MIKALABS · TECHNICAL REPORT

Vector-L1-4B

Building a small, open teaching model —
and what we learned about why specialised models fail.

Report & model by Ahmed Zraiqat

Model: Vector-L1-4B (Light, version 1)
Report & model by: Ahmed Zraiqat
Publisher: MikaLabs
License: Apache 2.0
Released: 2026
Availability: Hugging Face · Ollama

— Abstract

> *Vector-L1-4B is a compact, open language model built to help teachers create classroom materials. This report describes the thinking behind it: why we built a specialised teaching model, a failure we encountered along the way that taught us something general about fine-tuning small models, and an honest account of what the finished model can and cannot do.*

Our central finding is simple but easy to get wrong in practice: **when you fine-tune a small model on a narrow distribution of output shapes, it does not just get good at those shapes — it loses the ability to produce others, even when explicitly instructed to.** We call this format collapse. The fix was not more data or more training; it was deliberately diversifying the *kinds* of artifacts the model was trained to produce. A second design decision addressed speed: rather than rely on a slow inference-time "thinking" phase, we moved that reasoning into the training data itself — so that Vector produces reasoned, well-structured output **directly and immediately.** This report explains these principles without publishing the underlying dataset, which remains private.

1 Motivation

Teachers spend enormous amounts of time producing materials: differentiated worksheets, mark schemes, lesson plans, quizzes, misconception guides, and explanations pitched at different abilities. General-purpose assistants can help, but they have three drawbacks for this audience: they require connectivity and per-use cost, they are not specialised for the structure of teaching artifacts, and they cannot be run privately on a school's own hardware.

We set out to build a model that was **small enough to run locally** on modest consumer hardware, **open** so that schools and educators could use it freely, and **specialised** for the particular shapes that teaching materials take. The "Light" designation reflects the first goal: Vector-L1-4B is the smallest, most portable member of a planned family.

2 The First Attempt — and How It Failed

Our first version was trained on a dataset that was heavily dominated by a single output shape: multiple-choice questions. The intention was reasonable — multiple-choice items are easy to generate at scale and easy to verify. But the result was instructive.

When we evaluated the first model on a genuinely complex teaching task — a differentiated worksheet requiring multiple difficulty tiers, a mark scheme with separated method and answer marks, a list of misconceptions, and an explicit instruction *not* to use multiple choice — the model failed in a revealing way. It produced multiple-choice questions **against the explicit instruction.** It contradicted itself within its own mark scheme. It generated filler where genuine pedagogical content was required.

> **KEY FINDING — FORMAT COLLAPSE**
>
> A small model fine-tuned on a narrow distribution of output formats does not retain its other capabilities in proportion. The dominant training shape becomes a strong prior that overrides explicit user instructions. The model had not become “better at teaching” — it had become unable to produce anything that did not resemble its training diet.

Critically, the underlying problem-solving ability was intact — the model could still do the mathematics. What it had lost was **behavioural flexibility**: the ability to follow instructions that pointed away from its training distribution. This distinction — between capability and behaviour — shaped everything that followed.

3 The Principle Behind the Fix

The instinct when a fine-tune underperforms is often to add more data or train for longer. Neither would have helped here, because the problem was not quantity — it was *distribution*. A model trained mostly on one shape will collapse toward that shape regardless of how much of it you provide.

Our second version was built on a single guiding principle:

> **DESIGN PRINCIPLE**
>
> **No single output shape should dominate the training distribution.** A specialised model needs breadth *within* its specialism — many distinct kinds of artifacts, in deliberately balanced proportions — so that instruction-following, rather than format-memorisation, is what the model learns.

In practice this meant rebalancing the data so that the previously dominant shape became a small minority, and the model saw a wide variety of teaching artifacts — worked solutions, multi-tier worksheets, mark schemes, misconception guides, lesson plans, varied question formats, and concept explanations — with no one type overwhelming the others. The model was trained to respond to *what was asked*, across many shapes, rather than to reproduce one familiar template.

The specific composition, generation process, and training configuration are part of MikaLabs' private methodology and are not published here. The transferable insight is the principle above, not the recipe.

4 Evaluation — An Honest Account

We evaluated the finished model against its own untuned starting point on a fixed set of representative teaching tasks. We report the results plainly so that users know what to expect.

Where fine-tuning clearly helped

Dimension	Result
Instruction-following	On complex, multi-constraint requests, Vector reliably followed all parts of the instruction. The untuned model frequently dropped or violated constraints, and in one case visibly broke down mid-output attempting to self-correct.
Output discipline	Vector produces the requested artifact directly. The untuned model tended toward conversational filler and preamble unsuited to a teaching tool.
Level calibration	When a specific age or ability was named, Vector pitched difficulty appropriately. The untuned model frequently under-pitched.
Artifact structure	Mark schemes, differentiated tiers, and misconception guides were markedly more usable and correctly structured.

Factual accuracy and calculation

Across our testing, Vector's factual accuracy and calculation were strong, matching or exceeding the untuned base model. The fine-tuning that improved instruction-following and structure did not come at the cost of correctness — on the contrary, Vector's disciplined, well-structured working tended to produce cleaner and more reliable results on the kinds of applied problems teachers actually set.

Efficiency — moving the thinking from inference to training

One of the clearest bottlenecks we observed was latency. The base model relies on an inference-time reasoning phase — a "thinking" pass — before it answers. On a typical teaching request this added on the order of **forty seconds of delay** before any usable output appeared, and the eventual result was still weaker than Vector's. For a tool used interactively, often on modest school hardware, a pause of that length before every response is a serious barrier to use.

This shaped a deliberate design decision. Rather than have the model reason slowly at the moment of use, we moved that reasoning **upstream, into the construction of the training data.** The examples Vector learned from were not terse answers; they were fully reasoned, worked-through artifacts — the kind of output a model would normally arrive at only *after* an extended thinking phase. The careful reasoning was performed once, in advance, and captured in the data itself.

DESIGN PRINCIPLE

Pre-compute the thinking, so the model doesn’t have to. By training on data that already embodies the reasoning a thinking phase would produce, Vector learns to generate well-structured, reasoned output *directly*. The thinking still happens — but at training time, not on every request. In effect, we did the thinking ahead of time, once, so that every future answer is immediate.

The result is a model that is both faster and more useful in its setting: it answers without delay, and answers well. Better output produced immediately is worth far more to a teacher than a slowly-reasoned answer that takes the better part of a minute to arrive.

A NOTE ON SCALE

Vector-L1-4B is a compact, four-billion-parameter model. A model of this size carries inherent technical limitations, and like any AI tool its output is best treated as a strong draft for a qualified educator to review — particularly answer keys intended for students. This is the appropriate and standard safeguard, not a reflection of weakness in the model.

5 Responsible Use

Vector is designed for school and secondary-level teaching and is intended as an assistant to a qualified educator, not a replacement for one. Because the model can produce confident output on problems beyond its reliable reach, we recommend a simple discipline: **a teacher should review answer keys and factual content before using them with students.** This is standard practice for any AI tool, and it is the appropriate safeguard given the model’s scale.

6 Roadmap

Vector-L1-4B is the first release, not the destination. The dataset and approach that produced it are model-agnostic, which makes both scaling and specialisation a matter of applying the same foundation in new directions. Several efforts are underway.

- **Continued data refinement.** Ongoing improvements to the training data, carrying the same diversity-and-pre-computed-reasoning principles forward into every future model.
- **Vector Studio.** A desktop application bringing Vector to teachers through a simple interface — with built-in file and quiz generation, so educators can produce and export classroom materials directly, without touching a command line.

Future models

The Vector family will extend beyond the Light model in three directions:

- **Vector-1-14B** — the flagship. A larger model for higher accuracy and stronger performance on demanding material, built on the same teaching foundation.
- **Vector-R1** — a reasoning model, for tasks where transparent, step-by-step working is the goal rather than immediacy.
- **Vector-V1** — a vision model, extending Vector to diagrams, figures, and image-based teaching material.

7 Conclusion

The most valuable thing we learned building Vector-L1-4B was not about teaching — it was about fine-tuning. A small model is shaped as much by the *variety* of what it is trained on as by the volume. Narrow it too far and it loses the very flexibility that makes it useful, even on tasks it is otherwise capable of. Breadth within a specialism is what lets a small model stay obedient to instructions rather than collapsing toward a single learned template.

The result is a compact, open, locally-runnable model that does something genuinely useful for educators — and a principle we will carry into every model in the Vector family.
