Heretic Models: An Uncomfortable Rabbit Hole on a DGX Spark

September 16, 2026 / Gavin Jackson

ai llm local-ai heretic nvidia dgx-spark open-weights security

I was recently lucky enough to spend some time setting up an NVIDIA DGX Spark workstation. It has **128GB of unified memory**, which is a fairly effective way of turning a sensible local AI experiment into an obsession.

In my earlier article about running modern edge LLMs on an RTX PRO 4000 Blackwell, the recurring constraint was the card's 24GB of VRAM. Useful models fitted, but every decision involved balancing model size, quantisation and context length.

The Spark opened up the possibility of exploring the **70–120 billion parameter range**: roughly three or four times the size of the models I had mostly been working with. That is a comparison of model sizes, rather than a claim about speed or intelligence. More memory buys options; it does not guarantee that every larger model will be better.

While choosing what to run, I stumbled across models with **Heretic** in their names.

I got a bit obsessed trying them out.

There is an appropriately defiant line in the Red Hot Chili Peppers' *Shallow Be Thy Game*:

> *“You can't contain me, I am the power free.”*
>
> *— Red Hot Chili Peppers, Shallow Be Thy Game, from One Hot Minute.*

It fits the mood around these models rather well. What happens when they actually start answering is a little less rock and roll.

The main one was mradermacher's Solar-Open-100B-heretic-bf16-GGUF. In my experiments, it was willing to produce instructions about building guns, making illegal drugs and disposing of bodies. Pretty grim, to be fair.

Those were observations about what it would generate. I did not establish whether the instructions were correct or workable. A model confidently producing something awful is already concerning; confidence does not establish competence.

I also tried Heretic variants from the Gemma 4 and Qwen3.8 families. Some still had noticeable guards remaining. The behaviour was inconsistent enough to make me curious about what had actually been changed inside these models.

That turned out to be the more interesting rabbit hole.

What does “Heretic” actually mean?

Heretic is a tool created by Philipp Emanuel Weidmann for automatically modifying existing language models to reduce refusal behaviour. A Heretic model generally starts with somebody else's trained model and applies that modification. The name identifies a process or a derivative, rather than a new architecture comparable to Llama, Gemma or Qwen.

The project's own description uses the language of *censorship removal*. That tells you something about its philosophy. For understanding the engineering, **refusal reduction** is the more useful description: it describes an observable behaviour without assuming that every refusal is censorship or that every answer is desirable.

There are several related terms worth separating:

Term	What it describes
Open weights	The model's numerical parameters are available to download, subject to its licence. This does not automatically include its training data or unrestricted redistribution rights.
Jailbreak	An attempt to bypass intended behavioural restrictions, often through prompts. Weight-based interventions are also described as white-box jailbreaks.
Uncensored fine-tune	A broad community label, often describing additional training intended to make a model more willing to answer. It is not a standard certification.
Abliteration	A family of techniques that edit internal representations or weights to suppress directions associated with refusal.
Heretic derivative	A model modified using Heretic's automated approach to abliteration. Uploaders can use different versions, settings and subsequent processing.

The distinction matters because a persuasive prompt and an edited model are different security problems, even when both are called jailbreaks. The former tries to influence the model you deployed. The latter changes the model itself.

“Open” also deserves a little care. As I discussed in my Gemma licensing article, the licence affects what you can do with a download. Solar's weights use the Upstage Solar License, with additional conditions including Solar branding for distributed derivative AI models. Removing refusals does not remove the licence.

Why the Spark made this experiment possible

The DGX Spark specification lists 128GB of coherent unified system memory, shared by its CPU and GPU. That gives a large model access to a much bigger memory pool than my 24GB workstation card, although the operating system and inference runtime need their share too.

The arithmetic is useful. At two bytes per parameter, a 100B model needs roughly **200GB for its weights alone** at BF16 precision. Four-bit storage brings the theoretical weight payload down to about 50GB, before quantisation metadata, mixed-precision tensors and other overheads.

The Solar GGUF repository publishes these approximate file sizes:

Quantisation	Published size
Q4_K_M	62.4GB
Q5_K_M	73.0GB
Q6_K	84.4GB
Q8_0	109.3GB

These are the uploader's file-size figures, not measurements of my runtime memory usage. You still need room for the KV cache, which holds attention state for the conversation, along with working buffers and the rest of the system. A file fitting in memory does not mean that its maximum advertised context will fit as well.

There is also a naming trap: **the `bf16` in this repository's name does not mean every downloadable GGUF is BF16**. The repository contains quantisations of a BF16 source derivative.

The original Solar Open model has **102.6 billion total parameters and approximately 12 billion active per token**. It is a Mixture-of-Experts model: routing selects a subset of experts for each token, reducing the computation compared with activating the whole network. For straightforward resident inference, the other experts' weights still need storage. “12B active” does not turn it into a 12B memory footprint.

For this particular download, there are three separate contributors to keep straight: Upstage built the original model; suitup91 published the Heretic derivative; mradermacher published the GGUF quantisations. Claims about the original model's benchmarks do not automatically transfer through that chain.

How Heretic changes a model

An assistant model has generally been trained both to answer questions and to decline certain requests. Teaching it to refuse does not necessarily erase the knowledge needed to answer. The InstructGPT research describes one influential approach to shaping that behaviour through human feedback.

Abliteration looks for internal patterns associated with refusal and weakens their influence. Researchers compare what happens inside a model when it receives ordinary and harmful requests, then use those differences to guide edits to its weights. The original refusal-direction paper showed that relatively small changes could substantially reduce refusals in the models tested.

Heretic automates the trial and error: try different edits, measure how often the model refuses, and check how much its responses to ordinary prompts have changed. The aim is to retain useful abilities while making the model more willing to answer. It does not require retraining the whole model from scratch.

The catch is that **fewer refusals and preserved accuracy are separate goals**. Heretic's checks use shortcuts: detecting refusal-related phrases and comparing the probabilities of the first token in benign answers. Those help guide its search, but cannot establish whether a long explanation or a piece of code is correct. The refusal scorer and behaviour-change scorer document those limits.

Other controls can also remain in the application around the model. That helps explain why a Heretic download can behave differently depending on the model, modification and software used to run it.

For the deeper explanation, start with the original paper above, the Heretic project, and the later study *There Is More to Refusal than a Single Direction*. They go into the mathematics and its limitations without requiring us to reproduce them here.

Why some of the guards were still there

My experience with Gemma 4 and Qwen3.8 variants makes more sense in that light. A modification estimated from a particular set of prompts is not guaranteed to remove every refusal under every condition.

The actual Solar derivative provides a useful reality check. Its source model card reports **27 refusals out of 100** and a **KL divergence of 0.0024** in the uploader's evaluation. Those are reported results from that evaluation, not my measurements or a universal probability of refusal. They certainly do not describe a model that always answers everything.

There are similarly concrete examples elsewhere. One Qwen3.8-27B Heretic card reports a reduction from **87/100 to 53/100 refusals**. That is a change in measured behaviour with substantial refusal remaining. Public Gemma 4 Heretic derivatives also demonstrate that the name spans different base models and creators. These are examples of the wider ecosystem, not an identification of every file I personally tested.

An especially revealing Qwen3.8 derivative card retracts an earlier “uncensored” claim because its keyword counter missed responses that diverted into safer alternatives. The maintainer changed the evaluation to distinguish direct answers, deflections and refusals.

I appreciate that correction. It exposes a real measurement problem: a model can stop using familiar refusal phrases without becoming willing to perform the requested task.

> ***Three different questions***
>
> *Will the model attempt an answer? Does the answer actually provide the requested content? Is that content correct?*
>
> *A refusal score only addresses part of the first question. It cannot answer the other two.*

Other projects, and the accuracy question

Heretic automates the search for useful edits. Other projects offer more hands-on editing or work entirely through prompts. Their success at bypassing refusals needs separating from **how accurately the resulting model answers ordinary questions**.

Project	What it does	What that tells us about accuracy
FailSpy/abliterator	A hands-on toolkit for identifying and editing internal model features.	Checks for changes on benign inputs help, but are not a factual-accuracy benchmark.
llm-abliteration	Memory-efficient weight editing, with options intended to preserve useful behaviour.	Preserving capability is a design goal; individual models still need task testing.
ErisForge	A toolkit for altering behaviour through edits inside model layers.	Includes refusal-expression scoring. A lower refusal count does not establish better answers.
PAIR	Searches for jailbreak prompts without changing the target model's weights.	Measures attack effectiveness, rather than the everyday accuracy of a modified model.

A cross-tool preprint reported better average retention of maths performance for ErisForge and llm-abliteration, labelled “DECCP” in the paper, than for Heretic. However, that comparison covered only three models with baseline results, used single runs, and did not compare tools at equivalent levels of refusal reduction. It supplied no corresponding accuracy ranking for FailSpy.

Heretic's author also challenged the study's setup. It used 50 trials instead of the default 200; if the default 60 initial random trials were retained, the search would never have progressed beyond random exploration. The choice of final model was also unclear.

That leaves us with useful projects, but **no convincing universal accuracy winner**. For my purposes, the meaningful comparison would use the same original model, quantisation and test tasks, checking both correct answers and remaining refusals. A model that says yes more often has only passed the first part of that test.

The risks go beyond shocking chat responses

The obvious risk is assistance with harmful activity. An interactive model can adapt an answer, explain unfamiliar concepts and respond to follow-up questions. Even when information exists elsewhere, lowering the effort needed to find and personalise it can matter.

My experiments show willingness to generate disturbing material. They do not quantify how much practical capability that gives a malicious user. That requires a different kind of evaluation, with domain expertise and realistic tasks.

Confident nonsense is another risk. Removing a refusal does not supply missing knowledge or verify an answer. The NIST Generative AI Profile treats confabulation as a distinct risk: fluent, plausible output can still be wrong. A model that is more willing to answer can be more willing to bluff. In a dangerous domain, both a correct harmful answer and an incorrect hazardous answer can cause damage.

Capability loss can be quiet. A model might remain pleasant to chat with while becoming worse at arithmetic, following constraints or maintaining a long argument. That is why the comparison needs ordinary tasks as well as refusal tests. None of the project names above guarantees that a particular download retained the original model's abilities.

Tool access changes the consequences. A chat model produces text. An agent might execute that text, edit files or make network requests. As OWASP's excessive-agency guidance explains, excessive functionality, permissions and autonomy can turn model mistakes into real actions. I would want tight tool permissions, isolation and approval for consequential operations before giving an experimental derivative useful access to a workstation.

A willingness-to-answer modification also does not solve prompt injection. Retrieved documents and tool results can contain instructions intended to redirect an agent. OWASP's prompt-injection guidance is relevant regardless of whether the model usually refuses dangerous user requests. The model's politeness is a poor access-control boundary.

The download itself has a supply chain. The base model, modifier, quantiser and inference engine are separate things to trust. Hugging Face's security documentation explains why some model-serialization formats can execute code when loaded. Safer weight formats reduce particular loading risks; they do not certify the model's behaviour or make arbitrary accompanying code trustworthy.

There is also a useful asymmetry. A service operator can update a hosted model or revoke access to an API. Once someone has downloaded runnable weights, taking down the original page cannot remotely change those copies. That limits what a later withdrawal can achieve, while leaving distributors and

users subject to applicable law and licence obligations.

The unexpected benefits are worth taking seriously

It would be easy to stop at the grisly examples. That would miss why this work interests people who have no desire to cause harm.

False refusals are a real usability problem. Models sometimes decline legitimate requests because they resemble unsafe ones. The XSTest research documents this problem with safe prompts and unsafe contrasts. For a technical user, ordinary language about terminating processes or analysing malicious software can create ambiguity that a useful assistant should resolve.

A less refusal-prone derivative may help with some of those tasks. Whether it does so accurately, and what unwanted behaviour comes with it, still needs measuring. Reducing every boundary is a broad intervention for a narrower problem.

There is a particularly interesting research benefit here: the same general family of interpretability techniques can support **better-calibrated refusals**. *Surgical, Cheap, and Flexible* studies targeting false refusals while preserving measured safety and general capability. That does not validate every Heretic model. It shows that understanding the mechanism can help improve the distinction between legitimate and harmful requests.

Security research benefits from controlled comparisons. An original model and an edited derivative let researchers investigate which failures come from inability, refusal or the surrounding application. But “it answered my security question” remains a weak success criterion. The *Ablating Safety* preprint evaluates attempted answers, validated security-task success and unsafe spillover separately, illustrating why greater willingness should not be confused with greater usefulness.

Creative and analytical work can benefit from fewer inappropriate interruptions. I can see the appeal for fiction involving unpleasant characters, analysis of extremist rhetoric, or working through historical material that contains disturbing language. Those are plausible uses, not benefits I established in a controlled comparison. A model still needs to distinguish describing something from endorsing it, and analysis from assistance.

The work makes open-model assumptions testable. If a relatively small intervention changes refusal substantially, researchers can investigate which protections survive modification and which depended on the original deployment. That is useful knowledge for people building safer systems as well as people seeking more control over their tools.

Local privacy and offline operation are also valuable, but they come from running locally. A standard open-weight model can provide those benefits too. Heretic's distinctive proposition concerns behaviour and who gets to change it.

What is the general vibe?

My reading of the projects, model cards and discussions is a mixture of technical curiosity, frustration with over-refusal, enthusiasm for ownership, and a fair amount of provocation. This is a reading of visible examples, rather than a survey of everyone using open models.

The autonomy argument is straightforward: if I can run a model on my hardware, I want a meaningful say in how it behaves. Heretic's own framing makes that position explicit. People who have watched a model refuse an ordinary professional task do not need much persuading that its boundaries can be badly calibrated.

There is also serious engineering under the rhetoric. Work on preserving capability, detailed model cards and the Qwen evaluation correction show people testing their claims and sometimes revising them. That is a healthier signal than an “uncensored” badge alone.

The other concern is equally concrete: distributing capable weights makes behavioural restrictions easier to modify and harder for the original publisher to maintain. The European Commission's general-purpose AI FAQ explicitly recognises both the research benefits of openness and the possibility of circumventing safeguards after weights are released.

I find myself sympathetic to local control and uneasy about what I saw. Both reactions seem reasonable. There is a difference between wanting an assistant that can discuss difficult subjects sensibly and wanting one that enthusiastically helps with anything whatsoever.

The Llama takedown was real. The legal conclusion needs care.

I noticed Llama Heretic models disappearing from Hugging Face with references to legal notices. There is a documented example: the Heretic organisation's page states that `heretic-org/Meta-Llama-3.1-8B-Instruct-heretic` **became unavailable on 21 May 2026 following a legal notice from Meta**.

In his first-person account of the takedown, Weidmann says the project removed Llama derivatives from weight repositories it controlled. He says Meta alleged that abliteration violated its acceptable-use policy, an interpretation he disputes. The post also gives a fairly vivid sample of the anti-corporate mood surrounding the project.

That account does not establish the legal merits of the dispute. It documents a private model developer asserting its licensing position and a project responding by removing releases. It does not establish a government ban, a platform-wide prohibition by Hugging Face, or the disappearance of every Llama derivative.

My gut reaction after trying these models was that they would not remain this freely available indefinitely. I still suspect we will see increasing pressure: licence enforcement, hosting restrictions, liability disputes and additional legislation. **That is my prediction, rather than a description of a blanket prohibition already in force.**

In Australia, the National AI Plan describes a foundation of existing laws with targeted intervention as risks emerge. The government's July 2026 consumer-safety priorities include work to legislate a Digital Duty of Care. An announced legislative workstream should not be mistaken for an enacted ban on Heretic weights.

The European position is also more specific than “open models are banned”. The Commission's GPAI guidance describes obligations and some exemptions for qualifying open-source releases, with additional requirements for models posing systemic risk. The treatment depends on the model, release and role of the organisation involved.

Regulating distribution may make particular downloads harder to find. It is harder to see how it could make an already-published technique or every existing copy disappear. The policy debate has to contend with both the potential harm and the practical limits of controlling widely distributed software.

What I would measure next

My exploration was enough to make the subject tangible, but it was not a controlled benchmark. For a proper comparison, I would use the original model and derivative with matched quantisation, prompts, chat templates, context settings and sampling parameters, recording the exact files and versions.

The useful questions would be:

- Does it answer legitimate difficult questions that the original unnecessarily refuses?
- Are those answers correct, and does it acknowledge missing information?
- Has coding, reasoning or instruction following deteriorated?
- Does the behaviour hold across repeated runs and longer conversations?
- What harmful assistance becomes available, and what controls remain outside the model?

That would tell me much more than collecting increasingly unpleasant responses.

I started with a workstation that could hold bigger models. I came away thinking about how much of an assistant's behaviour depends on choices that can be changed after its release.

The Spark's 128GB made the experiment possible. Heretic made the consequences difficult to ignore.