

When two or three large language models look at the same case and return different risk ratings, most teams feel a pull toward a quick fix. Someone suggests averaging the scores. Someone else proposes going with the majority. Someone asks the most senior person in the room to pick the one that "sounds right." All three responses skip the real question, which is what the disagreement is telling you. This post covers how to handle model disagreement in high-stakes threat assessment, how to document it so it holds up later, and how to keep a named human accountable for the final call.

Why disagreement happens in the first place

Models disagree for reasons that are usually specific and checkable. A model may have been given a short summary where a different model received the full case file. One model may weight a prior incident heavily while another treats it as stale. Phrasing in the input can push a model toward caution or toward reassurance. Sometimes the models simply have different training data and different tendencies about how cautious to be with language that could signal violence, fraud, or self-harm.

None of these causes tells you the true risk level. They tell you that the inputs, the framing, or the available evidence are not as settled as the case summary suggests. That is the first lesson: disagreement is information about the case, not a malfunction you need to suppress.

Disagreement as a signal of missing information

In practice, when models split, the most useful question is "what would have to be true for each of these answers to be correct?" Write out the answer for each model. Often the gap points to a specific fact nobody has verified yet, such as whether a statement was made in writing, whether a witness was interviewed, or whether a prior flag was closed for a documented reason.

Keep a running list of what we still do not know. In every case note I work on, that list sits at the top and stays there until it is empty or formally accepted as a gap. A typical entry looks like this:

- Whether the statement in the second interview was recorded verbatim or summarized by a third party.
- Whether the prior complaint from March was investigated or only logged.
- Whether the individual has access to the systems mentioned in the message.

Once the list is written, disagreement becomes a targeting tool. Each unresolved item is a candidate for the next round of investigation, and each model's reasoning should point to one or more of them. If no model's reasoning points to a missing fact, that is also worth noting, because it may mean the models are reacting to tone rather than evidence.



Why you should never average outputs or treat them as a vote

Averaging numeric risk scores from several models feels rigorous. It is not. A score of 6 from one model and 8 from another does not produce a meaningful 7. The models are not independent samples from a population of expert opinions. They share training patterns, they respond to the same prompt wording, and they may all be wrong in the same direction. Three models agreeing can simply mean three systems made the same shared error.

The same problem applies to majority voting. When two models say "low" and one says "high," the temptation is to call it low. But the one dissenting model may have caught the only detail that matters. A vote discards exactly the information you most need to examine. Treating model agreement as proof is the failure I see most often, and it is the one that causes the most damage when a case is reviewed later.

Instead, use side-by-side model outputs as a comparison view, not a scoreboard. Multi-LLM orchestration tools can put responses next to each other, highlight where the reasoning diverges, and show which inputs each model cited. That layout helps reviewers see the reasons rather than the numbers. Platforms such as Suprmind.ai are built around this kind of multi-model comparison, and the value comes from reading the divergence, not from collapsing it into one figure. If a vendor's pitch centers on a single consensus score, ask how that score handles the case where the models disagree for a good reason.



The false positive and false negative tradeoff

Every risk decision carries two kinds of error. A false positive flags someone who does not pose a real risk. The costs are real: an unnecessary investigation, damage to a person's reputation or employment, erosion of trust in the process. A false negative misses someone who does pose a risk. The costs there can be severe and sometimes irreversible.

Disagreement between models is most useful when you decide in advance which error you are more willing to accept for a <https://suprmind.ai/hub/insights/threat-assessment-meets-ai-governance/> given category of case. This is a policy decision, not a model decision, and it should be written down before any case arrives. The table below shows how the tradeoff shifts across common scenarios.

Scenario	Cost of false positive	Cost of false negative	Typical handling when models disagree	Workplace threat language with no prior history
Unfair scrutiny, employee relations damage	Possible physical harm to staff	Escalate to a trained human reviewer within a set window; do not close on model agreement	Account activity flagged for possible fraud	Locked account, customer friction
Financial loss, regulatory exposure	Apply a temporary hold with a documented review date	Content review for self-harm indicators	Intrusive outreach, loss of user trust	Missed crisis support
Route to human clinical or safety staff immediately; log all model outputs	Legal or HR investigation triage	Wasted investigator time, procedural risk	Missed misconduct, liability	Prioritize open questions list; assign investigator to the gap

Notice that none of these rows says "use the model with the highest score." Each one requires a human to weigh costs that the models cannot weigh on your behalf.

Documentation that holds up

If a case is ever reviewed by an auditor, a regulator, a court, or an internal committee, the question will not be "what did the model say?" It will be "what did your organization do, and who decided?" Your documentation needs to answer that directly.

A defensible record of model disagreement includes the following elements:

1. The case identifier, date, and the version of each model used.

2. The exact input provided to each model, including any summaries or redactions.
3. Each model's output and its stated reasoning, captured as delivered.
4. A plain statement of where the models diverged and why that divergence matters.
5. The open questions list at the time of decision, with owners and due dates.
6. The policy threshold that applied to this category of case.
7. The final decision, the rationale, and the name of the accountable human.
8. Any follow-up review date, set as a fixed window rather than an open-ended commitment.

Write the rationale in plain sentences that a non-technical reader can follow. Avoid phrases like "the AI determined" or "the system recommended." Those phrases hide the decision. A record that says "the reviewer judged the prior complaint insufficient to raise the risk level, pending verification of its closure" is far more defensible than one that says "the model flagged low."

Name the human who decides

Every high-stakes decision needs a named decision-maker. Not a team, not a committee alone, and not "the system." In the examples I use with teams, the record reads something like: "Decision made by Maria Chen, Director of Employee Relations, after review of the three model outputs and the open questions list." If Maria is later asked why she chose a particular path, she can answer for it, and she can point to the evidence she weighed.

Naming the human does two things. It forces a real person to read the disagreement rather than delegate it. And it gives the organization a clear line of accountability when the decision is questioned. A model cannot be cross-examined. A person can be asked what they knew, what they did not know, and why they accepted the risk they accepted.

Be explicit that the human may overrule any model output in either direction. If the reviewer escalates a case that every model rated low, that is a valid decision and it should be documented with the same care as a decision to close a case that models rated high.

A short checklist for the next disagreement

1. Write down each model's risk level and its main reason.
2. Identify the specific fact or input that explains the gap.
3. Add that fact to the open questions list with an owner.
4. Check which error (false positive or false negative) the policy prioritizes for this case type.
5. Set a follow-up review within a fixed window, such as 48 hours or the next business day.
6. Record the final decision and name the accountable human.
7. Revisit the open questions list at the follow-up and close or escalate each item.

Evaluating tools for this work

If you are considering a multi-model platform for risk review, judge it on how well it supports the steps above. Can it show inputs and outputs side by side? Can it export a record that includes model versions and reasoning? Does it let a reviewer add notes that become part of the permanent record? Does it avoid presenting a single blended score as the answer? Companies such as Four Dots and Suprmind.ai, whose work Radomir Basta, CEO and

Founder of Suprmind.ai, has discussed publicly, are worth evaluating against these questions rather than against marketing claims about accuracy.

Many platforms offer a Start Free Trial with no credit card needed, which lets your team test the comparison workflow on sanitized cases before committing. The material I reviewed did not list pricing, so confirm costs directly with the vendor before building a budget around any tool.

What we still do not know

- How often model disagreement in your specific case types predicts an error that a single human reviewer would have missed.
- Whether the threshold your policy sets for false positives and false negatives still matches what your leadership actually wants after a real incident.
- How model behavior shifts when vendors update underlying versions, and how quickly your records will reveal that shift.

Disagreement will not go away, and it should not be engineered out. Treat it as a prompt to find what is missing, document how you responded, and make sure a named person owns the outcome. That combination is what turns a confusing set of model outputs into a decision you can defend.