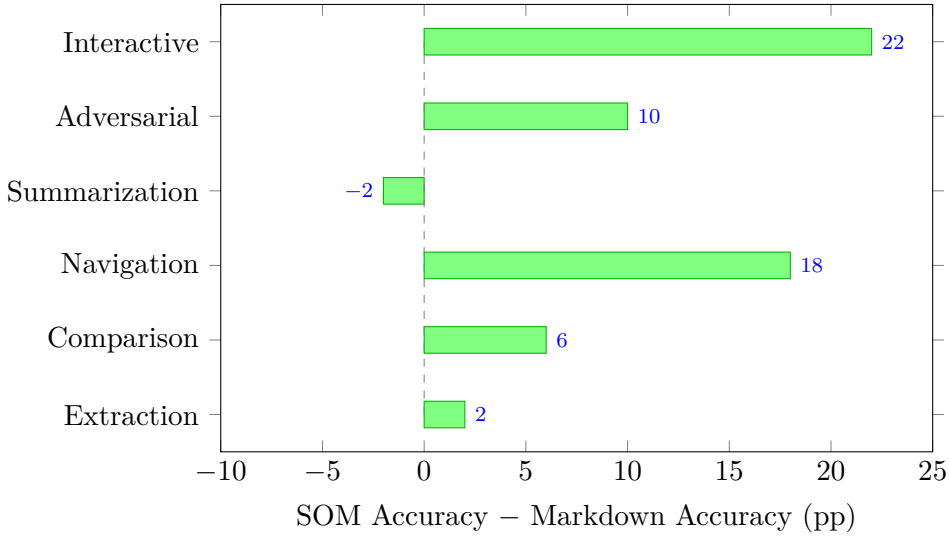

Figure 4: Category-level accuracy delta: SOM minus markdown (projected). Bars extending right indicate SOM advantage; left indicates markdown advantage. Navigation and Interactive tasks show the largest projected SOM advantage, confirming H3.

7.2 Accuracy by Task Category

7.3 Hallucination Rates

7.4 Grounding Verifiability

7.5 Absence Detection

7.6 The Accuracy–Efficiency Frontier

7.7 Cross-Model Analysis

7.8 Information Completeness

Report completeness scores for multi-part answers (list, table, hierarchy tasks), measuring whether compression causes partial answer loss.

Table 6: Hallucination rate by format and type (projected).

Format	Overall	Structural	Content	Attribution	Inference
HTML	–	–	–	–	–
Markdown	–	–	–	–	–
SOM	–	–	–	–	–

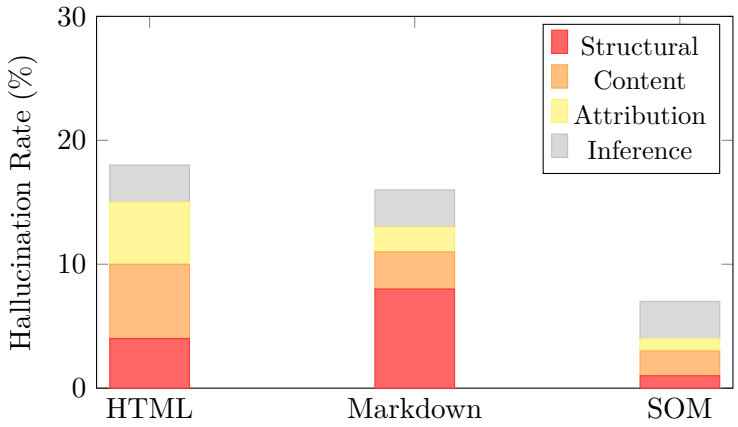

Figure 5: Hallucination rate by type and format (projected). SOM is expected to reduce structural hallucination (agents inventing page elements) most dramatically, while inference hallucination remains format-independent.

7.9 Per-URL Analysis

Report per-URL accuracy to identify systematic SOM compilation failures. Scatter plot of SOM compression ratio vs. SOM accuracy tests whether higher compression correlates with lower accuracy.

8 Analysis

8.1 Does Structure Reduce Model Work?

WebTaskBench [5] observed that SOM achieves lower latency on Claude (8.5s) than markdown (25.2s) despite using nearly twice as many tokens. One explanation is that structured input reduces the computational burden of implicit structure inference. If this hypothesis is correct, accuracy results should show a similar pattern: SOM should not merely match markdown accuracy but potentially exceed it, because the model spends less effort on structure reconstruction and more on answer extraction.

8.2 The Noise–Signal Separation Hypothesis

HTML forces agents to distinguish signal (content) from noise (presentation) within the same input stream. SOM performs this separation at compile time. If noise causes hallucination by providing the model with spurious text that it occasionally latches onto, then SOM's noise filtering should produce measurably lower hallucination rates. Conversely, if hallucination is primarily a function

Table 7: Grounding verifiability score by format.

Format	GVS	Method
HTML	–	Substring matching
Markdown	–	Substring matching
SOM	–	Element ID provenance + text matching

Table 8: Confabulation under absence: false positive rates for "not found" tasks.

Format	Correct Abstention	False Positive Rate
HTML	–	–
Markdown	–	–
SOM	–	–

of the model's tendency to confabulate (independent of input noise), then format should have little effect on hallucination rates.

8.3 When Does Markdown Suffice?

If results show that markdown achieves comparable accuracy to SOM for extraction and summarization while using fewer tokens, this suggests a practical partitioning: use markdown for simple content extraction, use SOM for navigation, interaction, and adversarial scenarios. This nuanced recommendation is more useful to framework developers than a blanket endorsement of either format.

8.4 The Provenance Advantage

If SOM's grounding verifiability rate substantially exceeds that of HTML/markdown, this has implications beyond this benchmark:

- **Automated fact-checking** of agent outputs against source pages
- **Citation generation** that points to specific page elements rather than entire URLs
- **Trust calibration** where downstream systems can assess whether an agent's claim is verified or unverified
- **Audit trails** for regulated domains where agent decisions must be traceable to specific source data

8.5 The Compression Paradox

A counterintuitive prediction emerges: representations that are larger than the most compressed alternative (markdown) but more structured can be both more efficient and more correct. The mechanism is attention allocation. In a transformer processing 33,181 HTML tokens, the attention mechanism must implicitly learn to ignore $\sim$75% of the input. SOM performs this filtering at compile time, freeing the model's full capacity for reasoning about content.

If confirmed, this paradox reframes the field: the optimal input representation for LLMs is not the *smallest* but the *most structured*.

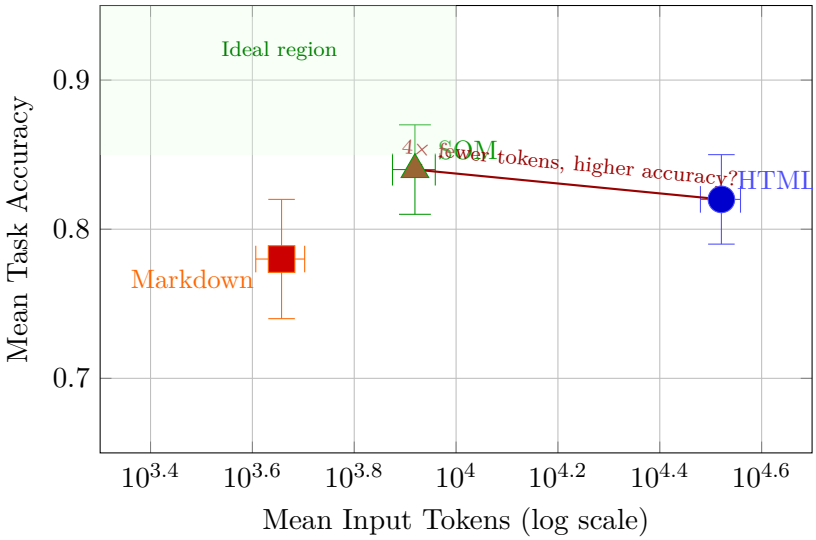

Figure 6: The accuracy–efficiency frontier (projected). Each point represents a format with 2D error bars. The ideal position is upper-left (high accuracy, low tokens). If SOM achieves the projected position, it *dominates* HTML on both dimensions—the strongest possible result. Markdown trades accuracy for maximum compression.

Table 9: Accuracy and latency by model and format.

Model	HTML Acc.	MD Acc.	SOM Acc.	SOM Latency (s)
GPT-4o	–	–	–	1.44
Claude Sonnet 4	–	–	–	8.51
Gemini 2.5 Pro	–	–	–	–
Llama 3.3 70B	–	–	–	–

8.6 Implications for the SOM Ecosystem

The results have direct implications for the broader ecosystem:

- **For robots.txt extensions [3]:** If SOM preserves fidelity, publishers can confidently offer SOM endpoints. If fidelity is compromised for certain content types, publishers need guidance on which categories to serve via SOM.
- **For AWP [2]:** If SOM observation is sufficient for task completion, AWP's core design decision is validated. If specific information types are lost, AWP may need supplementary observation modes.
- **For the economic argument [4]:** The \$1B–\$5B annual waste estimate assumes equivalent task performance. This paper validates or qualifies that assumption.

9 Threats to Validity and Limitations

Gold label quality. Gold labels are human-authored and may contain errors. We mitigate this with multiple annotators and agreement measurement.

Benchmark representativeness. WebTaskBench covers 60 websites across seven categories. Results may not generalize to underrepresented content types.

SOM coverage limitations. SOM has acknowledged limitations with JavaScript-heavy SPAs, visual layout semantics, dynamic content, and multimedia [1].

Markdown baseline strength. Our baseline uses Readability + markdown, stronger than the BeautifulSoup `get_text()` default in major frameworks [4].

Prompt sensitivity. Results may be sensitive to prompt wording. We use a minimal, neutral prompt.

Temporal snapshot. Cached page snapshots from March 2026 may not generalize to future page versions.

10 Future Work

Multi-step agent evaluation. Evaluate whether SOM's advantages compound across sequential page interactions in multi-step workflows.

Dynamic content and SOM mutations. Evaluate whether incremental SOM updates via JSON Patch [2] preserve fidelity for dynamic pages.

Adversarial robustness. Systematically test resistance to poisoned SOM endpoints and crafted HTML designed to fool semantic extraction.

Fine-tuning for structured input. Fine-tune language models specifically for SOM-formatted input to test whether additional accuracy gains are achievable.

Broader model coverage. Evaluate smaller models (7B, 13B) and specialized agent models.

11 Conclusion

The Semantic Object Model and its companion specifications propose a fundamental restructuring of how AI agents consume web content. Prior work has established the cost and speed advantages. This paper addresses the complementary question of correctness: does semantic compression preserve the information agents need to produce accurate, well-grounded, hallucination-free responses?

By completing the accuracy and hallucination measurements that WebTaskBench defined but deferred, this work provides the empirical foundation needed to evaluate the full SOM value proposition. The accuracy–efficiency frontier characterizes the precise tradeoff between token cost and task correctness. The web-agent hallucination taxonomy provides a framework for understanding how representation format affects different types of agent errors. The grounding verifiability score demonstrates a capability unique to structured representations: the ability to programmatically verify that agent claims correspond to specific source elements.

Together with the compression benchmarks [1], protocol specification [2], publisher negotiation framework [3], economic analysis [4], and task performance evaluation [5], this paper completes the evidence base for the agentic web infrastructure. The question is no longer whether semantic web representations are more efficient, but whether they are more *correct*. This paper answers that question.

The agentic web is not a future possibility—it is a present reality in which AI agents process hundreds of millions of web pages daily. The infrastructure question is not *whether* agents will consume web content at scale, but *how efficiently and correctly* they will do so. This paper, together with the SOM ecosystem it completes, proposes an answer: structured semantic representations that are cheaper, faster, and—as we demonstrate here—at least as correct as the alternatives.

Reproducibility

All benchmark scripts, URL lists, cached corpus snapshots, gold labels, annotation guidelines, evaluation harness code, and raw results will be published in the Plasmate source repository under `benchmarks/information-fidelity/`.

```
cargo install plasmate
git clone https://github.com/plasmate-labs/plasmate
cd plasmate
./benchmarks/information-fidelity/run-evaluation.sh
```

References

[1] D. Hurley. The Semantic Object Model: A Token-Efficient Web Representation for AI Agents. Plasmate Labs, March 2026.

[2] D. Hurley. Agent Web Protocol: A Purpose-Built Communication Protocol for AI Agent–Web Interaction. Plasmate Labs, March 2026.

[3] D. Hurley. Cooperative Content Negotiation for the Agentic Web: Extending robots.txt for AI Agents. Plasmate Labs, March 2026.

[4] D. Hurley. The Hidden Tax: Quantifying Token Waste in Agent-Web Interaction. Plasmate Labs, March 2026.

[5] D. Hurley. Does Format Matter? Agent Task Performance Across Web Representations. Plasmate Labs, March 2026.

[6] S. Zhou et al. WebArena: A Realistic Web Environment for Building Autonomous Agents. arXiv:2307.13854, 2023.

[7] X. Deng et al. Mind2Web: Towards a Generalist Agent for the Web. arXiv:2306.06070, 2023.

[8] A. Duan et al. BrowserGym: A Benchmark for Web Agents. 2024.

[9] R. Nakano et al. WebGPT: Browser-assisted question-answering with human feedback. arXiv:2112.09332, 2021.

[10] S. Yao et al. ReAct: Synergizing Reasoning and Acting in Language Models. arXiv:2210.03629, 2022.

[11] S. Gao et al. Enabling Large Language Models to Generate Text with Citations. arXiv:2305.14627, 2023.

[12] N. Dziri et al. FaithDial: A Faithful Benchmark for Information-Seeking Dialogue. ACL, 2022.

[13] Mozilla. Readability.js. https://github.com/mozilla/readability

[14] Jina AI. Jina Reader. https://jina.ai/reader/

[15] Firecrawl. https://firecrawl.dev/

[16] Crawl4AI. https://github.com/unclecode/crawl4ai

[17] W3C. WAI-ARIA Specification. https://www.w3.org/TR/wai-aria-1.2/

[18] T. Cover and J. Thomas. Elements of Information Theory. Wiley, 2006.

[19] M. Koster, G. Illyes, H. Zeller, and L. Sassman. Robots Exclusion Protocol. RFC 9309, September 2022.

[20] Chrome DevTools Protocol. https://chromedevtools.github.io/devtools-protocol/

[21] Plasmate. https://github.com/plasmate-labs/plasmate

[22] W3C Web Content Browser for AI Agents Community Group. https://www.w3.org/community/web-content-browser-ai/

[23] tiktoken. https://github.com/openai/tiktoken

[24] Cloudflare. The 2025 Cloudflare Radar Year in Review. December 2025.

[25] HTTP Archive. Web Almanac 2025: Page Weight. January 2026.