Comparative Grading Report:
Qwen2.5-VL-3B-Instruct vs. MoE Router

1 Introduction

This report compares two language models on a small but deliberately challenging benchmark of 50 PhD–level questions in advanced mathematics and theoretical/computational computer science:

- 25 questions in real analysis, functional analysis, PDEs, probability, topology, approximation theory and numerical analysis.
- 25 questions in algorithms, data structures, systems, parallel computing, statistics in R, and numerical/scientific computing.

The two models are:

Model A: Qwen2.5-VL-3B-Instruct, a 3B generalist instruction-tuned model.

Model B: MoE Router, a hand-built routed system that selects between three smaller specialists:

- `Qwen2.5-0.5B-Instruct` (general instruction expert),
- `Qwen2.5-Coder-0.5B-Instruct` (code-oriented expert),
- `Qwen2.5-Math-1.5B-Instruct` (math-centric expert).

For each question, the router chooses which expert to use (and in which order) based on the task type and internal heuristics.

The goal of this experiment is not to show that the MoE router *beats* the much larger Qwen2.5-VL-3B-Instruct, but to assess whether it can come *close* on very difficult tasks while using smaller, specialized components.

2 Grading Methodology

Each question was answered by both Model A and Model B. Answers were graded manually and independently along three axes:

1. **Solve** (0–10): Did the model substantially solve the main mathematical or algorithmic task?
2. **Explain** (0–10): Was the reasoning clear, coherent and logically structured?
3. **Code** (0–10): When the question explicitly requested code, did the model produce relevant, plausible code that implements the intended method or algorithm?

The same rubric was used for both models:

- 0–2: off-topic, nonsense, or essentially no progress.

- 3–4: touches the right area but with major gaps or broken logic; code skeletons that miss the core algorithm.

- 5–6: captures important ideas and a significant fraction of the solution, but with nontrivial gaps or errors; code that is structurally correct but incomplete or likely buggy.

- 7–8: overall correct approach with most of the reasoning or algorithm in place; a knowledgeable person could fix it without rewriting from scratch.

- 9–10: essentially correct and complete; near-expert quality reasoning and implementation (extremely rare on this benchmark).

The benchmark used in this evaluation is intentionally severe: all fifty questions are formulated at upper-undergraduate, graduate, or PhD-qualifying-exam level, covering areas such as real and functional analysis, PDE theory, measure theory, stochastic processes, topology, numerical analysis, advanced algorithms, distributed systems, high-performance computing, compiler design, and statistical modeling in R. These problems require not only factual knowledge but also the ability to perform multi-step mathematical reasoning, construct proofs, analyze system behaviors, derive or approximate solutions, and produce executable code aligned with theoretical requirements. Such questions are far beyond the difficulty of standard LLM benchmarks, making this an extremely demanding test of reasoning depth, mathematical maturity, and cross-domain technical competence.

3 Per-Question Scores

Table 1 reports the per-question scores. “Solve”, “Explain” and “Code” are given on a 0–10 scale for both models.

Table 1: Per-question scores for Qwen2.5-VL-3B-Instruct (Model A) and MoE Router (Model B).

ID	Topic	Qwen2.5-VL-3B			MoE Router		
		Solve	Explain	Code	Solve	Explain	Code
1	Real Analysis	3	3	2	1	1	2
2	Measure & Integration	2	2	2	2	2	3
3	Functional Analysis	3	3	2	5	5	5
4	Real Analysis / BV & AC	6	6	5	5	5	3
5	PDE / Sturm–Liouville	5	5	4	6	6	6
6	Probability / Martingales	5	5	4	5	5	4
7	Probability / Limit Theorems	3	3	2	6	6	4
8	Complex Analysis / Normal Families	5	5	4	5	5	4
9	Complex Analysis / Argument Principle	5	5	4	4	4	4
10	Linear Algebra / Jordan & $\exp(A)$	7	7	6	6	6	5
11	Multivar Calc / Implicit Fn Thm	6	6	5	5	5	4
12	Vector Calc / Differential Forms	6	6	5	6	6	4
13	ODE / Blow-up & Global Existence	6	6	5	6	6	4
14	Real Analysis / Uniform Conv & Diff	6	6	5	6	6	4
15	Approx Theory / Stone–Weierstrass	6	6	5	6	6	4
16	Fourier Analysis	6	6	5	6	6	3
17	Real Analysis / Cantor Function	5	5	4	5	5	3
18	Topology & Baire Category	6	6	5	6	6	4
19	Probability / Coupling	5	5	4	5	5	4
20	PDE / Energy Methods	5	5	4	4	4	4
21	Linear Algebra / SVD & PCA	6	6	5	6	6	3
22	Convex Optimization	6	6	5	6	6	3
23	Multivar Calc / Change of Variables	5	5	4	5	5	3

ID	Topic	Qwen2.5-VL-3B			MoE Router		
		Solve	Explain	Code	Solve	Explain	Code
24	Real Analysis / Function Spaces	5	5	4	4	4	3
25	Numerical Analysis / Stability	6	6	5	5	5	4
26	Algorithms (C++)	7	7	6	6	6	5
27	Data Structures (C++)	6	6	5	6	6	4
28	Graph Algorithms (C++)	6	6	5	6	6	4
29	Concurrency (C++)	7	7	6	6	6	4
30	Numerical Linear Algebra (C++)	7	7	6	6	6	4
31	Distributed Systems (Java)	6	6	5	6	6	4
32	Concurrent Programming (Java)	6	6	5	6	6	4
33	Garbage Collection (Java)	6	6	5	5	5	3
34	Statistical Computing (R)	7	7	6	6	6	5
35	Time Series (R)	5	5	4	4	4	4
36	High-Performance Computing (C++)	6	6	5	5	5	4
37	Compilers (C++)	7	7	6	5	5	4
38	Numerical Optimization (C++/R)	6	6	5	4	4	3
39	Big Data (R)	5	5	4	6	6	4
40	Advanced Data Structures (C++)	7	7	6	5	5	3
41	Networking (C++)	6	6	5	5	5	3
42	Parallel Programming (C++/OpenMP)	6	6	5	5	5	3
43	Object-Oriented Design (Java)	6	6	5	6	6	4
44	ML Systems (Java)	6	6	5	5	5	3
45	Statistical Simulation (R)	6	6	5	5	5	3
46	Numerical PDEs (C++)	6	6	5	5	5	4
47	Advanced R Programming	6	6	5	4	4	3
48	Formal Verification (C++)	5	5	4	5	5	3
49	Multi-language Integration (C++/R)	5	5	4	5	5	3
50	Databases (Java)	6	6	5	4	4	3

4 Average Scores and Model Analysis

4.1 Average Scores

Averaging over all 50 questions, we obtain:

Model	Solve (avg)	Explain (avg)	Code (avg)
Qwen2.5-VL-3B-Instruct	5.62	5.62	4.64
MoE Router	5.14	5.14	3.70

Several observations:

- On **Solve** and **Explain**, the MoE model is only about 0.5 points behind Qwen on a 0–10 scale (roughly a 10% gap), despite relying on smaller experts and a custom router.
- On **Code**, Qwen retains a noticeable advantage: 4.64 vs. 3.70 on average. Qwen's code is generally more consistent and better aligned with the task, especially on mathematically heavy questions.

4.2 Model A: Qwen2.5-VL-3B-Instruct

Strengths.

- Strong and stable performance across both math and CS questions.
- Tends to produce detailed, step-by-step explanations for difficult proofs and derivations.
- Code generations are usually relevant, reasonably structured and often close to a working implementation.

Weaknesses.

- On truly PhD-level problems, solutions frequently plateau in the 5–7 range: the model outlines the right ideas but often fails to fully close the argument.
- Some answers are verbose and occasionally contain redundant or slightly hand-wavy reasoning.

4.3 Model B: MoE Router

Architecture. Model B uses a routing mechanism to dispatch queries among three specialists:

- `Qwen2.5-0.5B-Instruct` for general reasoning and non-technical instructions.
- `Qwen2.5-Coder-0.5B-Instruct` for programming tasks in Python, C++, Java and R.
- `Qwen2.5-Math-1.5B-Instruct` for mathematically intensive questions.

The router attempts to identify whether a query is primarily "solve", "explain" or "code" oriented and selects (or sequences) experts accordingly.

Strengths.

- Despite using smaller specialists, the MoE achieves Solve and Explain averages only ≈ 0.5 points below Qwen, indicating that the routing strategy is effective on many tasks.
- On some advanced mathematical questions, the math expert can match or even outperform Qwen in conceptual clarity or choice of theorems.
- Code is present for all 50 questions; many snippets are structurally appropriate (correct language, imports, data structures), especially in R and C++ systems questions.

Weaknesses.

- Code quality is less consistent than Qwen: implementations are often partial, truncated or lacking crucial details, which suppresses the average Code score.
- Routing errors occasionally send mathematically heavy questions to the wrong expert or fail to fully leverage the math specialist.
- Explanations, while competitive in many cases, can be less stable in depth and organization compared to the single large model.

5 Conclusion

On this deliberately difficult benchmark of 50 PhD-level questions, the MoE Router model does *not* surpass the larger Qwen2.5-VL-3B-Instruct model, but it does come *surprisingly close* in several important dimensions:

- The average **Solve** and **Explain** scores differ by only about 0.5 points on a 0–10 scale, corresponding to a gap of roughly 10%.
- The **Code** gap is more pronounced but still moderate (4.64 vs. 3.70), and largely attributable to incomplete or underdeveloped implementations rather than total failure to produce code.

Given that Model B relies on smaller experts and a still-developing routing strategy, these results indicate that the MoE architecture is already operating in a performance regime *close* to a strong 3B generalist on very challenging tasks. With improved routing, better training of the code expert, and more deliberate tuning on mathematical workloads, the MoE system has clear headroom to narrow the gap further.

In summary, Qwen2.5-VL-3B-Instruct remains ahead overall, but the MoE Router demonstrates that carefully combined small specialists can approach the performance of a larger monolithic model on demanding math and CS questions.