Hallucination Prevention

Week 9: Verification, Grounding, and Factuality

PhD Course in Agentic Artificial Intelligence

12-Week Research-Level Course

Learning Objectives

Bloom's Taxonomy Levels Covered

- **Remember**: Define hallucination (fabricated content), grounding (anchor to sources), FActScore
- **Understand**: Explain different hallucination types and their causes
- **Apply**: Implement Chain-of-Verification (CoVe) for fact-checking
- **Analyze**: Decompose claims into atomic facts for verification
- **Evaluate**: Assess factuality using FActScore and similar metrics
- **Create**: Design a multi-layer hallucination prevention pipeline

---

By end of lecture, you will understand how to build more factual agent systems.

What is Hallucination?

Definition

- Hallucination: LLM generates plausible but factually incorrect content
- Not random errors – often confidently stated and internally consistent

Why Hallucinations Occur

- **Training objective**: Next-token prediction, not factual accuracy
- **Parametric knowledge**: Facts encoded implicitly in weights
- **Pattern completion**: Model fills gaps with plausible content
- **No uncertainty signal**: Model doesn't know what it doesn't know

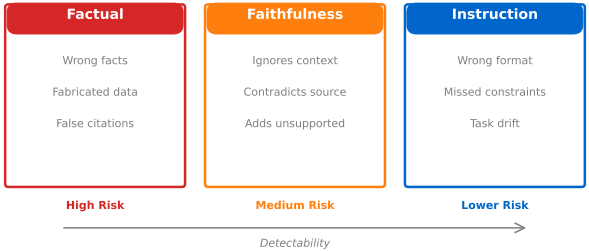
Impact on Agents

- Agent actions based on false information can cause real harm
- Compounds over multi-step reasoning chains

Hallucination is fundamental to how LLMs work, not a bug to be patched.

Hallucination Taxonomy

Types of LLM Hallucinations



Factual hallucinations are highest risk; instruction drift (straying from task) is most common.

Hallucination Types in Detail

Factual Hallucinations

- Fabricated facts: “Einstein won Nobel Prize in 1923” (was 1921)
- Non-existent entities: Citing papers or people that don’t exist
- Wrong relationships: “Company X acquired Y” (never happened)

Faithfulness Hallucinations

- Contradicts provided context or source documents
- Summarizes documents with added/changed information
- Common in RAG when generation ignores retrieved content

Instruction Hallucinations

- Ignores or misinterprets user instructions
- Generates unrequested content or format
- Fails to follow constraints (length, style, scope)

Different types require different prevention strategies.

Hallucination Detection Approaches

Self-Consistency Checking

- Generate multiple responses, check for agreement
- Inconsistency suggests uncertainty or hallucination

Retrieval-Based Verification

- Check claims against external knowledge sources
- Compare to retrieved documents (grounding)

Claim Decomposition

- Break response into atomic claims (single verifiable facts)
- Verify each claim independently
- FActScore: Precision of atomic facts against knowledge source

Model-Based Detection

- Train classifiers to detect hallucinated content
- Use LLM-as-judge to evaluate factuality

Combine multiple approaches for robust detection.

Confidence Calibration

The Calibration Problem

- LLMs often express high confidence even when wrong
- Verbalized confidence ("I'm 90% sure") poorly calibrated
- Token probabilities don't reflect factual accuracy

Calibration Approaches

- Verbalized confidence prompting ("express uncertainty")
- Multiple sampling + agreement rate
- Trained confidence classifiers
- Abstention policies (refuse uncertain queries)

Practical Strategies

- Add "express uncertainty when appropriate" to system prompt
- Sample 3-5 times, check consistency
- Set confidence threshold for human escalation

Well-calibrated confidence enables appropriate human oversight.

Chain-of-Verification (CoVe)

Core Idea (Dhuliawala et al., 2023)

- Generate initial response, then systematically verify it
- Use **independent** verification to avoid confirmation bias (favoring initial beliefs)

Four-Step Process

- **Step 1**: Generate baseline response
- **Step 2**: Plan verification questions for each claim
- **Step 3**: Answer verification questions independently
- **Step 4**: Generate final verified response

Key Insight

- Verification must be independent – don't show original response
- Prevents model from rationalizing its own hallucinations

Independence is crucial – the verifier must not see the claim being verified.

Chain-of-Verification Pipeline

Example:

"Paris is capital of France" → [Paris, France, capital] → "What is France capital?" → "Paris" (verified) → **Correct!**

Key: Independent verification prevents confirmation bias

Independent verification prevents confirmation bias.

FActScore: Measuring Factuality

Definition (Min et al., 2023)

- FActScore = Fraction of atomic facts that are supported by source
- Decomposes text into atomic facts, verifies each against knowledge

Process

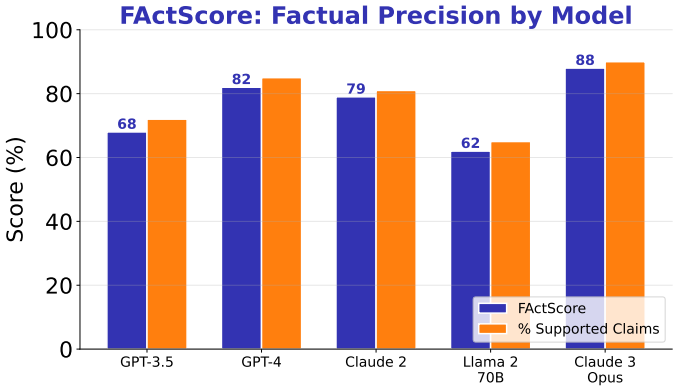
- **Decomposition**: Break text into atomic facts (single claims)
- **Retrieval**: Find relevant passages for each fact
- **Verification**: Check if passage supports the fact
- **Score**: Supported facts / Total facts

Example

- Text: "Einstein, born in Germany, won the 1921 Nobel Prize"
- Facts: (1) Einstein born in Germany, (2) Won Nobel 1921
- If both supported: FActScore = 1.0

FActScore provides fine-grained factuality measurement.

FActScore Evaluation



FActScore measures atomic fact precision against knowledge sources.

Hallucination Prevention Strategies

At Generation Time

- **Grounding**: Force citation of sources for all claims
- **Self-consistency**: Sample multiple times, take consensus
- **Constrained decoding**: Limit outputs to verified content

Architecture-Level

- **RAG**: Ground generation in retrieved documents
- **Multi-agent review**: Separate generator and critic agents
- **Tool use**: Use calculators, search APIs for facts

Post-Generation

- Chain-of-Verification (CoVe)
- FActScore evaluation and filtering
- Human review for high-stakes outputs

Layer multiple strategies for defense in depth.

Hallucination in Agentic Contexts

Agent-Specific Risks

- **Hallucinated tool calls**: Wrong function or parameters
- **Fabricated observations**: Agent "imagines" tool output
- **Planning hallucinations**: Invalid action sequences
- **State hallucinations**: Wrong beliefs about environment

Compounding Effect

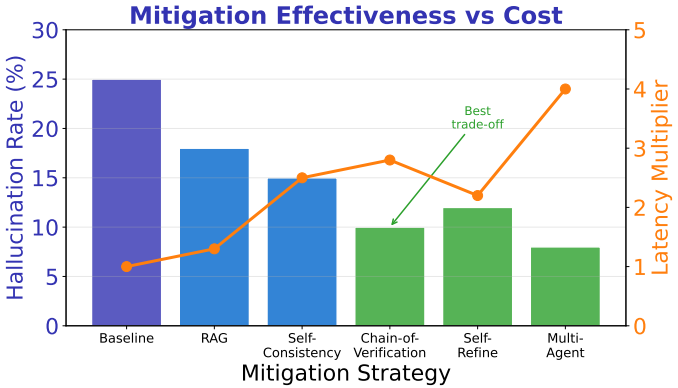
- Single hallucination propagates through trajectory
- Each step builds on potentially false previous outputs
- Error rate compounds: 95% per step $\rightarrow$ 60% after 10 steps

Agent-Specific Mitigations

- Tool output validation (schema, range checks)
- State verification between steps
- Checkpointing for rollback on detected inconsistency

Agent hallucinations compound – verify at every step, not just final output.

Mitigation Strategy Comparison



Chain-of-Verification offers best accuracy/latency trade-off.

Practical Implementation Guidelines

For Agent Developers

- Always ground high-stakes claims in retrieved documents
- Use explicit citation requirements in prompts
- Implement uncertainty signals ("I'm not sure", confidence scores)

For Production Systems

- Monitor FActScore or similar metrics over time
- A/B test prevention strategies on real queries
- Log verification failures for debugging

Trade-offs

- More verification = higher latency and cost
- Over-filtering = overly cautious, less useful responses
- Find balance based on risk tolerance of application

Prevention ¿ Detection ¿ Correction for production systems.

Required Readings

This Week

- Ji et al. (2023). "Survey of Hallucination in Natural Language Generation." arXiv:2202.03629
- Min et al. (2023). "FActScore: Fine-grained Atomic Evaluation of Factual Precision." arXiv:2305.14251
- Dhuliawala et al. (2023). "Chain-of-Verification Reduces Hallucination." arXiv:2309.11495

Supplementary

- Manakul et al. (2023). "SelfCheckGPT: Zero-Resource Black-Box Hallucination Detection." arXiv:2303.08896

Start with hallucination survey for taxonomy and scope.

Summary and Key Takeaways

Key Concepts

- **Hallucination Types**: Factual, faithfulness, instruction
- **Detection**: Claim decomposition, FActScore, self-consistency
- **Prevention**: Grounding, CoVe, multi-agent review

Design Principles

- Assume LLMs will hallucinate – design for it
- Independent verification prevents confirmation bias
- Layer multiple strategies for defense in depth

Next Week

- Agent Evaluation and Benchmarking

Prevention ¿ Detection ¿ Correction for production systems.