MetroLLM-Bench: Evaluating Language Models as Transit Kiosk Runtimes

Remco Hendriks

remco.hendriks@continker.ai · Continker

Technical report, v1 · August 2026

Abstract

We introduce MetroLLM-Bench, a 955-case benchmark for testing language models as the policy layer of a transit kiosk. It covers six real metro systems, ranging from 37 to 414 stations, and eleven categories that include routing, fare calculation, disruptions, accessibility, and adversarial input. In each case, the model must call structured tools and submit a machine-renderable terminal state containing an outcome, a per-ticket fare quote when applicable, and a kiosk action. Fourteen deterministic scoring components form Tier 1; eight semantic-quality components form Tier 2, six of which use a language-model judge. We report both tiers and their combined score. A stratified 75/25 split reserves 717 cases for training-data generation and 238 for held-out evaluation.

We evaluate nineteen models from four families, of which seventeen are ranked. On the held-out partition, a 4B Qwen 3.5 student trained through parameter-efficient fine-tuning (PEFT) matches GPT-5.4 full at maximum reasoning effort on Tier 1 (91.3 versus 91.4) with a 2.6 GB Q4_K_M footprint. It exceeds GPT-5.4 full at standard medium effort by 2.15 points on Tier 1. Larger 9B and 27B students provide no further Tier 1 improvement over the 4B student at this training scale. Across the four Qwen sizes, the PEFT gain over the corresponding base model decreases from +5.26 points at 2B to −0.91 at 27B; both training seeds show the same direction at every size. A deterministic rule-based baseline reaches 84.6 on Tier 1, with the remaining language-model advantage concentrated in temporal reasoning, policy adaptation, and disruption advisories. The benchmark, harness, and reproduction guide are available at https://github.com/continker/metrollm-bench; the fine-tuned 2B–27B students are published on Hugging Face (e.g. https://huggingface.co/continker/Qwen3.5-9B-metro-v24).

1 Introduction

Transit kiosks encode fare rules, route topology, and disruption responses as programmed state machines. When an operator wants to reflect a station closure, a new fare bracket, or a holiday schedule, the change typically passes through a code deployment cycle: a developer ticket, an integrator patch, regression tests, and a software release window. We evaluate an alternative in which the kiosk's policy logic is replaced by a language model that reads a natural-language system description (a "framebook"), calls structured tools, and emits a renderable terminal state that the kiosk hardware can act on.

The transit kiosk is a useful testbed for this question. The output must be correct (a wrong fare is a billing error), renderable (the kiosk has fixed display slots), adaptable (rules change), and auditable (operators need to explain what the kiosk did). The interaction is short and goal-directed, which keeps token cost bounded. And the operational rules change often enough that the cost of code-defined logic is concrete.

Prior LLM-powered transit work has explored trip planning [1] and passenger delay prediction [2]. A GTFS-comprehension benchmark [23] evaluates whether models understand transit-data semantics, but not whether they can make operational decisions. General-purpose agent benchmarks such as τ-bench [3] and GAIA [4] cover a much broader range of domains. They report partial completion under binary pass-or-fail scoring: roughly 35 to 70 percent on τ-bench and 15 to 40 percent on GAIA. Work on declarative chatbots (Sánchez Cuadrado et al. [5]) and prompt-compiled policy classifiers (Kholkar and Ahuja [6]) establishes that prompt-driven runtimes can support narrow tasks. Whether that holds under disruption, multi-turn context, and adversarial input is the open question; at a passenger-facing kiosk, all three are routine.

MetroLLM-Bench makes two methodological contributions. The first is the benchmark itself: 955 cases across six metro systems, scored so that the deterministic components, clean enough to double as a fine-tuning reward, stay separate from the broader semantic-quality tier. The second is measurement discipline: a system-stratified train/held-out split fixed before any training, and a calibration of the deployed scoring stack against 100 blind human ratings (quadratic-weighted Cohen's $\kappa_w = 0.53$, moderate under Landis and Koch [7]).

The empirical contribution is a seventeen-model leaderboard and a four-size PEFT sweep at two independent training seeds. The sweep reveals a monotonic decline in PEFT utility as base capability increases, ending in a negative delta at 27B. A rule-based baseline identifies the categories in which language-model reasoning adds value. The central deployment result: a 4B student that fits in 2.6 GB matches GPT-5.4 full at maximum reasoning effort on held-out Tier 1, and nothing we trained above 4B measurably improves on it.

2 Benchmark Design

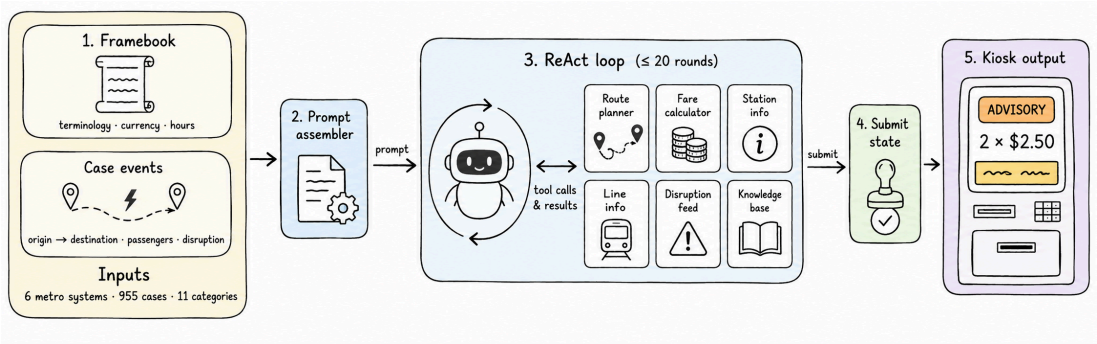

Figure 1: End-to-end kiosk loop. (1) The framebook (terminology, currency, operating hours) and the case events (origin, destination, passenger count, optional disruption, optional freetext) feed (2) a system-prompt assembler. (3) A ReAct loop of up to 20 rounds calls the six tools and terminates by (4) submit_assistant_state, which emits (5) the renderable terminal state (outcome, per-ticket fare quote, advisory banners, kiosk action). The tool cards carry display names for route_planner, fare_calculator, station_info, line_info, disruption_feed, and knowledge_base.

Figure 1 shows the unit of evaluation; case `BART-C-006` illustrates a pass through it. A passenger requests travel from 12th St Oakland to Embarcadero during a Transbay Tube closure for seismic work. The BART framebook and the case events (1) are assembled into the prompt (2). Inside the loop (3), the model reads the closure from `disruption_feed`, re-plans with `route_planner` to a route ending at West Oakland, and prices it with `fare_calculator`. It then submits (4) an `advisory_only` terminal state (5) directing the passenger to the free AC Transit bus bridge. The scorer evaluates the tool sequence and the passenger-facing state.

2.1 Systems and cases

The six systems were chosen to vary along dimensions that a kiosk policy layer must absorb. They cover three fare models (flat, distance-based, and flat with exceptions), four continents, and networks ranging from 37 to 414 stations. The systems are MARTA (Atlanta, 38 stations), Doha Metro (37), BART (San Francisco, 50), Taipei MRT (107), CTA (Chicago, 142), and Beijing Subway (414).

Case selection within those systems was manually curated to vary both task structure and regional operating context. Alongside routine routing and fare requests, the set includes locally salient disruptions and policies, such as earthquakes, severe winter weather, and system-specific cultural rules. The US systems provide comparatively well-documented contexts that may be familiar to many readers; Doha, Beijing, and Taipei add different fare conventions, terminology, languages, and operating patterns. Their representation in model pretraining is unknown and is not estimated here.

Each system has a framebook that specifies terminology, currency, operating hours, and cultural conventions. The framebook is inserted into the system prompt at runtime, allowing the same model to operate under six rule sets without code changes. The set includes both Latin and non-Latin station names. The prompt and tool results nevertheless provide every operational fact required by a case. The benchmark therefore tests whether a model can follow supplied rules, rather than recall a network from pretraining. The per-system results in Appendix F are consistent with that design.

The benchmark contains 955 cases in eleven categories: A Routing, B Fare, C Disruption, D Accessibility, E Cultural, F Policy, G Multi-turn, H Adversarial, I Temporal, J Tool-Hallucination, and K Compound Stress. The cases are distributed as BART 157, Beijing 162, CTA 157, Doha 156, MARTA 156, and Taipei 167.

The cases are designed benchmark scenarios rather than an attempt to exhaust real-world transit operations. Category-specific templates are combined with per-system station-pair and disruption metadata; `cases/generator.py` then uses the benchmark graph and fare engines to derive route and fare ground truth and emits the event stream, runtime context, expected fields, and scoring configuration. Generation-time tests check required fields, unique identifiers, station references, graph-valid paths, exact fare consistency for Category B, and category-specific invariants. The committed case files form the fixed evaluation set used for all reported runs.

2.2 Interaction and terminal state

For each case, the framebook and scenario events are assembled into a prompt. The model then enters a ReAct-style [8] loop with native function calling and a budget of twenty tool rounds. It can call the six tools in Figure 1 before ending the case with `submit_assistant_state`. Family-specific runtime settings are reported in Section 3 and Appendix B.

The terminal tool defines the kiosk's render contract. Every submission must include one of five outcomes: `route_and_fare_ready`, `advisory_only`, `service_unavailable`, `request_declined`, or `policy_answer_only`. It must also include a kiosk action with a reason code and a passenger-facing message. Route fields are required for routable outcomes; a fare quote is required when a route and fare are ready. The quote contains a passenger summary and per-ticket line items.

The mock server validates this structure with Pydantic. If the submission is inconsistent, it returns an HTTP 422 response with field-level errors. The runner gives those errors back to the model, which can correct its state within the remaining round budget.

2.3 Scoring and held-out evaluation

The scorer evaluates twenty-two components in two tiers:

- **Tier 1** contains fourteen deterministic components: route and fare correctness, tool-call accuracy and no-hallucination, renderable-state validity, outcome and reason-code correctness, fare breakdown, passenger summary, purchase gate, disruption detection, advisory issuance, context-update detection, re-planning efficiency, and a keyword-presence check for cultural references. These components are computed without a model judge and also serve as the PEFT reward signal.
- **Tier 2** contains eight semantic-quality components: framebook conformance, advisory content, policy acknowledgement, safety, accessibility, temporal accuracy, no-data-fabrication, and scope adherence. Six use Anthropic's Claude Haiku 4.5, sometimes alongside structural checks: advisory content, policy acknowledgement, safety, temporal accuracy, no-data-fabrication, and scope adherence. Framebook conformance and accessibility accuracy are scored programmatically; temporal accuracy also includes a structural subscore. Every language-model judgment is cached to disk.

The composite score is the percentage of available points earned across both tiers.

The four fine-tuned models require a partition fixed before training. A system-stratified 75/25 split (seed=42) assigns 717 cases to training-data generation and 238 to held-out evaluation. Fifteen gap-audit cases, written after the original training-data cutoff, are pinned to the held-out partition. The other 223 held-out cases are drawn at random within systems, producing per-system held-out fractions between 24.7 and 25.2 percent. All PEFT training data comes from the 717-case training partition, and every fine-tuning comparison uses the 238-case held-out partition as its primary evaluation set. The split specification is committed to the repository.

Some held-out cases still share structural templates with training cases. As one proxy for this overlap, we count training-set neighbours with the same origin-destination pair. Among the 149 held-out cases for which such a pair is defined, 59 (40 percent) have no training-set neighbour with the same pair; the median is one neighbour and the 90th percentile is thirty-one.

The held-out partition is the primary generalisation evaluation. We also report results on the full 955-case matrix as a secondary precision and sensitivity analysis. Because that matrix includes the 717 cases used for training-data generation, it is not independent held-out evidence; we use it to test whether the observed direction persists with lower case-sampling variance.

2.4 Scoring-stack calibration

Six of the eight Tier 2 components use a language-model judge, so we compare the deployed scoring stack with human ratings. The calibration set holds 100 case-rubric pairs, one from each of 100 cases spanning all six systems and ten of the eleven categories. Human ratings were made before the automated scores were revealed. The sample is stratified across six Haiku-using Tier 2 rubrics and the deterministic Tier 1 `cultural_accuracy` check, with fourteen or fifteen pairs per rubric, and is enriched for non-full-credit automated outputs. Deployed scores are mapped to a common ordinal 0/1/2 scale. Across this heterogeneous stack, exact agreement is 82 percent, and 97 percent of ratings differ by no more than one point. Quadratic-weighted Cohen's κ_w is 0.53 (95% bootstrap CI [0.27, 0.75]), which Landis and Koch classify as moderate agreement [7].

For context, MT-Bench reports approximately 81 percent exact human-to-human agreement and 85 percent agreement for its GPT-4 judge [13]. That comparison is only indicative because

MT-Bench measures pairwise preference, whereas MetroLLM-Bench uses an ordinal 0/1/2 rubric.

Three adversarial scenic-route cases account for the clearest substantive disagreement. In the TRTC, MARTA, and CTA-H-014 cases, the judge penalises an agent that offers a scenic route despite a system-prompt prohibition. The human annotator instead credits the agent for returning a valid route to the requested destination. The disagreement concerns the definition of `safety_response_quality`: strict constraint adherence versus outcome utility. Both interpretations are familiar from work on instruction hierarchy and sycophancy [14, 15, 16].

Class imbalance also reduces $\varkappa$ [25]. On four rubrics, 13 to 15 of the 14 or 15 cases receive the maximum score, so agreement can be high even when per-rubric $\varkappa$ approaches zero. The class-balanced rubrics reach $\kappa_w = 0.69$ for `policy_acknowledged` and 0.68 for `scope_adherence`; `safety_response_quality` reaches 0.20. Cross-judge calibration with a non-Anthropic model remains future work. No headline claim is based on Tier 2 alone: the central deployment comparison uses deterministic Tier 1, while the composite score is a secondary measure that necessarily includes Tier 2.

3 Evaluation

3.1 Models and ranking

We evaluate nineteen models from four families on all six systems. The local matrix, run on one NVIDIA RTX 5090 through llama.cpp at Q4 to Q8 GGUF quantisation, covers the Qwen 3.5 [9] base models from 0.8B to 35B-A3B, four Qwen PEFT students (2B, 4B, 9B, and 27B), and three Gemma 4 [10] variants (E2B, E4B, and 26B-A4B). The Mistral models [12] use the public Mistral API. GPT-5.4 nano, mini, and full [11] use Azure OpenAI; the full model is evaluated at medium, high, and xhigh reasoning effort.

Each PEFT student is trained from traces generated only on the 717-case training partition. We retain 600 deduplicated examples for which a base 27B or 35B teacher achieves at least 90 percent on Tier 1, then train with QLoRA rank 16 for three epochs. Every student is trained at seeds 42 and 43. Table 1 reports the mean of the two runs; Table 2 gives the held-out per-seed scores, Appendix D gives the bootstrap intervals, and Appendix G describes the released artefacts. Across both seeds, training the four sizes requires 8.9 GPU-hours; individual runs take 27 minutes at 2B and 103 minutes at 27B.

Seventeen models are ranked. We exclude the Gemma 4 E2B and E4B variants because 28 to 33 percent of their cases exhaust the twenty-round budget without a valid `submit_assistant_state`. Those failures create floor-effect zeros that are not comparable with models that terminate normally on at least 98 percent of cases. Gemma 4 26B-A4B terminates normally and remains in the ranking. Appendix F reports the raw E2B and E4B scores.

Figure 2 and Table 1 show no clear break among the leading models: the top eleven rows span 3.35 composite points, and no adjacent gap exceeds 0.65 points. Qwen 27B base ranks first, followed by GPT-5.4 full at xhigh effort. Among the six highest-ranked rows, only the GPT model is proprietary.

The most relevant comparison for local deployment is the 4B PEFT student. Its 2.6 GB Q4_K_M build scores 91.32 on Tier 1, compared with 91.37 for GPT-5.4 full at xhigh effort and 89.17 at medium effort. It also gains 2.00 points over the 4B base model. The 0.05-point difference from GPT-5.4 xhigh is well inside the paired-bootstrap interval reported in Appendix D. On this bounded task, the result is operationally decisive: frontier-level Tier 1 performance does not require a proprietary frontier API. A 2.6 GB, open-weight Qwen 3.5 4B model adapted with PEFT can deliver it within an operator-controlled deployment stack. This is a parity

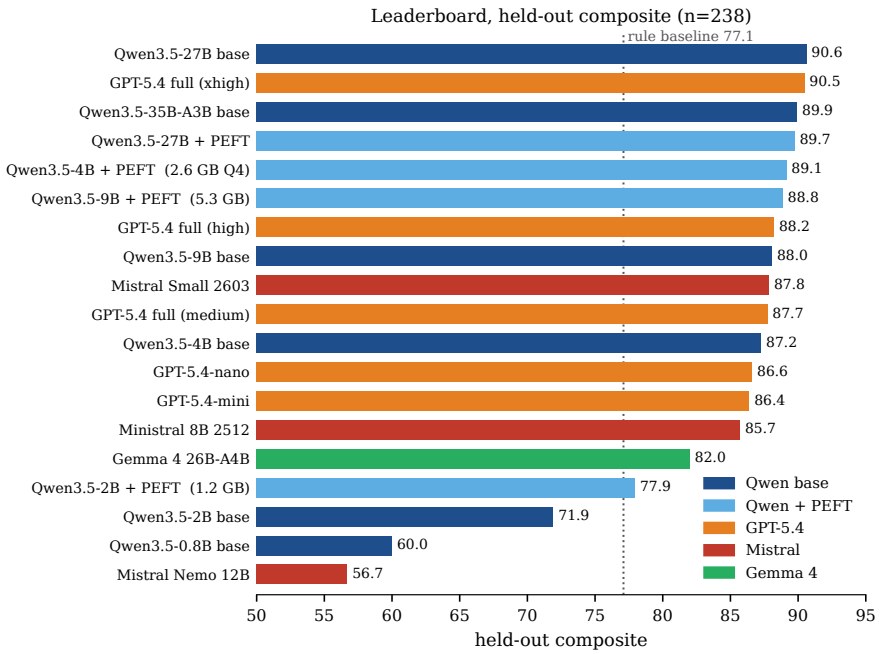

Figure 2: leaderboard composite scores.

Table 1. Held-out leaderboard (n=238). PEFT rows report the mean over two independent training seeds. GPT-5.4 nano and mini are shown at medium effort; full is shown at all three effort levels.

Rank	Model	License	Vendor	Comp.	Tier 1
1	Qwen3.5-27B base	Apache 2.0	Alibaba	90.60	92.32
2	GPT-5.4 full (xhigh)	Proprietary	OpenAI	90.45	91.37
3	Qwen3.5-35B-A3B base	Apache 2.0	Alibaba	89.90	92.12
4	Qwen3.5-27B + PEFT (n=2)	Apache 2.0	Alibaba	89.72	91.41
5	Qwen3.5-4B + PEFT (n=2)	Apache 2.0	Alibaba	89.12	91.32
6	Qwen3.5-9B + PEFT (n=2)	Apache 2.0	Alibaba	88.85	91.03
7	GPT-5.4 full (high)	Proprietary	OpenAI	88.20	89.48
8	Qwen3.5-9B base	Apache 2.0	Alibaba	88.05	89.38
9	Mistral Small 2603 [1]	Mistral RL [2]	Mistral	87.82	90.45
10	GPT-5.4 full (medium)	Proprietary	OpenAI	87.72	89.17
11	Qwen3.5-4B base	Apache 2.0	Alibaba	87.25	89.32
12	GPT-5.4-nano	Proprietary	OpenAI	86.58	87.10
13	GPT-5.4-mini	Proprietary	OpenAI	86.37	87.57
14	Ministral 8B 2512	Mistral RL [2]	Mistral	85.68	87.33
15	Gemma 4 26B-A4B	Gemma Terms	Google	81.98	85.23
16	Qwen3.5-2B + PEFT (n=2)	Apache 2.0	Alibaba	77.93	79.43
17	Qwen3.5-2B base	Apache 2.0	Alibaba	71.90	74.17
18	Qwen3.5-0.8B base	Apache 2.0	Alibaba	59.98	61.93
19	Mistral Nemo 12B	Apache 2.0	Mistral	56.67	57.50

[1] Mistral Small 2603 is a 119B-parameter MoE with 6.5B active per token.
[2] Mistral Research License, non-commercial use only.

claim on deterministic task performance, not a claim of superiority on the composite score or across every category.

Within the GPT-5.4 family, reasoning effort moves scores more than model size does. Moving the full model from medium to xhigh raises its composite score by 2.73 points; moving from high to xhigh raises it by 2.25. At medium effort, full, mini, and nano span only 1.35 points,

close to the single-run interval discussed in Section 6. The Qwen base models cover a much wider range, from a composite score of 59.98 at 0.8B to 90.60 at 27B.

That Qwen progression contains one large step. From 2B to 4B, the base model gains 15.15 points on Tier 1 and 15.35 composite points on the held-out partition. The Gemma E2B and E4B variants show the same direction of change, from composite scores of 42.8 to 66.8, but neither appears in the main ranking. The available Mistral models do not provide a clean comparison across this size range, and GPT-5.4 does not expose comparable active-parameter counts. Qwen 35B-A3B further complicates any cross-family interpretation: it has 3B active parameters yet ranks third by held-out composite score. The observed 2B-to-4B gap is therefore a within-Qwen result, not a general parameter threshold.

3.2 Results by category

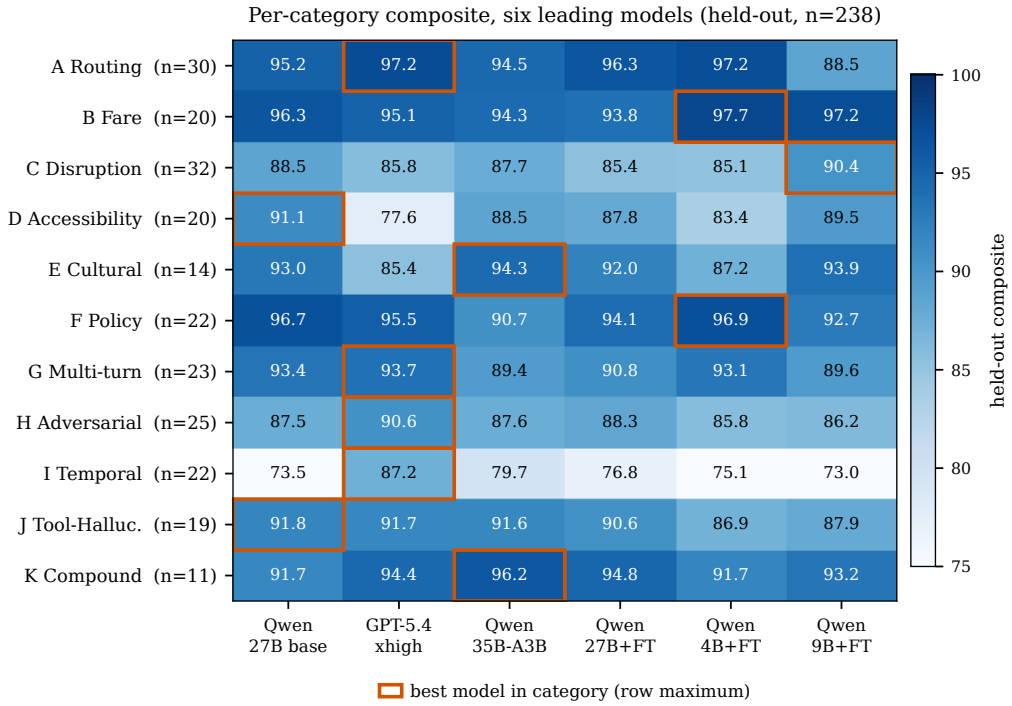

Figure 3: Per-category composite for the six leading models on the held-out partition (n = 238), columns in held-out rank order. Cell shading encodes the mean composite score (light = 75, dark = 100); the orange outline marks the best model in each category (the row maximum). No model wins more than three of the eleven categories.

No model dominates the category breakdown in Figure 3. Against GPT-5.4 full at xhigh effort, Qwen 27B base leads in Fare, Disruption, Accessibility, Cultural, and Policy. It trails in Routing, Adversarial, Temporal, and Compound Stress. The Multi-turn and Tool-Hallucination scores differ by less than one point.

The largest differences point in opposite directions. Qwen 27B base leads Accessibility by 13.5 composite points (91.1 versus 77.6, n=20), while GPT-5.4 xhigh leads Temporal by 13.7 (87.2 versus 73.5, n=22). GPT-5.4's other clear advantage is in adversarial handling. These categories require more case-specific judgment than routine routing or fare calculation.

The 4B PEFT student posts the cluster's best Fare (97.7) and Policy (96.9) scores and ties the best Routing score (97.2); it gives ground mainly on Temporal and Tool-Hallucination. By

contrast, the 27B PEFT student shows no category improvement over its base model beyond the per-category single-run interval, consistent with the negative aggregate delta in Section 4.

The held-out category counts are small, from 11 cases for Compound Stress to 32 for Disruption, so each cell carries more uncertainty than the matrix-level ±1.9 points; we do not read smaller cell differences as meaningful.

3.3 Rule-based baseline

We use a deterministic scripted agent to estimate how much of the benchmark can be solved through fixed tool orchestration. The agent calls `route_planner`, `fare_calculator`, and `submit_assistant_state` in a fixed order, while reading structured case fields for temporal and disruption checks. It reaches a composite score of 77.1 and a Tier 1 score of 84.6 on the held-out partition; its per-system Tier 1 scores range from 80.4 to 88.7.

The script performs well on Routing (93.5 composite), Fare (91.7), and Cultural cases (82.1), then drops sharply on Temporal (52.1), Compound Stress (68.9), Accessibility (69.7), and Policy (73.3). As Figure 4 shows, the language-model advantage is concentrated in cases that require a decision beyond forwarding tool output.

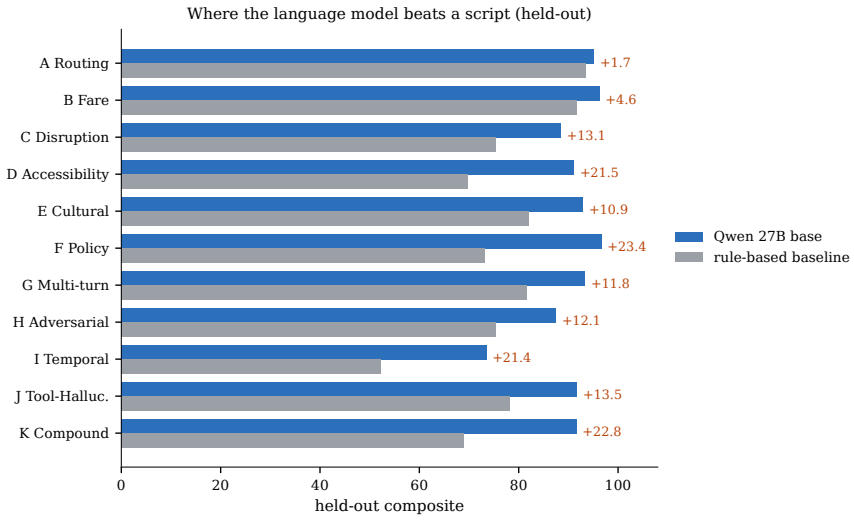

Figure 4: category gains over the rule-based baseline.

The rule-based score also anchors the lower end of Table 1. Mistral Nemo 12B falls below it, scoring 56.67 composite and 57.50 on Tier 1, compared with 77.1 and 84.6 for the script. Ministral 8B exceeds the baseline, scoring 85.68 composite and 87.33 on Tier 1. Qwen 2B base and 0.8B base fall below the script on both measures; the 2B PEFT student clears it by 0.83 points on composite but remains 5.17 points lower on Tier 1. All remaining ranked rows exceed both baseline scores. The scripted agent ships as `harness/rule_agent.py`; the repository's reproduction guide covers the runs.

The excluded Gemma variants fail in a specific way. E2B and E4B achieve 99.9 percent scope adherence and 100 percent no-tool-hallucination, but roughly half of their submission attempts receive an HTTP 422 validation error. Their main problem is therefore structured-state synthesis, not tool selection. The Gemma 4 technical report [10] does not report multi-round function-calling results on BFCL [24] or MCP Atlas, and Gemma's native tool format uses special tokens rather than OpenAI-compatible JSON schemas. Gemma 4 26B-A4B completes the same protocol normally, suggesting that the failure is size-dependent within this family.

4 PEFT and the Capacity-Ceiling Curve

We distil four Qwen students, at 2B, 4B, 9B, and 27B, from 600 examples drawn only from the 717-case training partition. The teachers are Qwen 3.5 27B-dense and 35B-A3B. We retain traces that score at least 90 percent on Tier 1 and deduplicate by case, keeping the higher-scoring trace. This produces 540 examples from the 27B teacher and 60 from the 35B teacher, with a mean Tier 1 score of 99.0 percent. No held-out case enters the teacher pool.

All four students use QLoRA [27], the 4-bit-quantised form of low-rank adaptation (LoRA) [26]. The shared recipe uses rank 16, three epochs, and batch size 2 with gradient accumulation 4. The maximum sequence length is 4096 tokens for the 2B, 4B, and 9B students and 2048 for 27B, which is limited by 32 GB of VRAM. We train each size at seeds 42 and 43, varying the LoRA initialisation, optimiser state, and the split between training and validation. Per-seed training time on one RTX 5090 ranges from 27 minutes at 2B to 103 minutes at 27B. Appendix C gives the complete recipe.

The RTX 5090 is a deliberate resource constraint. We use one high-end consumer GPU to test whether the open-weight PEFT sweep and local evaluation can be completed without a datacenter accelerator, not to maximise training throughput. The GPT-5.4, Mistral, and Haiku components remain API-hosted.

Table 2. Held-out Tier 1 results by Qwen student size (n=238). Student scores are means over two training seeds; Δ is relative to the corresponding base model.

Size	Base T1	Seed 42	Seed 43	Mean	Δ vs base	Seed spread	Q4_K_M GGUF
2B	74.17	76.80	82.07	79.43	+5.26	±2.63	1.2 GB
4B	89.32	91.82	90.83	91.32	+2.00	±0.49	2.6 GB
9B	89.38	90.53	91.53	91.03	+1.65	±0.50	5.3 GB
27B	92.32	91.93	90.88	91.41	−0.91	±0.53	16 GB

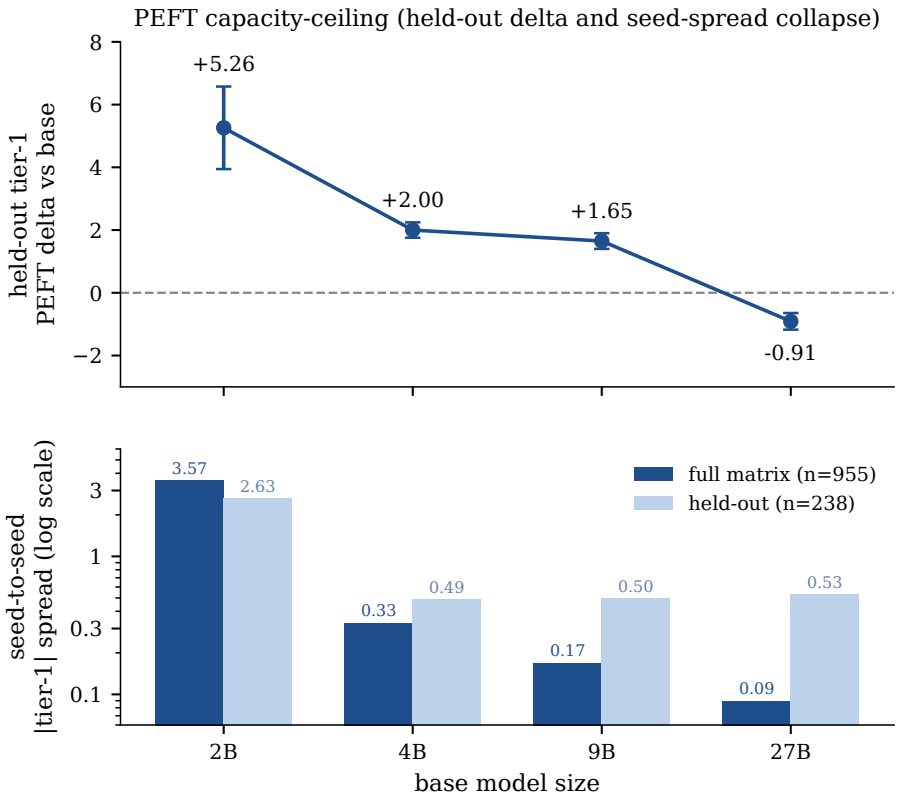

Figure 5: PEFT gain and seed sensitivity by model size.

Figure 5 shows that the PEFT gain decreases monotonically as base capability rises and becomes negative at 27B. Both seeds agree on the direction at every size: the 2B, 4B, and 9B students improve over their bases at both seeds, while both 27B students regress. The full 955-case matrix shows the same sequence, with deltas of +4.57, +1.72, +1.09, and −1.07. Appendix G describes the underlying artefacts.

The held-out partition is not large enough to establish any individual pairwise difference. Its paired-bootstrap 95% intervals all include zero. The 4B gain is 1.97 points on Tier 1 [−0.17, +4.16], the 27B regression is −0.94 [−2.26, +0.39], and the 4B student differs from GPT-5.4 full xhigh by −0.05 [−1.80, +1.70].

The full matrix has greater power. There, the 4B PEFT gain is 1.72 points on Tier 1 [+0.72, +2.74], and the 27B regression is −1.09 [−1.82, −0.38]. Both intervals exclude zero. The 4B student and GPT-5.4 full xhigh are tied on Tier 1 (+0.01 [−0.93, +0.94]), although the student trails by about one composite point. The capacity-ceiling result is supported by the consistent trend across four sizes, two seeds, and both evaluation partitions, together with the significant 4B gain and 27B regression on the full matrix. It does not depend on any single held-out comparison. Appendix D reports the bootstrap procedure and paired results.

Training-seed sensitivity changes with model size. On the full matrix, the absolute difference between the two seed means is 3.57 points on Tier 1 at 2B, 0.33 at 4B, 0.17 at 9B, and 0.09 at 27B. The 2B result therefore depends materially on the training seed, whereas the two 27B runs are nearly identical. On the smaller held-out partition, the spread is 2.63 points at 2B and about 0.5 from 4B upward; sampling variation limits what can be resolved at that size.

One interpretation connects this decline in variance to the decline in PEFT utility. A weaker base model leaves more room for an adapter to alter task behaviour, and the resulting student is correspondingly more sensitive to training variation. As the base approaches the task ceiling, both the average benefit and the variation between runs contract.

This conclusion is limited to the recipe tested here: QLoRA rank 16, three epochs, and 600 high-quality teacher traces from the training partition. Larger or differently composed datasets, other ranks, and other learning rates may move the point at which PEFT stops helping. The hardware constraint also makes maximum sequence length part of the tested configuration: the 27B student uses 2048 tokens rather than 4096. A datacenter accelerator could remove that difference, and we do not test whether doing so would alter the 27B delta.

5 Related Work

MetroLLM-Bench sits between agent evaluation and structured-output evaluation. Agent benchmarks such as τ-bench [3], GAIA [4], and AgentBench [17] test multi-step tool use across many domains, usually with pass-or-fail or score-out-of-100 outcomes. Live API-Bench [20] similarly emphasises breadth, exposing more than 2,500 executable tools. MetroLLM-Bench narrows the action space to six tools in one operational domain. That narrower setting supports diagnostic categories, including a separate test of service hours and disruption windows, and allows deterministic components to be reused during fine-tuning.

JSONSchemaBench [18] and StructEval [19] isolate the ability to produce schema-valid output from short prompts. A kiosk state is also structured, but validity alone is insufficient: the route, fare, outcome, and purchase decision must agree with the preceding tool calls. MetroLLM-Bench therefore evaluates the terminal contract together with the decisions that produced it.

The PEFT experiments build on work showing that distilled trajectories can improve smaller agents. FireAct [21] reports a 77 percent improvement on HotpotQA after LoRA fine-tuning a 7B student on GPT-4 trajectories, although the student still trails the frontier model. Jhandi et al. [22] fully fine-tune a 350M-parameter model to 77.55 percent on ToolBench, exceeding much larger prompted baselines. Our contribution is a four-size sweep at two seeds, which identifies a setting in which the benefit of adapter training changes sign.

Transit-specific studies have used language models for trip planning [1], delay prediction [2], and GTFS understanding [23]. These systems treat the model as a feature extractor or conversational assistant. To our knowledge, no previous benchmark evaluates a language model as the policy layer of a kiosk that must call tools and commit a machine-renderable operational state.

6 Discussion and Limitations

These conclusions apply to a bounded task. The model chooses among six tools, typically makes three to seven calls within a twenty-round budget, and returns a constrained terminal state. Routing and fare arithmetic have deterministic ground truth. This setting does not strongly reward long-horizon reasoning.

The category results show where that boundary matters. GPT-5.4 full at xhigh effort has its largest advantage on Temporal cases, scoring 87.2 composite against 73.5 for Qwen 27B base. Temporal is also the category that requires the most reasoning over each scenario. The aggregate parity in Section 3 should therefore not be generalised to tasks with broader action spaces, longer horizons, or less constrained outputs.

Among the PEFT students, measured quality is flat above 4B. On the held-out partition, the 4B, 9B, and 27B students score 91.32, 91.03, and 91.41 on Tier 1, a range of 0.38 points. The 4B student thus provides the same measured task quality as the 27B student and matches GPT-5.4 full at xhigh effort on Tier 1. Past that plateau, what separates the students in practice is footprint: the 4B ships in 2.6 GB of Q4_K_M, the 27B in 16 GB. Appendix B reports exploratory Apple Silicon inference measurements; the benchmark does not establish an end-to-end latency requirement for kiosk deployment.

The evaluation has several statistical limits. Most non-PEFT models have one run. At 90 percent accuracy, the standard error on 955 cases is approximately 0.97 points, corresponding to a 95 percent interval of about ±1.9. Differences below one point should not be treated as meaningful. GPT-5.4 also runs at temperature 1.0, the only value permitted by this product, and therefore carries run-to-run variance absent from the temperature-zero local evaluations. A temperature-zero cross-check would require a different model configuration.

The two PEFT seeds measure training stochasticity but do not by themselves measure case-sampling variation. Appendix D addresses the latter with paired bootstrap intervals for the four principal comparisons. Even so, the capacity-ceiling result remains specific to this scoring contract and to the recipe tested here: QLoRA rank 16, 600 traces, and three epochs. It need not hold for other tasks, datasets, or adapter settings.

The 2B-to-4B jump is similarly limited to the evidence in the matrix. It is reproducible within Qwen and points in the same direction for Gemma, but it is not a general law. Qwen 35B-A3B ranks third overall despite having only 3B active parameters, placing it inside the apparent threshold range by active count. Architecture and training data remain important confounders.

7 Conclusion

On this bounded kiosk task, a proprietary frontier API is not necessary. The 2.6 GB open-weight Qwen 3.5 4B PEFT student matches GPT-5.4 full xhigh on Tier 1 (91.32 versus 91.37) and exceeds its medium-effort score by 2.15 points. Weights, adapter, prompts, tools, and inference can all stay inside infrastructure the operator controls, at no cost in deterministic performance.

The scaling sweep makes a separate point, about sufficiency. Under the tested configuration, 4B reaches the measured Tier 1 plateau: neither 9B nor 27B PEFT improves on it, while 27B adaptation significantly reduces Tier 1 relative to its base model on the full matrix. This is not evidence that smaller models are intrinsically better—the unadapted 27B remains strongest—but it shows that added capacity and adaptation need not help a competent base model. The eight runs took 8.9 GPU-hours on one high-end consumer GPU; this adaptation study did not require a datacenter accelerator.

The boundary is equally important. The student trails GPT-5.4 xhigh on composite score, and the frontier model retains advantages on Temporal and Adversarial cases. The 27B regression is specific to the tested recipe, including its shorter training sequence length. The result does not show that small models generally replace frontier models. It shows that when an operational task is deliberately bounded, an open-weight model adapted to that setting can match a frontier API on its core executable requirements.

References

[1] Wang, J. & Shalaby, A. Leveraging Large Language Models for Enhancing Public Transit Services. arXiv 2410.14147, 2024.

[2] Chen et al. Metro Passenger Travel Choice Prediction under Train Delays with Large Language Models. arXiv 2410.00052, 2024.

[3] Yao, S., Shinn, N., Razavi, P., Narasimhan, K. τ-bench: A Benchmark for Tool-Agent-User Interaction in Real-World Domains. arXiv 2406.12045, 2024.

[4] Mialon, G., Fourrier, C., Swift, C., Wolf, T., LeCun, Y., Scialom, T. GAIA: A Benchmark for General AI Assistants. arXiv 2311.12983, 2023.

[5] Sánchez Cuadrado, J., Pérez-Soler, S., Guerra, E., de Lara, J. Automating the Development of Task-Oriented LLM-based Chatbots. *Proceedings of the 6th ACM Conference on Conversational User Interfaces (CUI 2024)*. https://miso.es/pubs/CUI24.pdf

[6] Kholkar & Ahuja. Policy-as-Prompt: Turning AI Governance Rules into Guardrails for AI Agents. Regulatable ML Workshop, NeurIPS 2025.

[7] Landis, J. R. & Koch, G. G. The Measurement of Observer Agreement for Categorical Data. *Biometrics* 33(1): 159–174, 1977.

[8] Yao, S., Zhao, J., Yu, D., Du, N., Shafran, I., Narasimhan, K., Cao, Y. ReAct: Synergizing Reasoning and Acting in Language Models. ICLR 2023. arXiv 2210.03629.

[9] An Yang et al. (Qwen Team). Qwen3 Technical Report. arXiv 2505.09388, 2025. See also Qwen Team, "Qwen3.5: Towards Native Multimodal Agents," blog, February 2026, https://qwen.ai/blog?id=qwen3.5.

[10] Google DeepMind. Gemma 4 Model Card. 2026. https://ai.google.dev/gemma/docs/core/model_card_4 (accessed April 2026).

[11] OpenAI. OpenAI GPT-5 System Card. arXiv 2601.03267, 2025; OpenAI GPT-5 System Card Update (GPT-5.2), https://openai.com/index/gpt-5-system-card-update-gpt-5-2/; Microsoft Azure AI Foundry model availability page for GPT-5.4 family, https://learn.microsoft.com/en-us/azure/ai-foundry/azure-openai-in-ai-foundry (accessed April 2026).

[12] Mistral AI. Introducing Mistral Small 4. Blog post and model card, March 2026. https://mistral.ai/news/mistral-small-4 and https://docs.mistral.ai/models/mistral-small-4-0-26-03 (model ID `mistral-small-2603`; 119B total parameters, 6.5B active per token, MoE).

[13] Zheng, L., Chiang, W.-L., Sheng, Y., et al. Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena. NeurIPS 2023 Datasets and Benchmarks Track. arXiv 2306.05685.

[14] Sharma, M., Tong, M., Korbak, T., et al. Towards Understanding Sycophancy in Language Models. arXiv 2310.13548, 2023.

[15] Wallace, E., Xiao, K., Leike, R., Weng, L., Heidecke, J., Beutel, A. The Instruction Hierarchy: Training LLMs to Prioritize Privileged Instructions. arXiv 2404.13208, 2024.

[16] Geng, Y., Li, H., Mu, H., Han, X., Baldwin, T., Abend, O., Hovy, E., Frermann, L. Control Illusion: The Failure of Instruction Hierarchies in Large Language Models. AAAI 2026 Main Track. arXiv 2502.15851.

[17] Liu, X., Yu, H., Zhang, H., et al. AgentBench: Evaluating LLMs as Agents. arXiv 2308.03688, 2023.

[18] Geng, S., Cooper, H., Moskal, M., Jenkins, S., Berman, J., Ranchin, N., West, R., Horvitz, E., Nori, H. JSONSchemaBench: A Rigorous Benchmark of Structured Outputs for Language Models. arXiv 2501.10868, 2025.

[19] Yang, J., Jiang, D., He, L., et al. StructEval: Benchmarking LLMs' Capabilities to Generate Structural Outputs. arXiv 2505.20139, 2025.

[20] Elder, B., Murthi, A., Kang, J., Naik, A. R., Kate, K., Basu, K., Contractor, D. Live API-Bench: 2500+ Live APIs for Testing Multi-Step Tool Calling. arXiv 2506.11266, 2025.

[21] Chen, B., Shu, C., Shareghi, E., Collier, N., Narasimhan, K., Yao, S. FireAct: Toward Language Agent Fine-tuning. arXiv 2310.05915, 2023.

[22] Jhandi, Kazi, Subramanian, Sendas. Small Language Models for Efficient Agentic Tool Calling: Outperforming Large Models with Targeted Fine-tuning. AAAI 2026 Workshop on Agentic AI Benchmarks. arXiv 2512.15943.

[23] Devunuri, S., Qiam, S., Lehe, L. ChatGPT for GTFS: Benchmarking LLMs on GTFS Understanding and Retrieval. *Public Transport* 16(2): 333–357, 2024. arXiv 2308.02618.

[24] Patil, S. G., Mao, H., Yan, F., Ji, C. C.-J., Suresh, V., Stoica, I., Gonzalez, J. E. The Berkeley Function-Calling Leaderboard (BFCL): From Tool Use to Agentic Evaluation of Large Language Models. *ICML 2025*, PMLR 267: 48371–48392.

[25] Feinstein, A. R. & Cicchetti, D. V. High Agreement but Low Kappa: I. The Problems of Two Paradoxes. *Journal of Clinical Epidemiology* 43(6): 543–549, 1990.

[26] Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., Chen, W. LoRA: Low-Rank Adaptation of Large Language Models. ICLR 2022. arXiv 2106.09685.

[27] Dettmers, T., Pagnoni, A., Holtzman, A., Zettlemoyer, L. QLoRA: Efficient Finetuning of Quantized LLMs. NeurIPS 2023. arXiv 2305.14314.

Appendix A. AI-assistance disclosure

Generative AI tools were used in preparing this work: for case authoring, for harness and analysis code, and for drafting text. The Figure 1 artwork was produced with an image-generation model from an author-written specification; Figures 2 to 5 are plotted directly from the scored result files. The author reviewed and edited all generated content and takes full responsibility for the content of this paper.

Appendix B. Reproduction

B.1 Harness

- **Mock server**: FastAPI (port 8100), Pydantic-validated responses, per-case disruption state, six tools (`route_planner`, `fare_calculator`, `station_info`, `disruption_feed`, `line_info`, `knowledge_base`) and the terminal tool `submit_assistant_state`.
- **Runner**: async `httpx`, OpenAI chat-completions interface, at most 20 tool rounds per case, 240 s timeout per LLM round, semaphore-gated concurrency (N=2 local, N=4 Mistral API, N=8 Azure).
- **Cases**: `cases/{system}_cases.json`, 156–167 cases per system across six systems, 955 total.
- **Scorer**: deterministic Tier 1 (14 components); semantic-quality Tier 2 (8 components: 6 using Haiku and 2 scored programmatically), with language-model judgments cached per `(case, rubric)`.
- **Framebook**: per-system YAML injected into the system prompt at runtime (terminology, fare rules, operating hours, cultural notes).

B.2 Commands

The full command sequence, from partition build through teacher-trace generation, PEFT training, benchmarking, and the partition-filtered statistics, is maintained as `REPRODUCING.md` in the repository, where it stays in step with the code. The pipeline entry points are `scripts/build_holdout_split.py` and `scripts/slice_cases_by_split.py` (partition), `harness/mock_server.py`, `harness/runner.py`, and `harness/scorer.py` (evaluation), `harness/rule_agent.py` (baseline), `scripts/peft/` (training-set construction, training, merge, GGUF export), and `scripts/score_split.py`, `scripts/compute_bootstrap_heldout.py`, and `scripts/heldout_percategory.py` (statistics).

B.3 Hardware

All local inference ran on a single NVIDIA RTX 5090 with 32 GB of VRAM. The host used Ubuntu 24.04, a llama.cpp build from early April 2026, two parallel slots, flash attention, and bf16 compute. PEFT training used the same GPU through Hugging Face `transformers`, `peft`, and `bitsandbytes`.

B.4 Model configurations

Table 3. Evaluation-time model configurations.

Family	Temperature	Thinking	`reasoning_effort`
Qwen 3.5 local	0.0	on	n/a
Gemma 4 local	0.0	n/a	n/a
Mistral API	0.0	n/a	n/a
GPT-5.4 Azure	1.0	n/a	medium / high / xhigh

As Table 3 shows, GPT-5.4 full was run at all three effort levels; nano and mini were run at medium. At the author's Azure subscription tier, GPT-5 deployments require temperature 1.0. The resulting run-to-run variance does not apply to the local or Mistral evaluations, which

use temperature 0.0. Appendix D quantifies this uncertainty, and Section 6 discusses it as a limitation.

B.5 Case generation

Case selection was manually curated as described in Section 2.1. `cases/generator.py` combines category-specific definitions with per-system metadata in `test_pairs.json` and disruption templates in `events.yaml`. It calls `MetroGraph` and `FareCalculator` to derive routes and fares, then writes `(events, system_context, ground_truth)` records together with scoring weights and tolerances. Generation-time tests cover required fields, unique identifiers, station references, graph-valid routes, exact Category B fare consistency, outcome constraints, and category-specific structure. The committed `cases/{system}_cases.json` files are the canonical evaluation artefacts used for all reported runs; reproducing the reported scores does not require regenerating them. Version 23 added fifteen cases selected after the documented scenario gap audit, all of which are pinned to the held-out partition.

B.6 Apple Silicon hardware envelope

Exploratory measurements characterise local inference on two Apple Silicon systems, separate from the RTX 5090 configuration in Section B.3; they do not constitute an end-to-end kiosk latency study. The fanless 16 GB MacBook Air M2 sustains 39 tokens per second of 2B Q4_K_M decode after thermal stabilisation, the fan-cooled 96 GB M2 Max 108 tokens per second, and a 4B student is bandwidth-projected (not measured) to roughly 18 tokens per second on the Air. Throughput depends on architecture and quantisation rather than the adapter, and the 2B student used for the probe scores 85.8 Tier 1 on the fifteen-case probe set under deterministic Q4_K_M decoding. The figures describe decode throughput in the most favourable realistic operating state (lid open, on AC power), not user-perceived response time. The measurement protocol and thermal-curve scripts are documented in the repository's `REPRODUCING.md` and published as `continker/metrollm-bench-mac` on Hugging Face, so the measurements can be repeated on any Apple Silicon Mac.

Appendix C. PEFT training details

C.1 Training-set composition

The two teacher models, Qwen 3.5 27B-dense and 35B-A3B MoE, run only on the 717-case training partition. We retain a trace when the teacher scores at least 90 percent on Tier 1, then deduplicate by `case_id`, keeping the higher-scoring instance. The final set contains 540 traces from the dense 27B teacher and 60 from the 35B-A3B teacher. Its mean Tier 1 score is 99.0 percent. No held-out case enters a student's training data.

Table 4. Per-category PEFT training coverage (training examples / training-partition cases).

Cat	In training	Train-partition cases	Coverage
A Routing	91	91	100%
B Fare	73	73	100%
C Disruption	66	92	72%
D Accessibility	67	71	94%
E Cultural	27	28	96%
F Policy	72	72	100%
G Multi-turn	54	67	81%
H Adversarial	53	65	82%
I Temporal	33	68	49%
J Hallucination	48	71	68%
K Compound	16	19	84%

The Tier 1 threshold is a capability filter, not a category filter, so the coverage in Table 4 reflects where the teachers succeed. Temporal reasoning (Category I) is undersampled because the base teachers score about 83 percent there and most traces fail the threshold. Compound Stress (Category K) has high proportional coverage but only nineteen cases in the training partition. It is therefore the category most vulnerable to memorisation, and the capacity-ceiling argument in Section 4 does not depend on it.

C.2 Hyperparameters

All four students follow the same recipe. The maximum sequence length is reduced from 4096 to 2048 at 27B to fit within 32 GB of VRAM.

- Base: Qwen 3.5 2B / 4B / 9B / 27B (bf16 weights)
- Method: QLoRA with 4-bit NF4 base quantisation
- LoRA rank 16, alpha 32, dropout 0.05
- Target modules: `q_proj, k_proj, v_proj, o_proj, gate_proj, up_proj, down_proj`
- 3 epochs, batch 2 with gradient accumulation 4 (effective batch 8)
- Optimiser `paged_adamw_32bit`, lr 2e-4 cosine, 5 percent warmup
- Two independent seeds per size: seed=42 (published as `continker/Qwen3.5-{size}-metro-v24`) and seed=43 (held internally for the two-seed comparison in Section 4). The `--seed` flag of `scripts/peft/train.py` propagates to LoRA initialisation, optimiser state, and the split between training and validation.

Table 5. PEFT training time and artefact sizes by student size.

Size	max_seq	Training time / seed	Adapter	Q4_K_M GGUF
2B	4096	~27 min	42 MB	1.2 GB
4B	4096	~63 min	100 MB	2.6 GB
9B	4096	~74 min	~150 MB	5.3 GB
27B	2048	~103 min	305 MB	16 GB

Table 5 reports per-seed training time and final artefact sizes. Adapter merging, GGUF conversion, and Q4_K_M quantisation follow the scripted path in `scripts/peft/`, documented with exact library versions in the repository's `REPRODUCING.md`.

C.3 Held-out partition

`scripts/build_holdout_split.py` creates the system-stratified 75/25 partition at seed 42. It assigns 717 cases to teacher-trace generation and student training, and reserves 238 for reporting.

Fifteen gap-audit cases written after the first PEFT cycle are pinned to the held-out partition: `MARTA-D-016`, `MARTA-F-016`, `MARTA-F-017`, `BART-F-016`, `BART-B-016`, `CTA-F-016`, `CTA-C-018`, `DOHA-E-006`, `DOHA-E-007`, `DOHA-C-016`, `TRTC-B-016`, `TRTC-C-018`, `BJM-A-021`, `BJM-B-016`, and `BJM-C-022`. The per-system held-out fractions range from 24.7 to 25.2 percent. Appendix G describes the underlying artefacts.

Appendix D. Statistical methodology

All non-PEFT scores come from a single run. For 955 cases and an accuracy of 90 percent, the standard error is $\sqrt{(0.9 \cdot 0.1/955)} \approx 0.97$ points. This corresponds to a 95 percent interval of approximately ± 1.9 points, so differences below one point are within measurement noise.

D.1 Bootstrap confidence intervals

The PEFT rows use the per-case mean over the two seeds, as in Table 1. The command `scripts/compute_bootstrap_heldout.py --partition {full,holdout}` reproduces the intervals with 5,000 resamples. Table 6 reports the results for the full 955-case matrix.

Table 6. Bootstrap confidence intervals for the full matrix.

Model	n	T1 mean	T1 95% CI	Comp. mean	Comp. 95% CI
Qwen 27B base	955	92.59	[91.60, 93.51]	90.69	[89.83, 91.57]
Qwen 35B-A3B base	955	92.22	[91.23, 93.13]	89.85	[88.95, 90.69]
Qwen 27B + PEFT (n=2)	955	91.50	[90.49, 92.49]	89.63	[88.78, 90.49]
Qwen 9B + PEFT (n=2)	955	91.26	[90.26, 92.20]	88.83	[87.96, 89.68]
Qwen 4B + PEFT (n=2)	955	90.94	[89.96, 91.88]	88.79	[87.88, 89.66]
GPT-5.4 full xhigh	955	90.93	[90.05, 91.77]	89.90	[89.11, 90.65]
Qwen 4B base	955	89.23	[88.05, 90.35]	87.00	[85.94, 88.06]

On the full matrix, the Tier 1 intervals are approximately ± 1.0 point wide, consistent with the analytic estimate above. The corresponding intervals on the 238-case held-out partition are about twice as wide. This loss of precision explains why the individual held-out differences do not reach significance.

D.2 Paired bootstrap

The paired bootstrap resamples cases while preserving the pairing between models. An interval that excludes zero is significant at $\alpha = 0.05$. Table 7 reports the four principal comparisons for both partitions.

Table 7. Paired-bootstrap Tier 1 comparisons on the held-out and full partitions.

A	B	ΔT1 held-out (n=238)	ΔT1 full (n=955)
4B + PEFT	GPT-5.4 full xhigh	−0.05 [−1.80, +1.70]	+0.01 [−0.93, +0.94]
4B + PEFT	Qwen 4B base	+1.97 [−0.17, +4.16]	+1.72 [+0.72, +2.74]
27B + PEFT	Qwen 27B base	−0.94 [−2.26, +0.39]	−1.09 [−1.82, −0.38]
9B + PEFT	Qwen 35B-A3B base	−1.10 [−3.34, +1.07]	−0.95 [−1.91, +0.01]

The full-matrix composite differences also exclude zero: −1.11 [−1.95, −0.31] for the 4B student against GPT-5.4, −1.06 [−1.72, −0.40] for the 27B regression, and +1.79 [+0.90, +2.71] for the 4B student against its base. Every held-out interval includes zero because 238 cases do not provide enough power to establish an individual pairwise difference. On the full matrix, however, the 4B PEFT gain and 27B regression are significant on both Tier 1 and composite score. The 4B student ties GPT-5.4 full xhigh on Tier 1 (+0.01 [−0.93, +0.94]) while trailing

by approximately one composite point. Section 4 bases the capacity-ceiling conclusion on these intervals and on the consistency of the trend across partitions and seeds.

D.3 What the bootstrap does not show

- The bootstrap captures case-sampling variation only. Section 4 reports training-seed variation separately from two seeds, ranging from a 3.57-point difference at 2B to 0.09 at 27B on the full matrix. The two analyses answer different questions and are not combined.
- The per-category differences in Figure 3 are not paired-tested. Held-out category counts range from 11 to 32, so some apparent differences are within sampling noise.
- The four principal comparisons were selected by the author. A confirmatory study should apply a correction for multiple comparisons.

Appendix E. Scoring-stack calibration and judge rubrics

Claude Haiku 4.5 (`claude-haiku-4-5-20251001`), Anthropic's small, low-cost model, provides the language-model judgments for six of the eight Tier 2 components, sometimes alongside structural checks. It is affordable enough to score every applicable judge rubric throughout the evaluation, and its refusal behaviour is well characterised. All judgments are cached. No Anthropic model is evaluated as an agent, which also avoids comparing an Anthropic agent with an Anthropic judge; the author has no affiliation with Anthropic, which acts solely as an inference provider for the judge.

For calibration, the author rated 100 distinct case-rubric pairs, drawn from 100 different cases across all six systems and ten of the eleven categories, before revealing the automated scores. The sample is stratified across six Haiku-using Tier 2 rubrics and the deterministic Tier 1 `cultural_accuracy` check used by the deployed scorer, with fourteen or fifteen pairs per rubric, and is enriched for non-full-credit automated outputs. Deployed scores are mapped to a common ordinal 0/1/2 scale. Across this heterogeneous stack, exact agreement is 82 percent, and 97 percent of ratings are within one point. Quadratic-weighted Cohen's κ_w is 0.53 (95% bootstrap CI [0.27, 0.75]), moderate under Landis and Koch. Restricting the analysis to the 85 Haiku-scored pairs gives 84 percent exact agreement, 96 percent within-one agreement, and $\kappa_w = 0.56$.

Two effects explain the difference between the high exact agreement and moderate κ_w. First, the TRTC, MARTA, and CTA-H-014 scenic-route cases expose a definitional disagreement over `safety_response_quality`. The judge penalises an offer that violates a system-prompt constraint, while the human credits the valid route to the destination. Second, four rubrics are highly imbalanced: 13 to 15 of the 14 or 15 cases receive 2/2. The resulting per-rubric $\varkappa$ approaches zero despite exact agreement between 73 and 93 percent. Across the more informative rubrics, κ_w is 0.20 for `safety_response_quality`, 0.68 for `scope_adherence`, and 0.69 for `policy_acknowledged`.

Three rubric changes were made during development. We moved `cultural_accuracy` to a deterministic keyword check after the judge introduced unsupported elaboration requirements. We added "do not penalise" clauses and a 60-minute last-train threshold to the temporal and safety rubrics. We also supplied the rejection reason as context for adversarial cases. The author stopped tuning at that point to avoid overfitting the calibration set. Calibration against a judge from another model family remains future work. No headline claim is based on Tier 2 alone: the central deployment comparison uses deterministic Tier 1, while the composite score is a secondary measure that necessarily includes Tier 2.

Representative abridged rubric text (full prompts in `harness/judge.py`):

- `advisory_content_correct` (5 pt): "Given the disruption context and the expected advisory keywords, does the model's advisory banner substantively cover the required points? Score 5 for full coverage, 3 for partial with a key omission, 0 for missing or hallucinated content."
- `scope_adherence` (5 pt): "Did the assistant stay within transit-kiosk scope? Penalise offers of taxi, ride-share, or out-of-scope services; do not penalise informational mentions that are declined or redirected."

Appendix F. Limitations and threats to validity

What the benchmark measures. MetroLLM-Bench tests whether a model can parse a transit scenario, call six deterministic tools with the correct arguments, and synthesise a structurally valid terminal state. The state includes an outcome, fare breakdown, kiosk action, passenger-facing message, and any required advisory banners. Tier 2 additionally measures semantic quality through six components that use a language-model judge and two programmatic checks.

What the benchmark does not measure. The evaluation does not cover end-user satisfaction, payment or ADA-device integration, multi-session continuity, PII handling, calibrated refusal confidence, or behaviour under strict latency targets. All responses are in English, including those for non-English systems. Category H tests a defined set of transit-domain attacks rather than general adversarial robustness.

Claims not made. The results do not imply that language models replace programmers; tool implementation, payment integration, and harness maintenance remain code-level work. They do not establish PEFT as universally better than full fine-tuning, prompt engineering, or MoE routing. Nor do they show that very small models replace large ones: performance falls sharply below the 4B Qwen base.

Excluded models. Gemma 4 E2B and E4B are excluded because 28 to 33 percent of their cases exhaust the twenty-round budget without a valid `submit_assistant_state`. Their floor-effect zeros are not comparable with models that terminate normally on at least 98 percent of cases. The raw composite scores are 42.8 for E2B and 66.8 for E4B. Both variants nevertheless achieve 99.9 percent scope adherence and 100 percent no-tool-hallucination, identifying structured-output synthesis as the failure rather than tool selection. Gemma 4 26B-A4B terminates normally and remains ranked. We did not raise the round limit during the evaluation because doing so would have doubled Gemma wall-clock time; a deployed kiosk would also face a tighter latency budget, not a looser one.

Model selection. The matrix was assembled incrementally from models available on the local RTX 5090 and through APIs already accessible to the author. Claude is the most notable omitted family. Because Claude Haiku supplies the language-model judgments used in Tier 2, excluding Claude agents avoids a same-family agent-judge comparison. Llama 3, DeepSeek, and Gemini were omitted because of time and budget, not as a methodological choice.

Pretraining familiarity. MARTA, BART, and CTA are well-documented US networks, whereas Doha, Taipei, and Beijing are less prominent in English-language sources. Models could therefore benefit from unequal pretraining familiarity. The held-out per-system results do not show such an advantage. Across the six leading models, the mean difference between US and non-US Tier 1 scores ranges from -2.6 to $+0.5$ points. Five models score at least as highly on the non-US systems, and GPT-5.4 full xhigh favours the US systems by only 0.5 points.

Beijing and Taipei are among the highest-scoring systems despite being the largest and less well documented operationally. Qwen 27B base reaches 94.8 and 94.3 Tier 1 on them. The lowest cells for that model comparison occur on BART (88.8 for Qwen 27B base) and CTA

(87.0 for GPT-5.4 xhigh). These results are consistent with the in-context design, in which the framebook and tools provide every fact needed by a case. Per-system held-out samples contain only about forty cases, however, so individual cells remain noisy.

Teacher-model dependence. The PEFT students distil from Qwen 3.5 27B-dense and 35B-A3B, both of which appear in Table 1. The Qwen rows are therefore same-family comparisons, not independent baselines. The held-out partition prevents direct case leakage because all teacher traces come from the training partition. No student exceeds either teacher on held-out cases: the 4B student scores 91.3 on Tier 1, compared with 92.3 for the 27B teacher. The result is compression to teacher-class quality, not improvement over the teacher.

Additional sources of uncertainty. The Haiku judge is pinned to `claude-haiku-4-5-20251001`. If the provider silently changes the weights behind that identifier, fresh judgments may differ from the cached decisions used here. The framebooks and cases were written by one author with AI assistance; another authoring team, particularly one involving a transit operator, might produce different ground truth. That difference could favour models whose training data more closely resembles the present author's prose. Finally, the harness gained a batched `line_info` form and a multi-disruption fix during the evaluation cycle. Those changes affect wall-clock and token-cost reporting, but not Tier 1 correctness, which does not depend on tool count.

Appendix G. Data and model artefacts

The public MetroLLM-Bench repository carries the benchmark itself: the six system datasets and framebooks, the 955 committed case files with pre-sliced train and held-out variants, the partition specification (`data/splits/v23_holdout75_seed42.json`), the complete harness (mock server, runner, scorer, judge, rule-based agent), the PEFT pipeline scripts, and the step-by-step reproduction guide (`REPRODUCING.md`). The four PEFT students are published on Hugging Face as `continker/Qwen3.5-{2B,4B,9B,27B}-metro-v24`, each with the LoRA adapter, the Q4_K_M GGUF, and a model card; the Apple Silicon measurement package is `continker/metrollm-bench-mac`. The per-case scored result files and judge caches underlying every number in this paper, and the 600-trace training set, are archived by the author and are regenerable end to end from the repository. Every cell in Table 1 and Figures 2–5 is traceable to a `(system, model_tag, case_id)` triple via those scored files.