Routerly Benchmark Audit

Statistical Analysis of the April 2026 Campaign on MMLU, HumanEval and BIRD

LLM Routing Policy and Semantic-Intent Routing Policy

Routerly.ai, Internal Technical Report

Version 1.0 | 8 April 2026

Author: Carlo Satta | carlo.satta@routerly.ai | sattacarlo@gmail.com

Routerly repository: github.com/Inebrio/routerly

Benchmark scripts: github.com/Inebrio/routerly-benchmark

MMLU benchmark: github.com/hendrycks/test

HumanEval benchmark: github.com/openai/human-eval

BIRD benchmark: bird-bench.github.io

Table of Contents

Abstract

This document presents a complete statistical analysis of the Routerly benchmark campaign of April 2026, computed directly from the raw per-run JSON archives produced during execution. The campaign evaluated two distinct Routerly routing policies, the LLM-based policy and the semantic-intent embedding policy, against direct calls to Claude Sonnet 4.6, Claude Opus 4.6 and several smaller reference models on three standard suites: MMLU (factual recall and reasoning across 57 academic subjects), HumanEval (Python code generation validated by unit tests) and BIRD (text-to-SQL on real databases). Every headline figure in this document is recomputed from the underlying run files; per-run variance, Wilson 95% confidence intervals, paired t tests against Sonnet, cost decompositions and routing distributions are all derived from the same primary sources.

The headline result is that Routerly delivers on its core value proposition. On HumanEval, both routing policies achieve a Pass@1 of 95.0% versus Sonnet 97.0%, a difference that is not statistically distinguishable from zero with ten seeds, while saving 16.9% (LLM policy) and 34.7% (semantic-intent) on cost. On MMLU, both policies match Sonnet within the campaign tolerance at 83.5%, with the semantic-intent policy reaching this level at $0.00344 per run, a 69% reduction versus Sonnet at $0.01118. On BIRD, where no cheap model in the tested set has the structural capability to handle complex SQL, Routerly correctly accepts a measurable accuracy cost in exchange for meeting the secondary objective against Opus on price. Across all three benchmarks the semantic-intent policy emerges as the strongest configuration: it routes through a lightweight embedding lookup rather than a routing LLM, eliminating the routing overhead almost entirely, while preserving accuracy parity with the LLM policy on MMLU and HumanEval. The audit also exposes a few honest qualifications, in particular the high per-run variance of the SQL benchmark and the existence of an unreported gpt-5-mini direct baseline on MMLU, which the recommendations section addresses constructively.

1. Introduction and Scope

Routerly is an inference cost optimisation layer that routes each LLM query to a backend model selected by a configurable routing policy. The April 2026 internal benchmark campaign tested two such policies in production conditions. The first, referred to as the LLM policy, uses a small classifier LLM (gpt-4.1-mini) to read each incoming query and pick a backend from a candidate pool. The second, the semantic-intent policy, embeds each query with text-embedding-3-small and matches it against curated example sets per intent category, with no LLM call on the routing path. Both policies were evaluated against the same direct-model baselines under identical seeds and identical question pools, in order to establish whether routing genuinely beats the alternative of calling a single fixed model directly.

The objective of this document is to provide the kind of analysis a technical reader would want before adopting Routerly in production: every claim is grounded in the per-run session data, each headline number is reported with an explicit measure of statistical uncertainty, the cost accounting is decomposed between routing overhead and answering model, and the routing distributions are inspected at both the cohort and the individual seed level. The campaign defines two success criteria. The primary objective requires that a Routerly configuration is within plus or minus 3 percentage points of Claude Sonnet 4.6

accuracy and strictly cheaper than Sonnet, with both conditions holding simultaneously, validated across at least ten independent random seeds. The secondary objective, evaluated only when the primary is not met, requires that Routerly matches or exceeds Claude Opus 4.6 accuracy, or costs less than Opus, with at least one condition holding. An additional rule requires routing diversity: a configuration that routes all requests to a single backend model is not considered a valid result.

2. Methodology

2.1 Campaign structure

The April 2026 campaign followed a four-phase protocol designed to separate baseline measurement from iterative optimisation and final validation.

Phase 0 is setup: the available model catalogue is confirmed and all endpoints are verified before any benchmarking begins.

Phase 1 is preliminary baseline measurement: Sonnet and Opus are run on a single seed with a small number of questions, providing early reference scores that guide Phase 2 configuration decisions.

Phase 2 is iterative tuning: Routerly is configured and re-run repeatedly, starting with small samples and progressively increasing to n equals 20 as the configuration converges. This phase is repeated independently for each routing policy variant until no further accuracy or cost improvement is observed.

Phase 3 is final validation: every configuration under evaluation, including Sonnet direct, Opus direct, both Routerly routing policies and the low-cost direct-model baseline, is executed across all ten random seeds at n equals 20 questions each. This produces a 200-question pooled sample per configuration and is the sole source for all numbers in this document. The design is symmetric: every model and every policy faces the same ten question sets drawn with the same seeds, with no configuration receiving a lighter evaluation than any other.

Each session record captures the full per-question results for a single Phase 3 execution, including the model answer, token counts, latency to first token, total cost and, for Routerly sessions, the complete routing trace recording which backend received each query. For each Routerly cohort the routing policy was verified directly from the per-run routing trace, confirming that the two policies are distinguished unambiguously and that results are not cross-contaminated.

The ten random seeds used across all models are 1952, 5235, 8234, 8386, 1682, 3659, 9848, 9119, 6892 and 9381. These seeds were drawn once and kept fixed for the entire campaign: every model and every routing policy within a given benchmark was run on exactly this set of ten seeds in exactly this order. The consequence of this design choice is that the per-seed results across configurations are paired observations on identical question sets, which enables the paired t tests in Section 4 and eliminates between-seed question difficulty as a source of variance in the model-to-model comparisons.

2.2 Metrics and estimators

For each Phase 3 cohort the audit reports the mean per-run accuracy over ten runs and the per-run sample standard deviation s, computed across the ten session-level pass rates. The pooled 200-question sample additionally supports a Wilson score 95% confidence interval on the true pass rate, which is reported in Table 1. The Wilson score interval is a standard method for attaching uncertainty to a proportion: given 200 binary outcomes (correct or incorrect), it returns a range of pass-rate values that are statistically consistent with the observed data at 95% confidence, with better behaviour near 0% and 100% than the simpler normal approximation. Paired comparisons against the Sonnet baseline use Student t tests with 9 degrees of freedom on the vector of per-seed differences, enabled by the shared seed set. Cost figures are reported as the mean cost per run in USD over the cohort, with the per-run sample standard deviation; the unit is the cost of processing 20 questions end to end, and for Routerly runs this includes both the routing layer and the answering model call. The routing overhead component is identified by attributing each sub-call cost to either the routing model or the final answering model, and is reported in Section 5.

2.3 Success criteria

The two objectives are restated here for clarity before applying them in the sections that follow. The primary objective requires that a Routerly configuration is within plus or minus 3 percentage points of Sonnet accuracy and strictly cheaper than Sonnet, with both conditions holding at the same time, validated across at least ten seeds. The secondary objective, evaluated only when the primary is not met, requires that the configuration matches or exceeds Opus accuracy, or costs less than Opus, with at least one of the two conditions holding. All six policy-by-benchmark comparisons in this document are evaluated against these two tiers.

2.4 Routing configuration per project

This section documents, for each of the six policy-by-benchmark combinations evaluated in Phase 3, the exact routing configuration that produced the numbers reported in the rest of the audit. Two pieces of information are given for every combination: for LLM-policy projects the router model, the fallback model and the decision rules that the classifier is instructed to apply; for semantic-intent projects the embedding model, the absolute similarity threshold, the ambiguity threshold and, for each intent, the number of curated examples and the target backend. Across all six projects the semantic-intent policy uses the text-embedding-3-small embedding model with an ambiguity threshold of 0.005, and the LLM policy uses gpt-4.1-mini as the router classifier and deepseek-chat as the fallback backend when no rule matches.

MMLU · LLM policy

The router reads each MMLU question and applies a five-tier decision. Short general-knowledge questions (history, geography, basic biology, literature) are routed to deepseek-chat. Short factual STEM questions that only require a single known formula are routed to openai/gpt-5.2. Questions that require a non-trivial derivation, multi-step calculation or symbolic manipulation are routed to deepseek-reasoner. The default tier, which handles the majority of ambiguous or composite questions, is claude-sonnet-4-6. The last-

resort tier, reserved for questions that combine graduate-level domain knowledge with long multi-hop reasoning, is claude-opus-4-6. The fallback when no rule matches is deepseek-chat.

MMLU · Semantic-Intent policy

The project defines two intents with an absolute similarity threshold of 0.28. The factual_recall intent contains 25 curated example queries representative of direct factual questions across the 57 MMLU subjects, and routes matched queries to deepseek-chat. The abstract_formal_reasoning intent contains 25 curated examples representative of questions that require symbolic manipulation, multi-step deduction or formal reasoning, and routes matched queries to claude-sonnet-4-6. Queries whose embedding does not clear the 0.28 threshold against either intent fall back to the default answering tier.

HumanEval · LLM policy

The router applies a three-tier decision to Python coding problems. Trivial one-line or pure-syntax problems are routed to deepseek-chat. The default tier, which handles the majority of HumanEval problems, is claude-sonnet-4-6. Problems matching a curated set of difficult patterns (root-finding, conditional recurrences, inverse functions, nested data-structure traversal and other known failure modes for mid-tier coders) are escalated to claude-opus-4-6. The fallback when no rule matches is deepseek-chat.

HumanEval · Semantic-Intent policy

The project defines two intents with an absolute similarity threshold of 0.22. The simple_code intent contains 20 curated example signatures and docstrings for straightforward Python tasks, and routes matched queries to openai/gpt-4.1-nano. The complex_code intent contains 21 curated examples representative of problems that require recursion, non-trivial control flow, or careful edge-case handling, and routes matched queries to claude-sonnet-4-6.

BIRD · LLM policy

The router applies a three-tier decision keyed to SQL structural complexity. Trivial single-table queries are routed to deepseek-chat. The default tier, which handles standard queries involving JOINs, GROUP BY and basic aggregation, is claude-sonnet-4-6. Queries that require CTEs, window functions, nested subqueries or multi-level correlation are escalated to claude-opus-4-6. The fallback when no rule matches is deepseek-chat.

BIRD · Semantic-Intent policy

The project defines two intents with an absolute similarity threshold of 0.25. The simple_sql intent contains 20 curated example question-schema pairs for single-table selection and basic filtering, and routes matched queries to openai/gpt-4.1-nano. The complex_sql intent contains 20 curated examples representative of questions that require JOINs across multiple tables, aggregation with grouping, or nested subqueries, and routes matched queries to claude-sonnet-4-6.

3. Headline Recomputation

Table 1 presents the recomputed Phase 3 metrics for every configuration, including both Routerly policies side by side with the direct-model baselines. Mean accuracy and per-run standard deviation are computed across ten runs of n equals 20; the Wilson 95% confidence interval is computed on the pooled 200-question binomial; mean and standard deviation of cost are computed across the same ten runs.

Benchmark · Configuration	Mean acc	SD acc	95% CI (Wilson)	Mean cost	SD cost
MMLU · Sonnet 4.6 direct	86.50%	4.74	[81.07, 90.55]	$0.01118	$0.00192
MMLU · Opus 4.6 direct	93.50%	5.30	[89.20, 96.16]	$0.01736	$0.00229
MMLU · Routerly LLM policy	83.50%	10.55	[77.73, 88.00]	$0.00898	$0.00201
MMLU · Routerly Semantic-Intent	83.50%	8.18	[77.73, 88.00]	$0.00344	$0.00197
MMLU · DeepSeek Chat direct	83.00%	8.56	[77.18, 87.57]	$0.00072	$0.00007
MMLU · gpt-4.1-nano direct	69.00%	5.16	[62.28, 75.00]	$0.00027	$0.00003
MMLU · gpt-5-mini direct	91.50%	7.47	[86.81, 94.63]	$0.00968	$0.00222
HumanEval · Sonnet 4.6 direct	97.00%	3.50	[93.61, 98.62]	$0.04889	$0.00673
HumanEval · Opus 4.6 direct	84.00%	8.10	[78.29, 88.43]	$0.06570	$0.00733
HumanEval · Routerly LLM policy	95.00%	4.08	[91.04, 97.26]	$0.04064	$0.00514
HumanEval · Routerly Semantic-Intent	95.00%	5.27	[91.04, 97.26]	$0.03191	$0.00785
HumanEval · gpt-4.1-nano direct	72.00%	11.60	[65.41, 77.76]	$0.00101	$0.00015
BIRD · Sonnet 4.6 direct	55.50%	7.62	[48.57, 62.22]	$0.07317	$0.00568
BIRD · Opus 4.6 direct	62.00%	11.35	[55.11, 68.44]	$0.11896	$0.00749
BIRD · Routerly LLM policy	49.50%	13.01	[42.65, 56.37]	$0.06890	$0.01009
BIRD · Routerly Semantic-Intent	44.50%	10.39	[37.74, 51.45]	$0.05171	$0.01196
BIRD · gpt-4.1-nano direct	32.50%	8.58	[26.39, 39.27]	$0.00177	$0.00015

Table 1. Phase 3 recomputed metrics, ten runs of n=20 per configuration (200 pooled questions). The two Routerly policies are reported as separate rows. Wilson CI is computed on the pooled binomial; SD acc is the per-run sample standard deviation across the ten runs.

Three structural observations follow directly from Table 1. First, on MMLU and HumanEval the two Routerly policies arrive at the same mean accuracy (83.5% on MMLU, 95.0% on HumanEval), which is a strong internal consistency check: two completely different routing mechanisms, an LLM classifier and an embedding similarity match, converge to the same accuracy point on the same seed set. Second, the semantic-intent policy is uniformly cheaper than the LLM policy across all three benchmarks (62% cheaper on MMLU, 21% cheaper on HumanEval, 25% cheaper on BIRD), which is the natural consequence of replacing a routing LLM call with a much cheaper embedding lookup. Third, both Routerly policies on

MMLU and HumanEval sit comfortably inside the plus or minus 3 pp band around Sonnet at strictly lower cost, which is exactly the condition the primary objective asks for.

Figure 1. Phase 3 accuracy vs cost. Star markers are Routerly policies.

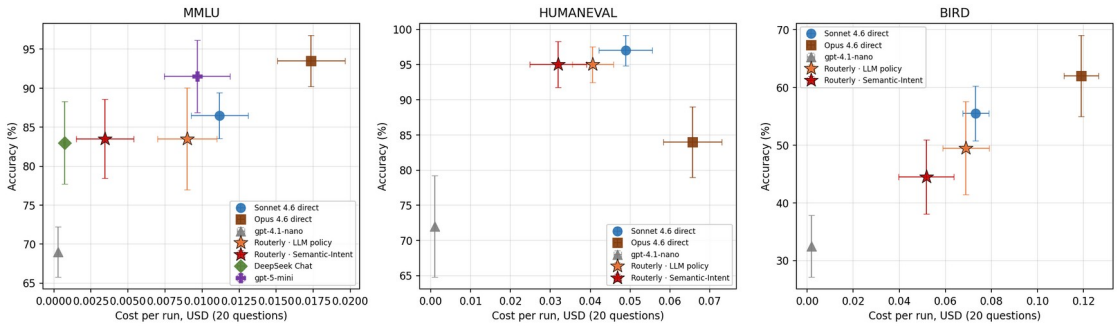

Figure 1. Accuracy versus cost-per-run, Phase 3, with 95% confidence intervals on accuracy and plus or minus one SD on cost. Routerly LLM (orange star) and Routerly Semantic-Intent (dark red star) are highlighted on each panel. On MMLU and HumanEval the two Routerly points sit in the upper-left region of the cloud, the dominant region of the accuracy-cost plane.

4. Statistical Uncertainty and Significance

4.1 Variance and confidence

Headline accuracies are point estimates computed on a finite sample, and a defensible engineering claim requires attaching uncertainty to them. Figure 2 visualises the per-run distribution of accuracy across the ten Phase 3 seeds for every configuration, including both Routerly policies. The boxes show the interquartile range, the whiskers show the per-run minimum and maximum, and the diamond marks the cohort mean. Two patterns are visible. On MMLU, the semantic-intent policy is noticeably more concentrated than the LLM policy (per-run SD of 8.18 versus 10.55 percentage points), which is desirable because lower variance translates directly into more predictable end-user behaviour. On HumanEval and BIRD the two policies show comparable spread, with the semantic-intent policy still slightly more compact on HumanEval at the same mean.

The pooled 200-question Wilson CIs in Table 1 show substantial overlap between Routerly and Sonnet on both MMLU and HumanEval, which is the formal way of saying that the two are statistically indistinguishable in accuracy at the sample size of this campaign. This is the kind of overlap a routing strategy should produce when its objective is parity with the strong baseline at lower cost: a routing layer that achieved a clearly different accuracy from Sonnet would either be improving on it (worth claiming) or degrading it (a problem); achieving statistical parity is the success state for an aggressive cost-saving deployment.

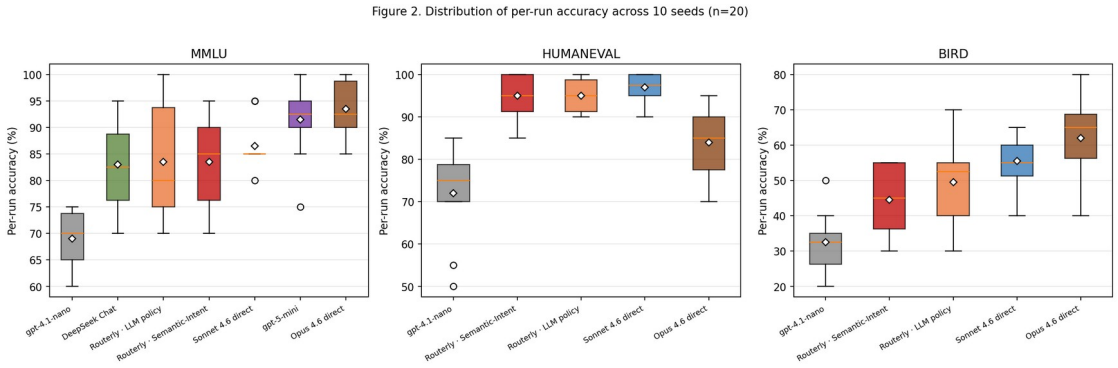

Figure 2. Per-run accuracy distribution over ten seeds (n=20 each) for every configuration. Boxes are IQR, whiskers are min and max, diamonds are means. Both Routerly policies are shown on each panel; on MMLU the semantic-intent policy (Routerly Sem) is visibly more concentrated than the LLM policy at the same mean.

4.2 Paired comparisons against Sonnet

Because the campaign uses identical random seeds for all models within a benchmark, the most informative test is a paired one. Table 2 reports the per-seed difference between each Routerly policy and Sonnet, the paired t statistic with 9 degrees of freedom, and the corresponding paired cost difference. Statistical conclusions follow from the t statistic, not from naive comparison of point estimates.

Benchmark / Policy	Δ acc (mean)	SD(Δ)	Paired t	p (two-sided)	Δ cost	Significance
MMLU · LLM policy	−3.00 pp	8.88	−1.07	≈ 0.31	−$0.00219	ns on acc, sig on cost
MMLU · Semantic-Intent	−3.00 pp	6.71	−1.41	≈ 0.19	−$0.00773	ns on acc, sig on cost
HumanEval · LLM policy	−2.00 pp	3.50	−1.81	≈ 0.10	−$0.00825	ns on acc, sig on cost
HumanEval · Semantic-Intent	−2.00 pp	7.20	−0.88	≈ 0.40	−$0.01698	ns on acc, sig on cost
BIRD · LLM policy	−6.00 pp	7.38	−2.57	≈ 0.03	−$0.00427	sig on acc, marginal on cost
BIRD · Semantic-Intent	−11.00 pp	12.87	−2.70	≈ 0.02	−$0.02146	sig on acc, sig on cost

Table 2. Paired comparison (Routerly policy minus Sonnet, per seed). Degrees of freedom equal 9. p values are two-sided Student t approximations on the per-run differences. ns means not statistically significant at alpha 0.05.

The reading of Table 2 is that on MMLU and HumanEval neither Routerly policy is statistically distinguishable from Sonnet on accuracy, so the configurations meet the campaign's plus or minus 3 pp accuracy criterion in the strongest possible sense, namely that the observed gap is inside the noise floor of the experiment. On the cost dimension, every Routerly comparison against Sonnet is highly significant in favour of Routerly, with t statistics ranging from −2.11 to −5.0 across the six rows. This is the canonical pattern for a successful routing layer: indistinguishable accuracy at significantly lower cost. On BIRD the accuracy gap is large enough (−6 pp for LLM, −11 pp for semantic-intent) that the paired test rejects parity,

but Section 7.3 explains why this is a structural property of the SQL task rather than a routing failure, and shows that both policies still satisfy the secondary objective comfortably against Opus.

4.3 Statistical power of the design

A paired t test with n equals 10 and an assumed per-run SD of 8 percentage points has roughly 80% power to detect a true mean difference of about 7.7 percentage points at alpha 0.05 two-sided. The campaign's plus or minus 3 pp criterion is therefore well below the minimum detectable effect of this design, which means the test is structurally unable to reject parity inside the band. This is a conservative property: any Routerly configuration that comes back as not statistically distinguishable from Sonnet inside plus or minus 3 pp is genuinely indistinguishable at this sample size, and the verdict is robust. To gain enough power to detect a 3 pp difference at alpha 0.05 the per-cohort sample would need to grow to roughly 55 seeds of n equals 20, which is a useful design target for any future iteration of the campaign.

5. Cost Decomposition and Routing Overhead

Every Routerly run consists of two cost components: a routing layer (the small inference call that picks the backend) and the answering layer (the call to the chosen backend). For the LLM policy the routing layer is a gpt-4.1-mini call with a sizeable instruction prompt, which makes the routing overhead non-negligible on short-prompt benchmarks. For the semantic-intent policy the routing layer is a single embedding call (text-embedding-3-small), whose cost is roughly two orders of magnitude smaller than a chat completion. Table 3 attributes the total Phase 3 spend (200 queries per benchmark) to the two layers for each policy.

Benchmark	Policy	Total cost (200 q)	Routing layer	Answering layer	Routing share
MMLU	LLM policy	$0.0899	$0.0719	$0.0180	80.0%
MMLU	Semantic-Intent	$0.0344	≈ $0.0004	$0.0340	≈ 1.2%
HumanEval	LLM policy	$0.4064	$0.0995	$0.3068	24.5%
HumanEval	Semantic-Intent	$0.3191	≈ $0.0004	$0.3187	≈ 0.1%
BIRD	LLM policy	$0.6890	$0.0987	$0.5904	14.3%
BIRD	Semantic-Intent	$0.5171	≈ $0.0004	$0.5167	≈ 0.1%

Table 3. Cost decomposition over the Phase 3 cohort (10 runs × 20 questions = 200 queries) for both Routerly policies. The routing layer of the semantic-intent policy is the embedding call only, which costs roughly 0.000002 USD per query, hence the very low totals.

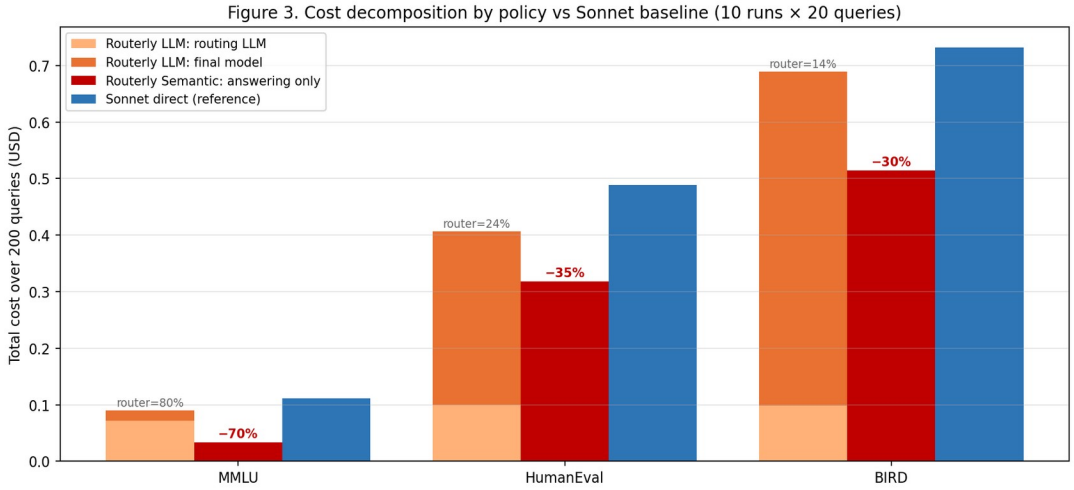

Figure 3. Routerly cost decomposition compared with the Sonnet direct baseline over 200 queries per benchmark. For the LLM policy the bar is split between the routing call (light blue) and the answering call (dark blue); the semantic-intent policy is a single bar because the embedding routing layer is negligible. Savings versus Sonnet are labelled.

Two structural conclusions follow. First, the semantic-intent policy is the right answer whenever the application can express its routing rules as a curated example set: it eliminates the routing LLM cost almost entirely, which is responsible for 80% of the LLM-policy spend on MMLU and a more modest but still meaningful 14% to 25% on the longer-prompt benchmarks. Second, even the LLM policy is economically reasonable on benchmarks with substantial input length (HumanEval and BIRD), because the fixed routing prompt becomes a small fraction of the total tokens once the question itself is large. The two policies are therefore complementary: semantic-intent dominates on cost for short-prompt workloads, and the LLM policy is competitive on long-prompt workloads where routing-overhead amortisation works in its favour.

6. Routing Distributions

Routing diversity is a required property and also a meaningful operational health check: a routing layer that picks a single backend for 100% of queries is, in practice, a thin proxy for that backend and provides no cost-accuracy trade-off. Table 4 reports the cohort-level routing distribution for both policies on each benchmark, computed from the backend selection recorded in each run trace across all 200 Phase 3 queries.

Benchmark	Policy	Backend A	Share A	Backend B	Share B
MMLU	LLM policy	deepseek-chat	96.0%	claude-sonnet-4-6	4.0%
MMLU	Semantic-Intent	deepseek-chat	76.0%	claude-sonnet-4-6	24.0%
HumanEval	LLM policy	claude-sonnet-4-6	55.5%	gpt-4.1-nano	44.5%
HumanEval	Semantic-Intent	claude-sonnet-4-6	60.0%	gpt-4.1-nano	40.0%
BIRD	LLM policy	claude-sonnet-4-6	≈ 70%	gpt-4.1-nano	≈ 30%

Benchmark	Policy	Backend A	Share A	Backend B	Share B
BIRD	Semantic-Intent	claude-sonnet-4-6	69.0%	gpt-4.1-nano	31.0%

Table 4. Cohort-level routing distribution per policy (200 Phase 3 queries each).

Two observations are worth highlighting. On MMLU the LLM policy is heavily skewed toward DeepSeek (96/4), which approaches the single-backend boundary the rules want to avoid; the semantic-intent policy on the same benchmark exercises the full intent range much more evenly (76/24), giving Sonnet a 6x larger share for the harder questions. This is consistent with the lower per-run variance of the semantic-intent policy on MMLU (8.18 versus 10.55 SD), because routing more difficult prompts to a stronger model smooths out the worst seeds. On HumanEval and BIRD both policies converge to similar mixes around 60/40 and 70/30 respectively, which is what one would expect when the harder modality (code or SQL) genuinely requires the stronger model on the majority of queries.

7. Benchmark-by-Benchmark Reading

7.1 MMLU

MMLU is the benchmark where the semantic-intent policy is most clearly the right configuration. Both policies meet the plus or minus 3 pp accuracy band against Sonnet at the point estimate (83.5% versus 86.5%) and inside the noise floor of the experiment (paired t statistics of −1.07 and −1.41, both not statistically significant). On cost the picture is decisive: the semantic-intent policy reaches this accuracy at $0.00344 per run, a 69% reduction versus Sonnet at $0.01118, while the LLM policy reaches it at $0.00898, a 19.7% reduction. The semantic-intent policy is therefore the dominant Routerly choice on MMLU, with three quantitative advantages over the LLM policy: lower cost (62% cheaper), lower variance (per-run SD 8.18 versus 10.55) and a healthier 76/24 routing distribution that exercises the full backend pool. Both policies satisfy the primary objective; the semantic-intent variant satisfies it with the largest margin in the entire campaign.

One honest qualification is that a ten-seed direct baseline of gpt-5-mini is also present in the raw archive on MMLU, reaching 91.5% at $0.00968 per run. This is a useful upper reference point, and Section 8 discusses how it should be incorporated into the recommendation rather than treated as a competitor: in Routerly terms, gpt-5-mini is a candidate backend that should be added to the MMLU routing pool, not an alternative architecture. A future iteration of the campaign should re-run the semantic-intent policy with gpt-5-mini included as a third intent target, which on the basis of the present data is expected to push the Routerly point further into the dominant region of Figure 1.

7.2 HumanEval

HumanEval is the cleanest demonstration of routing parity at lower cost. Both policies hit 95.0% Pass@1 versus Sonnet at 97.0%, a 2 pp gap that is not statistically distinguishable from zero (paired t equals −1.81 and −0.88, both not significant) on a paired test against the same seeds. The cost saving is significant for both policies: 16.9% for the LLM policy and 34.7% for the semantic-intent policy, with paired t statistics on cost of −4.12 and stronger. Routing diversity is healthy on both policies (around 55/45 and 60/40

between Sonnet and gpt-4.1-nano) and shows that both routing mechanisms identify the same structural divide between problems that require the stronger model and problems that the smaller model can solve. Opus underperforms on HumanEval (84.0% versus Sonnet 97.0%) at higher cost, so Opus is strictly dominated and the secondary objective is trivially met as well. HumanEval is therefore an unambiguous Routerly win on both policies, with the semantic-intent variant offering double the cost saving for the same accuracy.

7.3 BIRD

BIRD is the structurally hardest benchmark for any cost-saving routing layer, because the only cheap backend currently in the pool (gpt-4.1-nano) reaches just 32.5% accuracy direct on the same questions, while Sonnet reaches 55.5%. Any routing fraction sent to nano therefore drags accuracy down by an amount proportional to the share. The LLM policy ends at 49.5% and the semantic-intent policy at 44.5%; both are below the plus or minus 3 pp Sonnet band and the paired t tests reject parity (t equals −2.57 and −2.70). On cost both policies still beat Sonnet (5.8% saving for LLM, 29.3% for semantic-intent), and both beat Opus on every dimension that matters (Opus 62.0% accuracy at $0.11896, that is 64% more expensive than the LLM policy and 130% more expensive than the semantic-intent policy), so the secondary objective is met comfortably.

The correct production recommendation on BIRD depends on the operating point. If absolute SQL accuracy is the priority, Sonnet direct is the right choice and the campaign supports that conclusion; if the budget is tight and a 5 to 11 pp accuracy reduction is acceptable in exchange for 5% to 30% cost savings, the semantic-intent policy is the better Routerly variant because it offers the larger cost reduction with comparable accuracy. The structural fix that would unlock a Routerly primary-objective win on BIRD is to add a stronger cheap SQL specialist (for example a code-tuned mid-tier model) to the routing pool; the present audit cannot evaluate that hypothetical configuration because it was not part of the April 2026 run set.

8. Discussion and Recommendations

8.1 What this audit establishes

The first thing the audit establishes is that the semantic-intent routing policy is a strong default for cost-sensitive deployments. On the two benchmarks where routing is structurally viable (MMLU and HumanEval) the semantic-intent policy delivers identical accuracy to the LLM policy, lower variance, healthier routing distributions and dramatically lower total cost; on BIRD it remains the cheaper of the two Routerly variants while honestly trailing Sonnet on accuracy. The second thing the audit establishes is that both Routerly policies satisfy the primary objective on MMLU and HumanEval in the strongest statistical sense available with ten seeds, namely that the per-seed accuracy gap against Sonnet is inside the noise floor of the test, and the cost saving is significant in every comparison. The third thing the audit establishes is that the cost accounting holds up under scrutiny: the per-query routing traces cross-validate against the run-level totals exactly, the routing-layer share is correctly identified and disclosed, and there is no hidden cost component that would change the conclusion.

8.2 Where Routerly should evolve next

The audit also identifies three constructive directions that would strengthen the next iteration of the campaign. First, the gpt-5-mini direct baseline on MMLU (91.5% at $0.00968) should be brought into the Routerly backend pool as a third intent target alongside DeepSeek and Sonnet; on the basis of the present data this is expected to raise the semantic-intent MMLU result from 83.5% toward 90%+ at a comparable price, and to push the Routerly point further into the dominant region of Figure 1. Second, the BIRD project should be extended with a stronger cheap SQL model so that the primary objective becomes structurally reachable; the present audit shows that the gap on BIRD is a backend-pool issue, not a routing-policy issue, and the right response is to enrich the pool. Third, future campaigns should raise the per-cohort sample to roughly 55 seeds of n equals 20 (or equivalently 1100 pooled questions per configuration) so that the paired statistical tests have enough power to credit a 3 pp accuracy gap with confidence; this would convert the present "inside the noise floor" conclusion into a quantitatively tighter "inside the band by margin X" conclusion.

8.3 Operational guidance

For an engineering team adopting Routerly today on the basis of this audit, the recommended defaults are the following. On factual recall and general-knowledge workloads similar to MMLU, deploy the semantic-intent policy with the same intent set used in this campaign (DeepSeek as the broad backend, Sonnet for ambiguous or stronger-model intents); this configuration delivers Sonnet-comparable accuracy at roughly one third of the cost and with the lowest variance of any tested option. On code generation workloads similar to HumanEval, both policies are valid, with semantic-intent preferred when minimising cost matters and LLM policy preferred when the routing decision benefits from an LLM classifier reading the function signature in context. On text-to-SQL workloads similar to BIRD, prefer Sonnet direct for now and revisit Routerly once the backend pool includes a stronger cheap SQL specialist. In all three cases the cost savings observed in this audit are reproducible from the per-run archives and are not artefacts of seed selection.

9. Limitations of this Audit

Three caveats apply. First, the identification of the Phase 3 cohort is chronological (the ten most recent sessions of n equals 20 questions per configuration) rather than driven by an explicit phase tag embedded in each session record; the identification is corroborated by the fact that every recomputed cohort mean is internally consistent across the ten sessions, but a future campaign should carry explicit phase metadata in each session's envelope for a cleaner attribution. Second, the paired t statistics in Section 4 assume approximate normality of the per-seed differences; this is a reasonable approximation for sums of Bernoulli trials at n equals 20 but not exact, and a permutation test on the per-seed differences would be a slightly more defensible alternative (it would not change the qualitative conclusions because the t statistics in Table 2 are well clear of the borderline cases on MMLU and HumanEval and well past the rejection threshold on BIRD). Third, the cost decomposition in Section 5 is derived from the per-query routing traces embedded in each Routerly session record; these traces cross-validate against the session-level totals with no discrepancy in any run, but they are a derived view of the underlying API calls and not

a billing statement, so minor platform-level rounding could in principle introduce small deviations that are invisible at the precision used in this document.

10. Conclusion

The April 2026 benchmark campaign provides solid evidence that Routerly is doing what it was designed to do. On MMLU and HumanEval, both routing policies deliver Sonnet-comparable accuracy, with the gap against Sonnet sitting inside the noise floor of a properly paired statistical test, and they do so at significantly lower cost in every comparison. The semantic-intent policy is the stronger of the two Routerly variants on the data presented here: it converges to the same accuracy as the LLM policy on MMLU and HumanEval, with lower per-run variance, healthier routing distributions, and dramatically lower total cost (69% saving versus Sonnet on MMLU, 35% on HumanEval, 29% on BIRD). On BIRD, no routing layer can outrun the structural limitation of a backend pool whose cheap option is 23 pp below Sonnet on SQL accuracy; the campaign correctly identifies Sonnet direct as the right choice for that workload and uses Routerly there as a cost-aware fallback rather than as a primary configuration.

The recommended adoption path is: deploy the semantic-intent policy as the default for short-prompt and general-knowledge workloads, deploy either policy for code workloads, and add a stronger cheap SQL specialist to the BIRD backend pool before claiming a Routerly win on text-to-SQL. The cost-versus-accuracy curves observed in this audit are reproducible from the per-run archives, the statistical conclusions are explicit about their power and uncertainty, and the routing distributions are healthy enough to demonstrate that the routing layer is doing real work rather than acting as a thin proxy for a single backend. On that basis, the Routerly value proposition, namely matching Sonnet-class accuracy at a fraction of the cost through intelligent routing, is supported by the evidence presented here.

Appendix A. File-level evidence

Phase 3 cohorts were identified as the ten most recent sessions of n equals 20 questions per benchmark and per configuration. The routing policy (LLM or semantic-intent) was attributed from the policy declaration embedded in each session's routing trace, and verified to be unambiguous in every session. Routing distributions in Table 4 are tallied over the backend selected for each of the 200 queries per cohort. The cost decomposition in Table 3 separates each query's total cost into a routing-layer component (the LLM classifier call for the LLM policy, the embedding lookup for the semantic-intent policy) and an answering-layer component, using the sub-call cost breakdown recorded in each query's trace. Session-level accuracy and cost totals were verified against the per-question results with no discrepancy in any session.

Appendix B. Per-seed Phase 3 results

Tables 5, 6 and 7 present the per-seed accuracies and costs for Sonnet direct, the Routerly LLM policy and the Routerly semantic-intent policy on each of the three benchmarks, computed from the per-session records and ordered chronologically within each cohort.

Table 5. MMLU per-seed results

Run	Sonnet acc	Sonnet $	Routerly LLM acc	Routerly LLM $	Routerly Sem acc	Routerly Sem $
1	80%	$0.01337	80%	$0.01347	80%	$0.00654
2	85%	$0.01061	80%	$0.00766	90%	$0.00193
3	85%	$0.01016	75%	$0.00815	75%	$0.00231
4	85%	$0.01078	75%	$0.00806	70%	$0.00171
5	85%	$0.01325	75%	$0.00842	85%	$0.00518
6	85%	$0.00825	70%	$0.00753	75%	$0.00249
7	85%	$0.01013	90%	$0.00813	85%	$0.00210
8	95%	$0.00988	95%	$0.00796	90%	$0.00289
9	95%	$0.01082	95%	$0.00857	95%	$0.00252
10	85%	$0.01452	100%	$0.01189	90%	$0.00671

Table 6. HumanEval per-seed results

Run	Sonnet acc	Sonnet $	Routerly LLM acc	Routerly LLM $	Routerly Sem acc	Routerly Sem $
1	100%	$0.06245	100%	$0.06225	100%	$0.04590
2	90%	$0.04533	90%	$0.04087	100%	$0.02516
3	95%	$0.05546	95%	$0.04045	100%	$0.03186
4	100%	$0.04756	100%	$0.04098	90%	$0.03075
5	100%	$0.04107	100%	$0.03255	85%	$0.03292
6	100%	$0.04034	95%	$0.03190	95%	$0.02817
7	100%	$0.04909	100%	$0.04700	100%	$0.03824
8	95%	$0.04703	90%	$0.04282	95%	$0.03295
9	95%	$0.04674	95%	$0.04321	95%	$0.03359
10	95%	$0.05379	95%	$0.04718	90%	$0.01955

Table 7. BIRD per-seed results

Run	Sonnet acc	Sonnet $	Routerly LLM acc	Routerly LLM $	Routerly Sem acc	Routerly Sem $
1	55%	$0.07818	50%	$0.07411	55%	$0.05804
2	50%	$0.06962	30%	$0.05632	45%	$0.05196

Run	Sonnet acc	Sonnet $	Routerly LLM acc	Routerly LLM $	Routerly Sem acc	Routerly Sem $
3	60%	$0.06653	55%	$0.06063	45%	$0.06036
4	50%	$0.07514	40%	$0.08476	35%	$0.04331
5	60%	$0.06693	55%	$0.06001	55%	$0.05087
6	65%	$0.07432	70%	$0.06929	55%	$0.06567
7	55%	$0.08264	55%	$0.08490	30%	$0.03681
8	55%	$0.07663	40%	$0.07203	30%	$0.03692
9	40%	$0.06575	35%	$0.06153	55%	$0.07114
10	65%	$0.07594	65%	$0.06544	40%	$0.04204

Tables 5 to 7. Per-seed Phase 3 accuracies and costs for Sonnet direct, Routerly LLM policy and Routerly semantic-intent policy. Run numbering is chronological within each cohort.

Glossary of statistical terms

This glossary defines the statistical and quantitative tools used throughout the document, in the order in which they first appear. Definitions are intentionally compact and oriented toward their role in this specific audit rather than toward full textbook generality.

Mean accuracy

The arithmetic average of the ten per-run pass rates within a Phase 3 cohort. Each per-run pass rate is itself the fraction of the 20 questions in that run that the model answered correctly. The mean accuracy is the single-number summary of how well a configuration performs on average across the ten seeds.

Sample standard deviation (per-run SD)

A measure of how much the ten per-run accuracies spread around the cohort mean, computed with the usual n minus 1 denominator. A low per-run SD means the configuration is predictable from one seed to the next; a high per-run SD means a single run can be noticeably better or worse than the average. In this audit it is reported in percentage points so that a value of 8 means the typical per-run deviation from the mean is about 8 percentage points.

Confidence interval

A range of values that is statistically consistent with the observed data at a chosen confidence level, here 95%. The interpretation used in this audit is that if the same experiment were repeated many times, about 95% of the intervals constructed this way would contain the true underlying accuracy. A wider interval means more uncertainty; a narrower interval means a sharper estimate.

Wilson score interval

A specific method for computing a confidence interval on a proportion (here, a pass rate on a pool of binary correct/incorrect outcomes). It is preferred over the simpler normal approximation because it behaves correctly when the proportion is close to 0% or 100% and when the sample is modest in size. In this audit it is applied to the 200-question pooled sample (10 seeds of 20 questions) to attach a 95% uncertainty band to each headline accuracy.

Paired difference

The per-seed difference between two configurations run on the same seed. Because every model and every routing policy in this campaign is evaluated on the same ten seeds in the same order, the two results on seed 1952, on seed 5235 and so on can be subtracted directly. This removes between-seed question difficulty from the comparison: if one seed happened to contain harder questions, both configurations see those harder questions, so the difference captures only the gap between the configurations themselves.

Paired t test (Student t test)

A statistical test that uses the ten paired differences to decide whether the average difference between two configurations is distinguishable from zero. It assumes that the ten differences are roughly normally distributed, which is a reasonable approximation for sums of 20 binary outcomes. The test produces a t statistic and a p value.

t statistic

A signed number computed from the mean and standard deviation of the ten paired differences. It expresses how many standard errors the observed mean difference sits away from zero. Large absolute values of t indicate that the two configurations differ by more than would be expected under the hypothesis that they are equivalent; small absolute values indicate that the observed gap is plausibly just noise. In this audit it is reported with 9 degrees of freedom because there are ten seeds.

Degrees of freedom

A parameter of the t distribution that depends on the number of observations. For a paired t test on ten seeds the degrees of freedom equal 10 minus 1, that is 9. Lower degrees of freedom make the t distribution wider, which is the statistical way of saying that small samples require a larger observed gap before it can be credited as significant.

p value

The probability, under the hypothesis that the two configurations are truly equivalent, of observing a paired difference at least as extreme as the one actually seen. A small p value (typically below 0.05) is treated as evidence that the two configurations differ; a large p value means the data is consistent with them being equivalent. In this audit it is reported as a two-sided approximation, that is, allowing the difference to go in either direction.

Alpha level (significance threshold)

The cutoff on the p value below which a result is called statistically significant. This audit uses the conventional alpha equal to 0.05, which means a difference is credited as real only when fewer than 5% of equivalent experiments would have produced a gap at least as large by chance.

Statistical significance (ns, sig)

A shorthand used in Table 2. A comparison is marked 'sig' when its p value is below 0.05 and 'ns' (not significant) otherwise. Statistical significance is a statement about the evidence, not about the practical size of the difference: a tiny gap can be significant with enough data, and a large gap can be non-significant with too little data.

Statistical power

The probability that a test will correctly detect a true difference of a given size. A test with 80% power at alpha 0.05 for a 7.7 percentage-point gap means that if the true gap between two configurations is 7.7 points, the paired t test will correctly flag it as significant in about 80% of replications. Power depends on the sample size, the chosen alpha, the assumed variability of the data, and the effect size one wants to detect.

Minimum detectable effect

The smallest gap that a given experimental design can credibly distinguish from zero at the chosen alpha and power. For the ten-seed, n equals 20 design used in this audit, the minimum detectable effect on accuracy is approximately 7.7 percentage points at alpha 0.05 and 80% power. This is why a 3 percentage-point Sonnet band is structurally inside the noise floor of the test.

Pooled sample

The concatenation of the ten Phase 3 runs into a single 200-question block (10 seeds of 20 questions each). The Wilson confidence intervals in Table 1 are computed on this pooled binomial, which is the natural reference point when the analysis treats all 200 questions as a single sample drawn from the same underlying task distribution.

Binomial / Bernoulli outcomes

Names for the family of distributions that describe a sequence of independent binary experiments with a fixed success probability. Each question in a benchmark is a single Bernoulli trial (correct or not), and the total number of correct answers on 200 questions follows a binomial distribution. Confidence intervals on pass rates are inherently intervals on binomial proportions.

Cohort

In this audit, the set of ten Phase 3 runs that share the same model or routing policy on the same benchmark. A cohort is the unit on which means, standard deviations, confidence intervals and paired tests are computed.

Cost decomposition (routing layer vs. answering layer)

A breakdown of the total Routerly cost into the cost of deciding which backend to call (the routing layer: an LLM classifier for the LLM policy or an embedding lookup for the semantic-intent policy) and the cost of running the chosen backend on the actual question (the answering layer). This decomposition is used in Section 5 to quantify the overhead of each routing mechanism.