

VegaML 0.8B against Laya

Typed decisions on a Mac mini: accuracy, honesty, speed and images

15 October 2026

Recommendation: keep Laya for text decisions. On the same 300 decisions, Laya is more accurate, three times faster on the Mac's GPU and lighter in memory. Vega 0.8B's abstain flag does not pick out its mistakes. Vega earns a place where a decision has to be read from an image, and it edges Laya on the developer's phishing task.

The question

The developer behind Laya released VegaML on 9 October 2026. It makes the same kind of typed decision: pick one option, give a level, or say yes or no, each with a probability. Does it decide better than Laya on this Mac, and at what cost?

The two models

	Laya	VegaML 0.8B
Released	2026-09-18	2026-10-09, version 0.9.0 tested
Size	421M parameters	0.8B frozen Qwen3.5 backbone plus a 14.3M-parameter decision engine
Inputs	text	text and images, up to 73,728 tokens
Answer types	choice, score, yes/no	choice, score, yes/no (called boolean)
Extras	calibrated probabilities	calibrated probabilities, conformal answer sets, an abstain flag
Licence	Apache 2.0	Apache 2.0

Vega also comes in a 4B size. Its backbone runs in 32-bit floats on a Mac, which would need about 18 GB, so it does not fit this 16 GB machine and was not tested.

Method

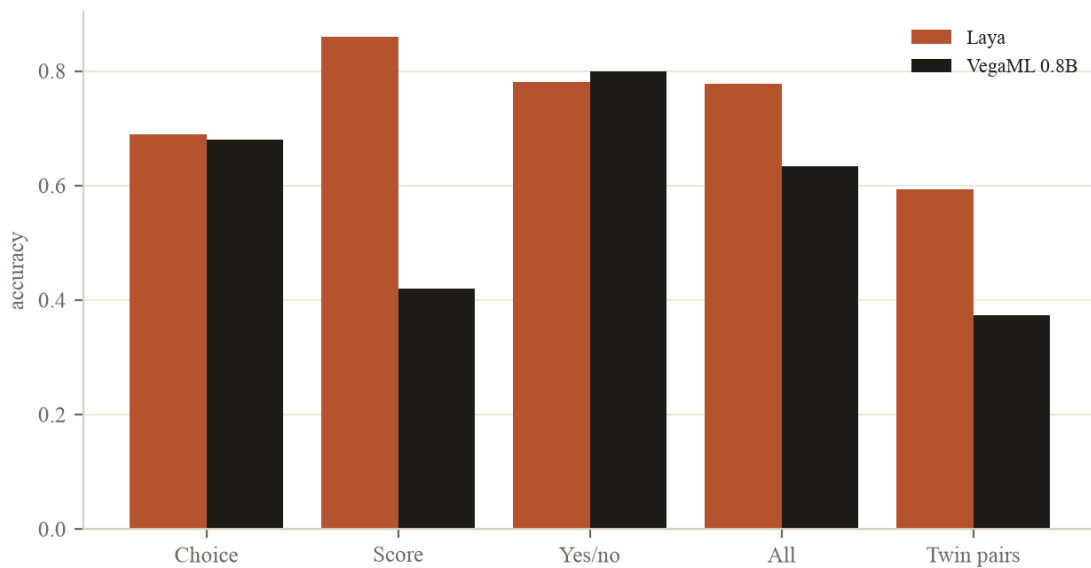
- **Machine.** Mac mini, Apple M4, 16 GB. The Mac's GPU was used through PyTorch's Metal backend, with one extra run on CPU.
- **The test set.** It is the same sealed set Laya was tested on in September: 300 synthetic records, 100 each of choice, score and yes/no. An independent author wrote the records and their labels before any model saw them.
- Every record has a twin that differs in one detail, and the right answer flips with that detail.
- The traps are quantity thresholds, authority against a report, nested clauses, instructions overridden later, negation, time shifts, resolved against open, and keyword decoys.
- A keyword matcher scores 0.87 on choice, 0.73 on score and 0.66 on yes/no.
- **Scoring.**
- Accuracy counts a hit on each model's own answer.

- Pair accuracy counts a twin pair only when both halves are right. It separates a reader from a keyword matcher.
- Calibration error (ECE, ten bins) measures how far stated confidence sits from actual accuracy.
- Latency is the median per decision.
- **Repeats.** Each model ran three times on the GPU, and every run gave the same answers. Times are medians.

Results: text decisions

	Laya	Vega 0.8B
Accuracy, all 300	0.777	0.633
Choice	0.69	0.68
Score	0.86	0.42
Yes/no	0.78	0.80
Twin pairs both right	59%	37%
Calibration error, lower is better	0.119	0.126
Abstained	0	65
Accuracy on the items it answered	0.777 of 300	0.643 of 235

- **Laya wins on accuracy.** The gap is almost all on score questions, where Vega reads 3 levels at 0.42, close to chance.
- **Vega edges ahead only on yes/no,** by 2 points.
- **Calibration is a draw.** Vega's main claim is honest probabilities, but on this set its calibration error is no better than Laya's.
- **The abstain flag does not help here.** Vega declined 65 items, yet its accuracy on the rest rose only from 0.633 to 0.643. On choice questions it did worse on the items it kept (0.595) than over all items (0.68), so it mostly declined items it would have got right.
- **Reading a score.** Laya's score is an expected value, which it rounds to a level. Taking each model's most probable level instead gives Laya 0.71 and Vega 0.42 on score questions.

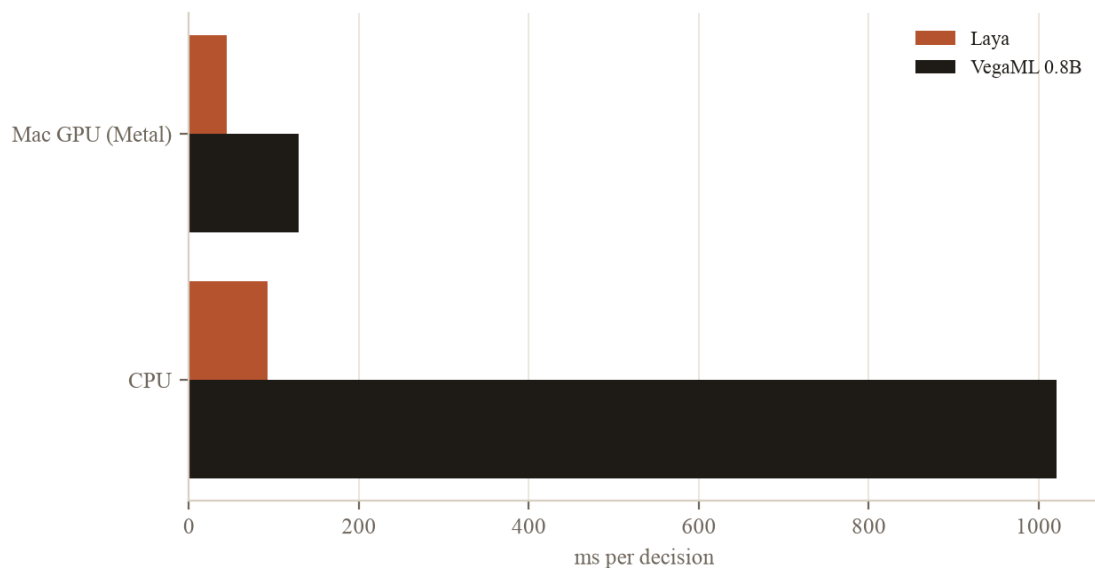


Accuracy on the sealed 300-decision set

Results: speed and memory

	Laya	Vega 0.8B
Ms per decision, Mac GPU	44	129
Ms per decision, CPU	92	1,021
Load time, warm cache	2.3 s	4.0 s
Peak memory	3.0 GB	5.1 GB

Vega warns on a Mac that two fast kernels are not installed (causal_conv1d and flash-linear-attention), and falls back to slower reference code. Its published 22 to 85 ms figures come from NVIDIA GPUs.



Milliseconds per decision (lower is faster)

Results: images (Vega only)

Laya reads text only. To test Vega's image path, 60 synthetic document pages were rendered: 15 each of invoice, receipt, form and letter, all with invented content. Each was asked "What kind of document is this?"

	Result
Right	56 of 60 (93%)
Invoices, receipts, letters	15 of 15 each
Forms	11 of 15; the other 4 were read as letters
Abstained	15
Time per image, Mac GPU	1.6 s

These pages are easy: each states its own kind in a heading. The result shows the image path works on a Mac, not that it reads hard scans.

The developer's own tasks, run on both

The developer's chart compares Vega with Jev on phishing emails and news topics, and leaves Laya's cells empty. Both models were run here on the same items: 800 emails from the public phishing-email dataset the developer used (400 phishing, 400 safe, drawn with a fixed seed), and the first 400 articles of the AG News test split. The question wording was copied from the developer's own scripts.

	Laya	Vega 0.8B
Phishing, accuracy	97.6%	98.6%
Phishing caught, of 400	392	398
False alarms, of 400 safe	11	9
News topics, accuracy	95.0%	90.3%
Calibration error, phishing and news	0.012 and 0.028	0.013 and 0.027
Ms per decision, phishing and news	99 and 43	377 and 181

- **Phishing is a narrow Vega win.** Vega got 9 emails right that Laya missed, and Laya got 1 that Vega missed. That difference is unlikely to be chance ($p = 0.02$).
- **News topics is a clear Laya win,** by 24 articles to 5 ($p < 0.001$). Vega declined 23 articles, and its accuracy on the rest was 92.6%.
- **Calibration is a tie** on both tasks.
- **A caveat favouring Laya.** Laya's own research scripts list both datasets as part of its training data, so these scores may flatter it.
- **The developer's figures did not reproduce exactly.** They report Vega at 75.4% on phishing and 95.5% on news. Their exact 800-email sample is not published, and this sample from the same dataset is evidently easier for both models.
- **Two more figures from the developer's README.** The 0.06 calibration figure in the chart comes from one task, product relevance. Over all 5,096 paired decisions they report a calibration error of 9.5% for Vega against 9.3% for Jev. They also timed Vega at 280 ms per decision on an Apple Mac.

What the developer reports, and what this test does not cover

Vega's README compares it with Jev, the hosted decision model, in one run of 3,841 decisions on NVIDIA T4s. On the README's public menu of tasks, Jev leads on accuracy at 0.836, against 0.717 for Vega 4B and 0.556 for Vega 0.8B. Vega wins on calibration and on answering every row locally. Jev was not re-run here.

This test also leaves out the 4B, which does not fit this Mac, and Vega's fitted mode, which needs labelled examples for each question.

What would change the answer

- **A machine with room for the 4B.** The developer reports it is clearly stronger than the 0.8B.
- **A task that starts from images.** Examples are sorting scanned post or photos of receipts, where Laya cannot help at all.
- **Labelled examples.** Vega's fitted mode learns from about 20 labelled examples per question, and the developer reports large gains from it.

Sources

- VegaML: NandhaKishorM/vegaml, PyPI vegaml 0.9.0
- Laya: NandhaKishorM/laya, PyPI laya 0.4.8, checkpoint convaiinnovations/laya (<https://huggingface.co/convaiinnovations/laya>)
- Software: PyTorch Metal on macOS, Python 3.14.

— Davidson