

Short answer: yes, in most high-stakes settings it is a bad idea. Longer answer: the problem is not arithmetic. The problem is that averaging quietly turns a question about a specific person's behavior into a question about which number sits in the middle. In threat assessment, HR investigations, legal review, and security triage, that substitution can cause real harm.

This post explains why teams keep reaching for averaging, where it goes wrong, and what to do instead. It also covers how multi-LLM orchestration tools and side-by-side model outputs can help, provided people remain responsible for the decision.

Why teams average model outputs in the first place

The appeal is understandable. You have three or four models, each returns a risk score or a label, and the results disagree. Averaging feels like a way to reduce noise and avoid cherry-picking. It looks rigorous. It produces one number for a dashboard. It also lets someone say, "We used multiple models," which sounds like due diligence.

The trouble starts with what the average represents. A mean of model scores is not an estimate of the probability that someone poses a risk. It is the midpoint of several outputs that may share training data, share prompt framing, and share the same blind spots. Agreement among them is not independent confirmation.

Three common assumptions are worth challenging directly:

- **Agreement equals accuracy.** Models trained on similar corpora can agree confidently and still be wrong for the same reason.
- **Each model is an independent voter.** In practice, they often respond to the same ambiguous phrasing in the same way.
- **A higher average is a safer average.** That depends entirely on which error you are trying to avoid.

The false positive and false negative tradeoff

Any risk assessment involves two kinds of error. A false positive flags someone who does not pose a meaningful risk. A false negative misses someone who does. The costs are rarely symmetrical, and they are rarely the same across contexts.

In a workplace investigation, a false positive can damage an employee's reputation, derail their career, and expose the organization to retaliation claims. A false negative can leave a real threat unaddressed. Neither outcome is acceptable, but they demand different levels of scrutiny and different escalation paths.



Averaging hides this tradeoff. A score of 0.5 from a mean of 0.9 and 0.1 looks the same as a score of 0.5 from three models all saying "moderate." Those are completely different situations. One reflects a sharp disagreement that needs investigation. The other reflects a consistent middling read. The average erases the difference.

Scenario What the average shows What is actually happening Right next step Models split 0.9 and 0.1 0.5, "moderate" Sharp disagreement, likely missing context Review inputs, gather more information Models consistently at 0.5 0.5, "moderate" Consistent ambiguity Define what evidence would change the read Three models share one framing error High confidence Correlated error, not validated risk Test the framing with a different source

Disagreement is a signal, not a defect

When models disagree, the instinct is to resolve the disagreement as quickly as possible. That instinct is often backwards. Disagreement usually means one of three things: the input is incomplete, the input is ambiguous, or the question being asked is not well defined.

Treat disagreement as a diagnostic. It tells you where the information is thin. If one model reads a message as threatening and another reads it as venting, the useful question is what additional context would separate those readings. Was there a prior incident? What was the tone of the surrounding messages? Is there a relationship history the models never saw?

Side-by-side model outputs make this visible. Instead of a single blended score, reviewers see each model's reasoning, the points where they diverge, and the evidence each one cited. That view supports a conversation. A single average supports a checkbox.

What multi-LLM tools are good for

Multi-LLM orchestration tools are useful when they expose differences rather than hide them. Good uses include:

- Surfacing where models cite different evidence from the same source material
- Flagging when a prompt produces unstable outputs across runs or providers
- Helping analysts notice that a conclusion depends on one assumption
- Creating a record of what was asked, what was returned, and what was done with it

Less useful uses include generating a consensus label, weighting models by vendor reputation, or treating a majority as proof. Vendors in this space, including Suprmind.ai, which is led by CEO and founder Radomir Basta, position their tooling around comparing outputs rather than collapsing them. Whatever tool you evaluate, ask it to show its disagreements plainly. If it cannot, that is a product gap, not a feature.

Teams piloting these tools often start with a free trial. Start Free Trial (no credit card needed) is a low-risk way to test whether side-by-side views fit your review process. The trial does not answer whether the tool's outputs are valid for your use case. That requires your own evaluation, your own labeled cases, and your own error analysis.

Do not average outputs, and do not treat them as a vote

This is the core rule. Here is why it holds up under scrutiny:



1. **Averages assume comparable, independent measurements.** LLM risk scores are neither reliably comparable across models nor independent.
2. **Votes assume the voters know the answer.** Models do not have ground truth about the person being assessed. They have text.
3. **Consensus can launder bias.** If every model was prompted with the same loaded framing, unanimity reflects the framing.
4. **Numbers invite false precision.** A score of 0.62 looks more reliable than a paragraph explaining uncertainty, even when the paragraph is more accurate.

If you need a single output for a workflow, produce a disposition instead: for example, "no action," "review with additional context," or "escalate to named investigator." Define the criteria for each in advance, and record which evidence supported the disposition.

Human judgment required, with named accountability

Every high-stakes assessment should have a named human who owns the decision. Not a team, not a model version, not a "process." A person. In our documentation, we name the decision-maker in every case file, along

with the date, the evidence reviewed, and the alternatives considered.

Want to know something interesting? here is a concrete example. Suppose a people-operations lead, Dana Whitfield, receives a threat assessment flagged by an internal tool. The models disagree. Dana's record should show:

- The question she was actually answering
- Which model outputs she reviewed, and what each said
- What she asked for from HR records, prior incident logs, or the employee's manager
- Her decision, her reasoning, and the people she consulted
- Who has authority to override her decision, and how that override would be recorded

If the record says "the AI said so," the decision is not defensible. If it says "Dana Whitfield reviewed three model outputs, two of which cited the same message out of context, and requested the full thread before deciding," it is.

A short checklist for defensible documentation

- Named decision-maker and their role
- Question being answered, stated in one sentence
- Inputs reviewed, including what was excluded and why
- Model outputs retained in full, not just a summary score
- Points of disagreement and how each was investigated
- Disposition and the evidence that supports it
- Review date for a time-boxed follow-up

What we still do not know

Honest documentation includes open questions. Here is our running list for this topic:

- Whether disagreement rates between models predict error rates in our specific context
- How often correlated framing errors appear in our prompts, and whether we can measure them
- Whether reviewers anchor on the first model output they read, regardless of order
- How to compare evaluation results across vendors that do not expose comparable internals
- Whether Four Dots or similar agencies' published comparisons of AI tools use methods we would accept for regulated decisions

Follow-up: revisit this list at the end of the next evaluation cycle, and close or add items. Time-box each open question to one review period, then decide whether it needs a formal study or can be dropped.

Bottom line

Averaging LLM risk scores like a vote feels efficient and produces tidy numbers. It also obscures the information that matters most: where the models disagree, why, and whether the available evidence is enough to decide. In high-stakes threat assessment, that obscuring is the failure.

Do not average outputs. Do not treat agreement as proof. Use multi-LLM tools and side-by-side outputs to find gaps in your information. Name the person accountable for each decision, document [threat assessment workflow](#)

the reasoning in plain language, and keep a visible list of what you still do not know. Human judgment is not a fallback for when the AI is unsure. It is the point of the process.