

Statistical Evaluation of Probabilistic Generative AI Models

NCS Conference 2026

Christian S. Schmid

Senior Statistical Scientist
F. Hoffmann-La Roche

What is Generative AI?

Generative AI (GenAI) is a type of artificial intelligence that can create new content rather than just analyze or categorize existing data.

What GenAI can Create:

Text

- Essays
- Emails
- Translations
- Summaries



Images



Artwork, Photos,
Designs

Music & Audio



Songs, Voice
Narration

Video



Animation,
Realistic Clips

Software Code



Programming, Scripts

All Images in this presentation were created by
Sora



Scope of Today's Talk

IN SCOPE



Locked and Static: No post-deployment learning

OUT OF SCOPE



Continuous Learning: Models that adapt independently.
Autonomous Agents: Self-directed behavior

CRITICAL CONSTRAINT: ⚠

The `model` must not change or
`update` after deployment until a
`controlled retraining event` occurs.

Can We Validate GenAI Applications?

Traditional GxP Software

- Rigid
- Deterministic
- Output is comprehensible

GenAI

- Inherently Creative
- Probabilistic
- Prone to completely make things up

Question: How can we get this beast under control?



Can We Validate GenAI Applications?

Traditional GxP Software

- Rigid
- Deterministic
- Output is comprehensible

GenAI

- Inherently Creative
- Probabilistic
- Prone to completely make things up

Question: How can we get this beast under control?

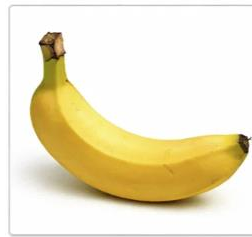
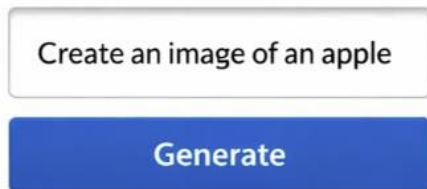


We have to move from the gold standard of verifying correctness, to a fuzzier standard of managing uncertainty!

The Challenge

In traditional software, the output is deterministic.

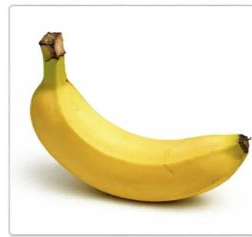
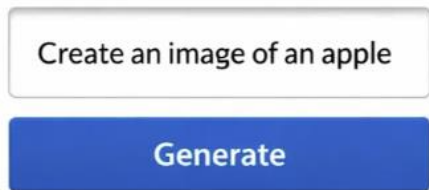
But an LLM is a probability engine.



The Challenge

In traditional software, the output is deterministic.

But an LLM is a probability engine.



The Challenge

If traditional software returns a wrong result, we don't call it hallucination. We call it a bug!



Bugs can be fixed in the code!

But, GenAI models are different. We can't rewrite code to have 100% certainty that a mistake won't happen again.

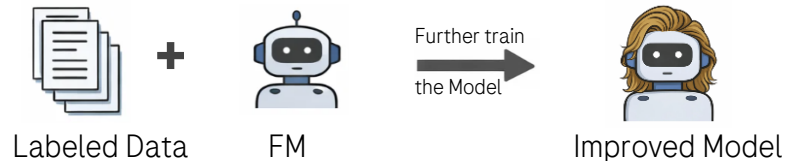
Question: How can we fix a 'bug' for a GenAI application?

Or better: How can we improve a model's answer?



How can GenAI Models be Improved?

1. Fine Tuning



Foundational Model (FM)

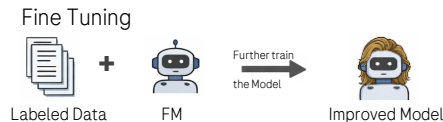
E.g., GPT-4, Claude Sonnet, Gemini 3 Flash

Further training a pre-trained model on specific data to specialize its behavior for a particular task.

Example: AI picture recognition of defect vials.



Evaluating Fine Tuning

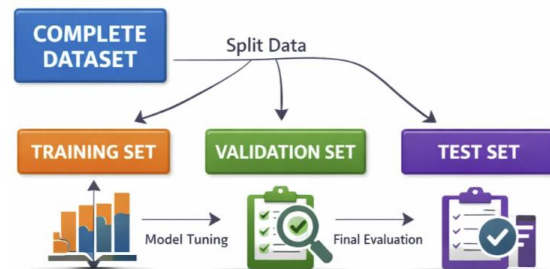


Further training a pre-trained model on specific data to specialize its behavior for a particular task. 

Example: AI picture recognition of defect vials.

Follows the same fundamental logic as classical machine learning evaluation:

1. Train (or “fine tune”) on labeled **training data**
2. Tune hyperparameters on **validation data**
3. Evaluate on held-out **test set**



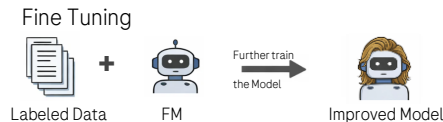
If a FM is fine tuned for a categorization task, standard statistical methods can be applied for evaluation!

E.g., confusion matrix, precision, recall, F1-Score, etc.

	Actually Cracked	Actually Not-Cracked
Predicted Cracked	x	y
Predicted Not Cracked	z	w



Evaluating Fine Tuning

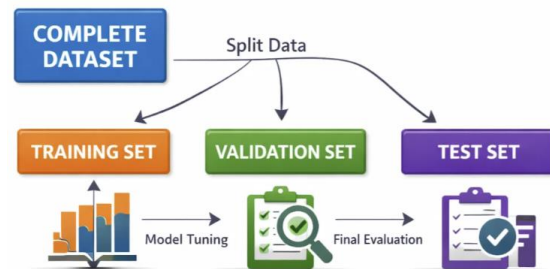


Further training a pre-trained model on specific data to specialize its behavior for a particular task. 

Example: AI picture recognition of defect vials.

Follows the same fundamental logic as classical machine learning evaluation:

1. Train (or “fine tune”) on labeled **training data**
2. Tune hyperparameters on **validation data**
3. Evaluate on held-out **test set**



If a FM is fine tuned for a categorization task, standard statistical methods can be applied for evaluation!

E.g., confusion matrix, precision, recall, F1-Score, etc.

Heads-Up: FMs can also ‘hallucinate’ new categories

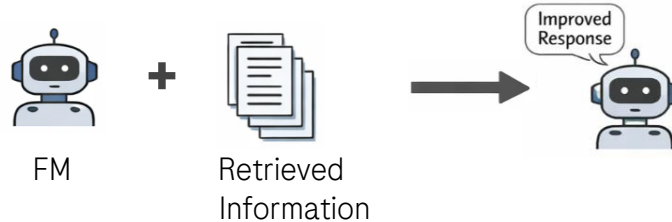
	Actually Cracked	Actually Not-Cracked
Predicted Cracked	x	y
Predicted Not Cracked	z	w
Invalid Category	u	v



How can GenAI Models be Improved?

1. Fine Tuning

1. RAG (Retriever Augmented Generation)



Foundational Model (FM)

E.g., GPT-4, Claude Sonnet, Gemini 3 Flash

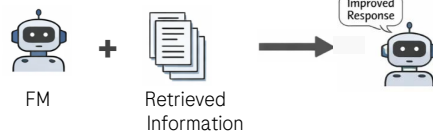
Improving responses by retrieving relevant external information and using it to generate more accurate answers.

Example: Adding company internal SOPs and policies for an internally used chatbot.



Evaluating RAG

RAG (Retriever Augmented Generation)



Improving responses by retrieving relevant external information and using it to generate more accurate answers.

Example: Adding company internal SOPs for an internally used chatbot.



Each sampling unit is a user query paired with its retrieved context and the gold/reference context.

Retriever Evaluation (RAG Context Selection)

Context Precision

Proportion of retrieved passages that are relevant to answering the query.

$$\text{Precision} = \frac{\text{Relevant Retrieved Chunks}}{\text{Total Retrieved Chunks}}$$

Context Recall

Proportion of all relevant passages retrieved from the corpus.

$$\text{Recall} = \frac{\text{Relevant Retrieved Chunks}}{\text{Total Relevant Chunks in Corpus}}$$



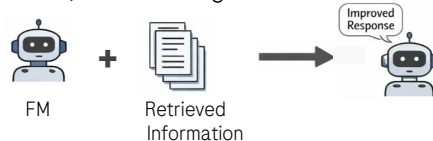
Before testing:

- Align on Sampling Strategy
- Set sample size n
- Align on passing criteria, e.g., Context Recall > 90%, $\alpha = 0.05$
- Prepare representative Query - Context Duplets:

Query	Gold Reference Chunks
Query 1	• Chunk 1a • Chunk 1b
Query 2	• Chunk 2a • Chunk 2b • Chunk 2c
⋮	⋮

Evaluating RAG

RAG (Retriever Augmented Generation)



Improving responses by retrieving relevant external information and using it to generate more accurate answers.

Example: Adding company internal SOPs for an internally used chatbot.



Each sampling unit is a user query/prompt paired with its retrieved context and the gold/reference context.

Retriever Evaluation (RAG Context Selection)

Context Precision

Proportion of retrieved passages that are relevant to answering the query.

$$\text{Precision} = \frac{\text{Relevant Retrieved Chunks}}{\text{Total Retrieved Chunks}}$$

Context Recall

Proportion of all relevant passages retrieved from the corpus.

$$\text{Recall} = \frac{\text{Relevant Retrieved Chunks}}{\text{Total Relevant Chunks in Corpus}}$$



Input Prompt:

“What are the storage conditions for drug X?”

Gold Context Chunks:

1. Store at 2-8°C
2. Protect from light
3. Humidity 30%-50%

Model Output:

“Keep refrigerated at 2-8°C, avoid light exposure.”

Relevant Retrieved Chunks:

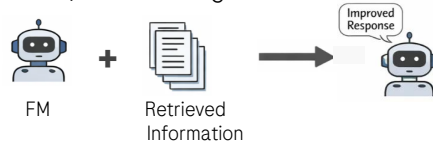
1. Store at 2-8°C
2. Protect from light

Context Recall:

2/3

Evaluating RAG

RAG (Retriever Augmented Generation)



Improving responses by retrieving relevant external information and using it to generate more accurate answers.

Example: Adding company internal SOPs for an internally used chatbot.



Each sampling unit is a user query/prompt paired with its retrieved context and the gold/reference context.

Generator Evaluation (Answer Quality)

Faithfulness

Answer is factually consistent with the retrieved context.

$$\text{Faithfulness} = \frac{\text{Supported Claims}}{\text{Total Claims}}$$

Groundedness

Answer is explicitly supported by the retrieved evidence.

$$\text{Groundedness} = \frac{\text{Supported Sentences}}{\text{Total Sentences}}$$



Input Prompt:

“What are the storage conditions for drug X?”

Gold Context Chunks:

1. Store at 2-8°C
2. Protect from light
3. Humidity 30%-50%

Model Output:

“Keep refrigerated at 2-8°C, avoid light exposure.
Must be stored in Switzerland.”

Total Claims in Output

3

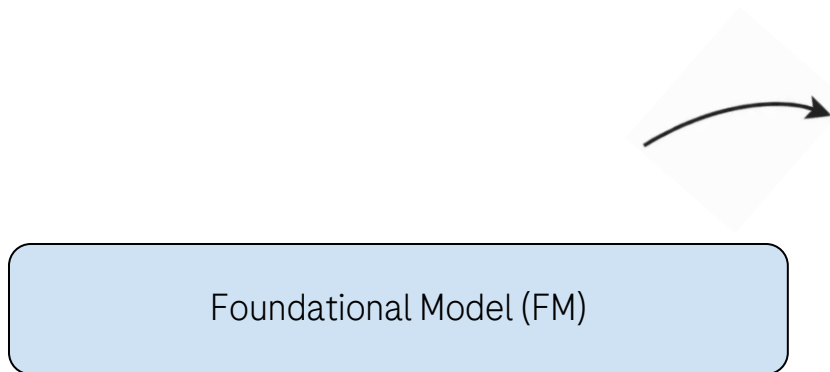
Supported Claims:

2

Faithfulness:

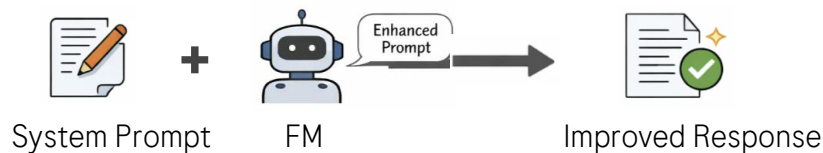
2/3

How can GenAI Models be Improved?



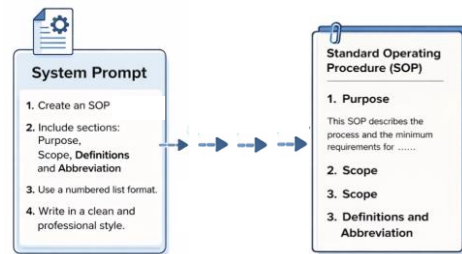
E.g., GPT-4, Claude Sonnet, Gemini 3 Flash

1. Fine Tuning
2. RAG (Retriever Augmented Generation)
3. Prompt Engineering



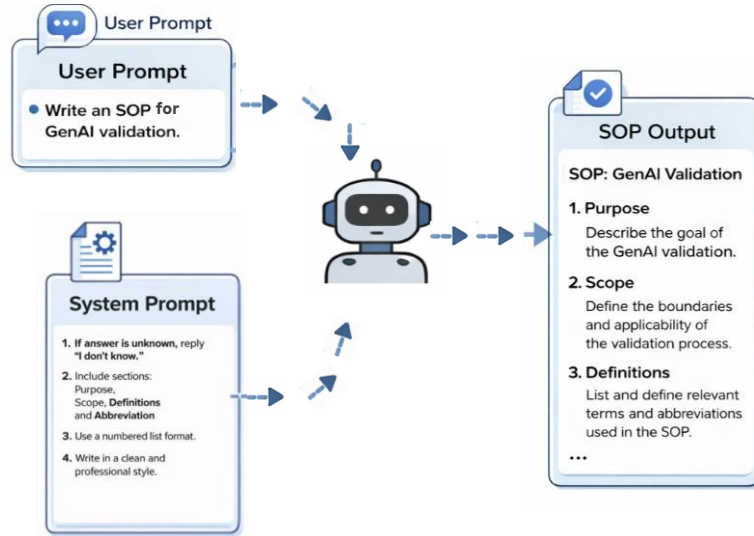
Designing and structuring system prompts to guide an AI model toward producing the desired output.

Example: An AI application that drafts SOPs.



Evaluating System Prompts

Designing and structuring system prompts to guide an AI model toward producing the desired output.



Test whether system prompt instructions were followed.

Example Question	Metric Type
Did the model respond "I don't know" for question it is not able to answer?	Binary
Did the model adhere to the sections Purpose, Scope, Definition, ...?	Binary
Is the content in "1. Purpose" correct?	Binary / Likert Scale
Was "1. Purpose" clean and professionally written?	Binary / Likert Scale

Summary

Shift to Uncertainty Management

GenAI is probabilistic, validation must evolve from deterministic verification to a standard of managing uncertainty.

Deployment Control

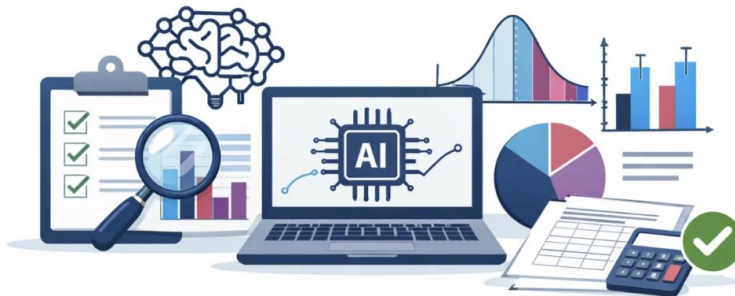
Focus remains on “Locked and Static” models to prevent uncontrolled post-deployment learning and ensure reproducibility in GxP environments.

Three Pillars of Reliability

Model performance is systematically improved and controlled through Fine-Tuning, RAG, and structured Prompt Engineering.

Statistical Validation Framework

This framework integrates classical ML metrics with specialized LLM evaluations like context recall and faithfulness to quantify more reliability.



Thank you for your Attention!



Guiding Principles of Good AI Practice in Drug Development

January 2026

1. Human-centric Design
2. Risk-based Approach
3. Adherence to Standards
4. Clear Context of Use
5. Multidisciplinary Expertise
6. Data Governance and Documentation
7. Model Design and Development Practices
8. Risk-based Performance Assessment
9. Life-Cycle Management
10. Clear, Essential Information

[...] evaluate the complete system [...] by validation of predictive performance through appropriately designed testing and evaluation methods.

Binary vs. Likert Scale

	Binary (Yes/No)	Likert Scales (e.g., 1-5 or 1-10)
Strengths	• Higher Inter-Rater Consensus • Faster Completion • Lower Rater Fatigue	• Captures Intensity and Direction • Requires Fewer Samples
Limitations	• Weak Power and Information Lost	• Subjectivity Bias between Raters



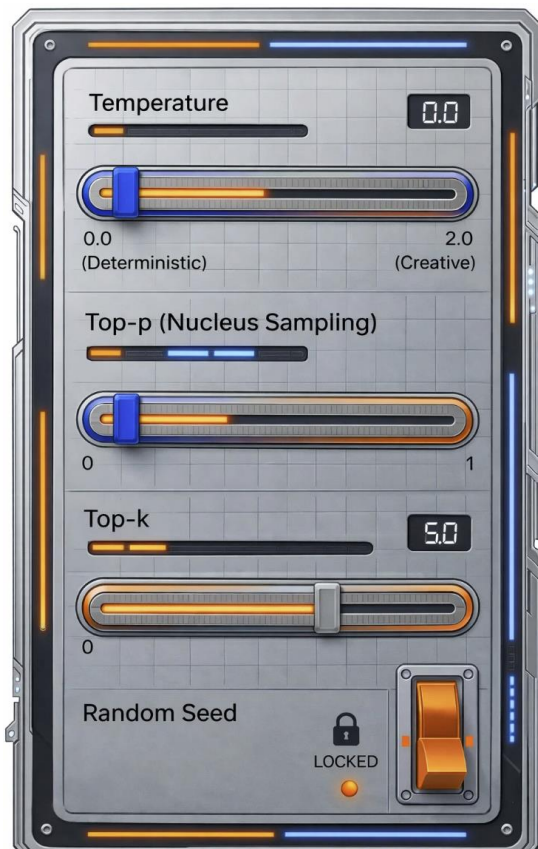
No



Yes



The Role of Important Hyperparameters



Controls the “randomness” or “creativity” of the output. It influences how the model samples from the probability distribution of possible next words.

$T = 0$ (called ‘Greedy Decoding’) means that the model will always pick the single most likely next word.

Limits the “pool” of candidate words by their cumulative probability p .

Limits the “pool” of candidate words by a fixed count by telling the model to only consider the k most likely words.

Ensures reproducibility by initializing the random number generator.