Dissertation on

Tinker RL Lab: A Multi-Framework Benchmark and Study of GRPO-Style Reinforcement-Learning Post-Training of Large Language Models

Cross-library attribution, signal-starvation diagnostics, and
a fourteen-suite held-out evaluation campaign

Submitted in partial fulfilment of the requirements for the award of the degree of

Master of Technology

in

Data Science and Machine Learning

UE20CS972 – Project Phase – 2

Submitted by:

Arvind C R PES2PGE24DS140

Under the guidance of

Ramesh Prakash Guledgudd

May 2026 – October 2026

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

FACULTY OF ENGINEERING

PES UNIVERSITY

(Established under Karnataka Act No. 16 of 2013)
Electronic City, Hosur Road, Bengaluru – 560 100, Karnataka, India

PES UNIVERSITY

FACULTY OF ENGINEERING

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the project report entitled **"Tinker RL Lab: A Multi-Framework Benchmark and Study of GRPO-Style Reinforcement-Learning Post-Training of Large Language Models"** is a bonafide record of the project work carried out by **Arvind C R** (SRN: **PES2PGE24DS140**), in partial fulfilment of the requirements for the award of the degree of Master of Technology in Data Science and Machine Learning of PES University, during the period May 2026 – October 2026. It is certified that all corrections/suggestions indicated for internal assessment have been incorporated in the report.

The candidate's individual contribution, the inherited Semester-3 group infrastructure, and the boundary between them are stated in the Declaration and in Chapter 1 §1.6.

. . . . .	. . . . .
Ramesh Prakash Guledgudd	**Head of the Department**
Project Guide	Department of CSE

. . . . . .

External Examiner

October 2026

Declaration

I declare that this project report is my own work. Where the work of others has been used, it is cited and acknowledged. The matter embodied in this report has not been submitted, in part or in full, to this or any other university or institution for the award of any other degree or diploma.

The multi-framework benchmark infrastructure, the baseline framework integrations, and the initial GRPO/PPO/DPO experimentation were produced by the Semester-3 project group, of which I was a member, and are inherited here as the shared foundation. The research direction, the experimental expansion, the statistical analyses, the P1–P8 paper series, the reporting and registry artefacts, the adaptive group-size controller, the applied fraud study, and the E1–E14 held-out evaluation campaign reported in this document are my individual contribution, carried out under the guidance of my project guide.

Every quantitative result in this report is bound to an evidence file in the project repository. Where a result is incomplete, blocked, or negative, it is reported as such rather than omitted. The boundary between inherited and individual work is stated precisely in `platform_hybrid/sem 4 work/PROVENANCE.md`.

.....

Arvind C R

SRN: PES2PGE24DS140

Acknowledgement

I thank my project guide, **Ramesh Prakash Guledgudd**, for direction and sustained feedback throughout this work, and for the freedom to pursue the conservative and negative results that form a significant part of it.

I thank the Semester-3 project group for the shared multi-framework benchmark and the baseline integrations on which the Semester-4 studies build. I thank the maintainers of the benchmark suites used in the E1–E14 campaign, whose native evaluators made an independently checkable evaluation possible, and the platforms — Tinker, Modal, and Google Cloud — on which the experiments ran.

Finally, I thank the reviewers whose questions at earlier reviews shaped the reporting discipline this report depends on: the insistence on stating what a number does *not* establish has improved the work materially.

Arvind C R

Abstract

This report presents a multi-framework benchmark for group-relative reinforcement-learning post-training of large language models, a study of that post-training's behaviour across libraries, scales, group sizes and datasets, and a held-out evaluation of a single trained actor against fourteen agentic and reasoning benchmark suites.

The benchmark is multi-framework by construction: a shared task, reward and decoding configuration is dispatched through thin adapters onto each backend's own interface, so the same experiment is runnable on any of them without re-implementing the grader. That design property is distinct from completed-run parity, and this report does not conflate the two. Of the cross-launcher roster, only Tinker and TRL produced completed runs in this release; the veRL and OpenRLHF entries in the comparison artefact are dry-run placeholders that resolve a launch plan without executing it, and are never pooled with real results or read as framework-comparison evidence (Chapter 3 §3.8; Chapter 5 §5.1). Comparative claims in this report are therefore constrained to the runs actually recorded, and the diagnostic portability results are stated at the strength their own source asserts.

The central methodological commitment is *attribution*: every comparative claim is made with the entire training stack held fixed except the one factor under test. Four de-confound pillars — a matched same-stack PPO/GRPO contrast, the Zero-Variance Fraction as a signal-starvation diagnostic, a group-size sweep, and a length-bias and held-out-generalisation study — are reported under that commitment, alongside a cross-library cross-scale scaling study, a minimum reporting standard for stack-conditioned results, a machine-readable stack registry, an adaptive group-size controller derived from the ZVF diagnostic, and an applied credit-card fraud case study.

The principal results are deliberately conservative. Across more than seventy runs spanning roughly 2.4 orders of magnitude in model scale, **no reliable scaling law is recovered from reward curves**. A same-stack comparison of PPO and

GRPO (Qwen2.5-0.5B, five paired seeds) finds the two estimators **statistically indistinguishable** (paired $\Delta = -0.002$, $p = 0.374$). A three-seed group-size sweep ($G \in \{2, 4, 8, 16\}$) finds its arms statistically indistinguishable and does **not** establish an optimum. A roster-level flag marks four of eleven GRPO runs as peaking before 65% of training and then decaying, but its thresholds are post hoc and it is not specific to length, so it is **not** evidence of length bias; in the controlled sixteen-run comparison neither GRPO nor Dr. GRPO shows the pattern, and completion length drifts shorter rather than longer.

The E1–E14 campaign evaluates one Tinker-trained actor (`Qwen/Qwen3.6-35B-A3B` with the `pavlov-portfolio-qwen36-seed809-stepfinal` adapter) using each suite's own native evaluator at a pinned revision, with no per-lane tuning. Three replacement scopes are strictly complete: LAB-Bench public split (1,967/1,967 questions, micro accuracy 0.2288), AgentDojo benign utility (97/97, utility 0.9072), and VerilogEval (312/312, 129/312 = 41.35% pass@1). Omni-MATH is terminal-complete under its native protocol (all 4,428 dispositions recorded, 4,426 accepted; official accuracy 2,271/4,428 = 51.31% reproduced). Two carry full verified scores under their original contracts: SWE-bench Pro (2/731 = 0.274% pass@1, with all eighteen non-native outcomes retained in the denominator) and VerilogEval. The remaining lanes are reported as partial, quota-blocked, or externally blocked, each with an explicit terminal state and a named reopen condition.

Original-contract and replacement-scope benchmark results are never pooled, no cross-suite aggregate is computed, and no claim of improvement over a baseline is made anywhere in this work. All structural claims were re-verified by deterministic code checks on 19 September 2026 (11/11 PASS).

Keywords: GRPO, RLVR, zero-variance fraction, signal starvation, stack-conditioned results, benchmark evaluation, reproducibility, held-out evaluation, attribution.

Contents

Declaration **i**

Acknowledgement **ii**

Abstract **iii**

I Foundations **1**

1 Introduction **2**

1.1 Background: critic-free RL post-training and the group-relative estimator 2

1.2 A Motivating Example 5

1.3 Problem Statement 8

1.4 Objectives 9

1.5 Scope and Non-Goals 11

1.6 Contribution of the Candidate 11

1.7 Organisation of the Report 14

2 Literature Survey **16**

2.1 From RLHF and PPO to Group-Relative Methods 16

2.2 Direct Preference Optimisation as a Contrast 20

2.3 Fixing GRPO's Biases and Collapse 21

2.4 Verifiable Rewards, Frameworks and Reporting Practice 24

2.5 Scaling and Layer-Adaptation Context 26

2.6 Process-, Tree- and Stability-Aware Variants 28

2.7 Positioning the Portfolio 29

3 System Requirements Specification **34**

3.1 Overview and Actors 34

3.2 Functional Requirements 36

3.3 Non-Functional Requirements 39

3.4 Test-Case Specification 40

3.5 Requirements Traceability Matrix 42

3.6 Hardware Requirements 45

3.7 Software Requirements 46

3.8 Constraints and Assumptions 47

II Design and Implementation **49**

4 Proposed Methodology **50**

4.1 The Design Commitment to Attribution 50

4.2 Per-Step Telemetry 55

4.3 Formalising the Zero-Variance Fraction 56

4.4 The Eight-Study Portfolio 61

4.5 The Design of the E1–E14 Held-Out Evaluation Campaign 65

5 Implementation Details **72**

5.1 Two Rosters, One Dispatch Surface 72

5.2 The Adapter Architecture: What versus Where 74

5.3 The Tinker GRPO Loop 77

5.4 The Parallel Runner and a Real Logging Defect 78

5.5 The Colab White-Box Path (Per-Layer Gradient Profile) 79

5.6 Software Modules 81

5.7 The E1–E14 Evaluation Harness 81

5.8 The Bridge and Sampler Path 84

5.9 Recovery Re-implementations under `zvf-program/` 85

III Results **88**

6 Results I: Core GRPO Studies (P1–P4) **89**

6.1 Scope, Sources and Evidence Boundary 89

6.2 P1 — Cross-Library and Cross-Scale GRPO Behaviour 91

6.2.1 The anchor pool and the fitted saturation forms 91

6.2.2 The cross-scale test and its nulls 92

6.2.3 Where the geometric template fails 93

6.2.4 A withdrawn architectural contrast 94

6.3 P2 — The Zero-Variance Fraction 94

6.3.1 Definition and measurement 94

6.3.2 Measured magnitudes across libraries 95

6.3.3 The all-correct wall and the mastery–incapacity aliasing . 96

6.3.4 Prognostic value 98

6.4 P3 — Group Size and Contrast Density 99

6.4.1 The measured sweep 99

6.4.2 The same-stack four-axis audit and the reconstructed budgets 101

6.4.3 The curriculum and matched-token null results 102

6.5 P4 — Length Bias and Held-Out Generalisation 104

6.5.1 The controlled null 104

6.5.2 The roster-level verbosity-trap audit 105

6.5.3 Elevated analyses 107

6.5.4 Held-out generalisation and the scale extension 109

6.6 Cross-Pillar Observations and Limitations 110

7 Results II: Reporting, Registry and Controller (P5–P7) **112**

7.1 A Minimum Reporting Standard for Stack-Conditioned Results (P5) 112

7.2 The GRPO-Registry (P6) 116

7.3 From Diagnostic to Controller (P7) 119

8 Results III: Applied Study (P8) **124**

8.1 Dataset, splits, and the two arms 125

8.2 The head-to-head result 126
8.3 Why the seat does not move even at parity 127
8.4 What the measured evidence adds 128
8.5 Where the LLM does contribute: four capability gaps 130
8.6 The hybrid architecture 131
8.7 Limitations 131
8.8 Relation to the thesis 132

9 The E1–E14 Held-Out Evaluation Campaign 133
9.1 The evaluated actor and the one-actor design 134
9.2 Suites with strictly complete scopes 136
9.3 Original-contract results 140
9.4 Partial and quota-blocked lanes 141
9.5 The E4 rerun as a diagnostic 143
9.6 Externally blocked lanes and terminal states 144
9.7 Summary of the campaign at the point of writing 146

IV Synthesis 147

10 Cross-Study Synthesis, Threats to Validity, Conclusions and Future Work 148
10.1 Cross-Study Synthesis 148
10.2 Rigour and Threats to Validity 152
10.3 Conclusions 156
10.4 Future Work 158

References 161

V Appendices 165

A Complete Run Registry 166
A.1 Composition and temporal split 167

A.2 Semester-3 shared benchmark runs 168
A.3 Semester-4 P1–P8 study runs 170
A.3.1 Opening and null-result runs 175
A.4 E1–E14 held-out campaign lanes 179
A.5 Completeness limitations 183

B Reproducibility and Provenance 185
B.1 What this appendix establishes 185
B.2 The pinned runtime 185
B.3 The reviewer's entry point 186
B.4 Where the documentation has drifted from the code 187
B.5 The audit apparatus 187
B.6 The deterministic ledger check 188
B.7 The receipt model 190
B.8 Evidence loss in the finish era 190
B.9 The Semester 3 and Semester 4 provenance boundary 191
B.10 Residual non-determinism and the limits of reproduction 192

C Notation, Symbols and Abbreviations 194
C.1 Scope and convention 194
C.2 Symbols 194
C.3 Four collisions worth flagging 196
C.4 Abbreviations 196
C.5 The E1–E14 lane identifiers 198

List of Figures

1.1 The GRPO training loop 3

1.2 The confounding problem this work addresses 6

1.3 Boundary between the inherited Semester-3 group infrastructure and the individual Semester-4 contribution reported in this document 12

2.1 Algorithmic lineage from RLHF through PPO and DPO to GRPO and its variants 18

2.2 The eight studies P1–P8 grouped by theme, with the question each one addresses 32

4.1 The four de-confound pillars 51

4.2 The attribution design 52

4.3 Per-step telemetry recorded by the unified harness, and the destinations each record flows to 56

4.4 Definition of the Zero-Variance Fraction 58

4.5 Gradient utilisation, $1 - \mathrm{ZVF}$, as a function of group size G . . . 59

4.6 Design of the E1–E14 held-out evaluation campaign: one frozen actor, each suite graded by its own native evaluator at a pinned revision, with original-contract and replacement-scope results kept separate 66

5.1 The evidence chain from sealed request through execution and native grading to a deterministic ledger check 73

5.2 Six reinforcement-learning backends behind one Tinker-style API 75

5.3 Evaluation harness internals 82

6.1 Cross-scale behaviour of reward across the studied model range . 90

6.2 Mean Zero-Variance Fraction by backend library 95

6.3 Held-out accuracy as a function of group size G over three seeds . 100

6.4 Roster-level reward trajectories 104

9.1 Terminal status of all fourteen E1–E14 lanes 133

9.2 LAB-Bench public split, per-category accuracy, all 1,967 questions graded across eight categories 136

9.3 VerilogEval decomposition: two native framings of 156 tasks, 67/156 code-completion and 62/156 spec-to-RTL, summing to $129/312 = 41.35\%$ pass@1 136

9.4 The E4 rerun diagnostic 143

9.5 Terminal-state taxonomy used to close every lane 145

10.1 Threats to validity grouped as internal, construct and external, each with the mitigation applied in this work 153

List of Tables

2.1 Positioning of the eight portfolio papers (P1–P8) against closest prior art and the narrowed axis each claims. 29

3.1 Requirements traceability matrix: each functional requirement mapped to the artefacts that implement it and the audit check that verifies it. 42

4.1 The four de-confound pillars, the single factor each isolates, and the design that holds the remainder fixed. 53

8.1 Head-to-head result on the synthetic card-fraud set, by arm and evaluation split. 126

A.1 Registry composition by logging project, with temporal split and distinct base models. 167

A.2 Semester-3 cross-framework shared benchmark runs. 168

A.3 Semester-3 Modal campaign runs. 169

A.4 Semester-3 group-size sweep runs. 170

A.5 Semester-4 P1-P8 study runs: each claim mapped to its linked runs, stack, model and algorithm. 171

A.6 Semester-4 opening and null-result runs: the curriculum, token-budget, hard-prompt, loss-formulation and layer-freeze blocks. . . 176

A.7 E1-E14 held-out campaign lanes with their run identifiers, stacks, platforms, models, algorithms and group sizes. 179

A.8 Field coverage across the 1,662-run ledger: rows carrying a value against rows blank or UNKNOWN. 183

B.1 The deterministic ledger check by class, listing each specific check and what it establishes. 189

C.1 Notation: symbols used in this report, their meaning, and any divergence in how the underlying sources use them. 194

C.2 Abbreviations and their expansions, with names that no source expands recorded as identities rather than glossed. 197

C.3 The E1-E14 lane identifiers, their original contracts, replacement scopes and terminal states. 199

Part I

Foundations

Chapter 1

Introduction

1.1 Background: critic-free RL post-training and the group-relative estimator

Post-training is now the stage at which a base language model is turned into something that reliably follows instructions and solves problems. For several years the dominant recipe was reinforcement learning from human feedback (RLHF), in which a learned reward model is trained on human preference comparisons and then used as the reward signal for a policy-optimisation loop. RLHF works, but it introduces a second learned model whose errors the policy can exploit, and it makes reward construction expensive and hard to audit. The alternative that has taken hold more recently is reinforcement learning with verifiable rewards (RLVR): instead of a learned reward model, the reward comes from a programmatic checker — exact-match against a reference answer for grade-school mathematics, unit tests for code, a formal or semi-formal validator for structured output. The checker is deterministic, transparent, and free to run. What it gives up in generality it gains in measurability: every reward is a fact about a generated string rather than a prediction from a model[1] (Figure 1.1).

The workhorse optimiser for both settings has been proximal policy optimisation (PPO), which optimises a clipped surrogate objective against a value function critic. The critic is a second network, usually of comparable size to the policy, trained to predict the expected return of the current state; its predictions supply the baseline that the advantage estimator subtracts. This works, but it is expensive and fragile. The critic consumes memory that could hold policy parameters, it has its own learning rate and its own failure modes, and its errors propagate

[1] Source: platform_tinker/reports/esa_phase1/Phase1_Project_Report_ZVF.tex.

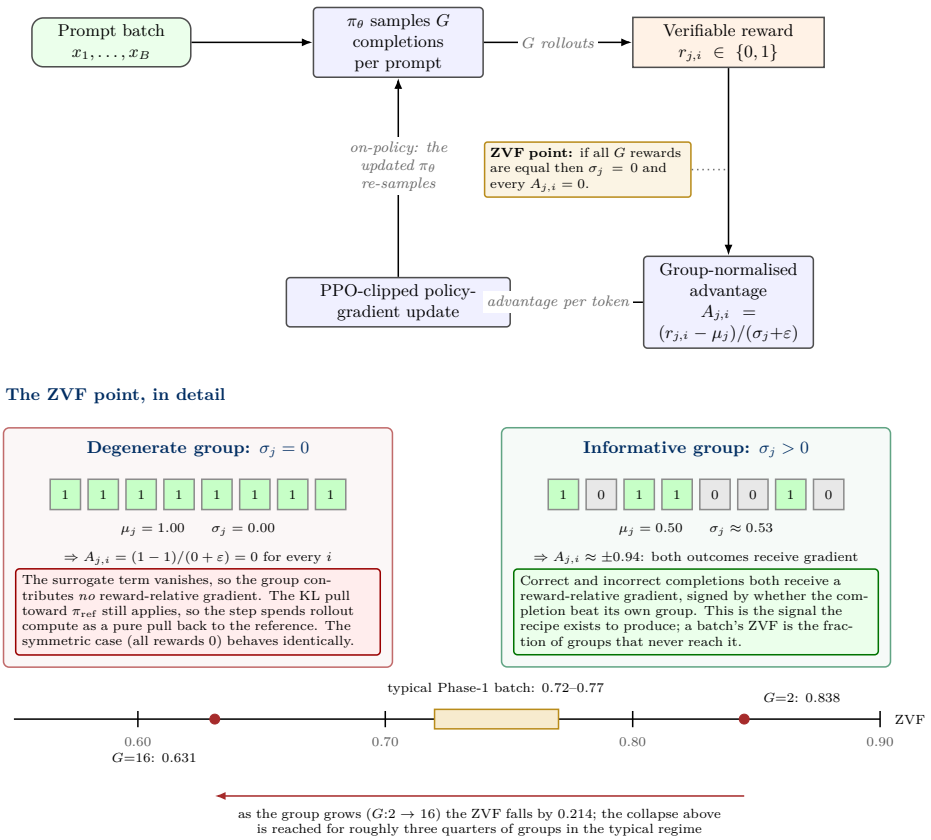


Figure 1.1: The GRPO training loop. A prompt is sampled G times, each completion is scored by a reward function, advantages are formed relative to the group, and the policy is updated. When all G rewards coincide the group-relative advantage collapses to zero and no gradient flows.

directly into the policy gradient as biased advantages.

Group Relative Policy Optimization (GRPO) removes the critic entirely. Rather than learn a value function, GRPO samples a group of G completions for the same prompt, scores all of them with the verifier, and derives the advantage of each completion from the group's own reward statistics. Concretely, for a group with rewards $r_1, \ldots, r_G$, the advantage assigned to completion i is a within-group standardisation,

$$A_i = \frac{r_i - \mu_g}{\sigma_g + \varepsilon}, \qquad \mu_g = \frac{1}{G}\sum_j r_j, \qquad \sigma_g^2 = \frac{1}{G}\sum_j (r_j - \mu_g)^2,$$

which is then used in a clipped, KL-regularised policy-gradient update. The empirical baseline is the group mean; there is no value network to train, no value loss to weight, and no critic to keep in sync with the policy. The trade is a much simpler optimiser for a much smaller effective batch, since G samples of the same prompt now carry one learning signal rather than G independent ones.

That trade has a sharp consequence, and it is the observation this project is organised around. Because the advantage is a *within-group* standardisation, its numerator is a deviation from the group mean and its denominator is the group's own spread. When every completion in a group receives the same verifier outcome — all correct, or all incorrect — the group contributes nothing: $\sigma_g = 0$, every numerator is zero, and every advantage A_i is exactly zero. The group has been fully sampled and fully scored, at full token cost, and the reward-relative part of the update is identically zero. The KL penalty toward the reference model still acts, so the policy does drift slightly toward the reference; but no *reward-driven* learning occurs on that group. We call the fraction of a training batch that falls into this state the **Zero-Variance Fraction (ZVF)**, and it is the single measurement that most of this report's diagnostics are built around[2].

ZVF is not a rare edge case. In the regime this project studies it is the *typical* state of a batch: the Phase-1 report records a ZVF of approximately 0.72–0.77 on its main configuration, meaning that roughly three quarters of a rollout batch contributes no reward-relative gradient at all[3]. The fraction is not fixed, however, and it responds to the one knob that most directly controls it: measured mean ZVF falls from 0.838 at $G = 2$ to 0.631 at $G = 16$, asymptoting at 0.631 on the measured sweep, a decline of 0.207 across that range[4]. Larger groups buy more

[2]Source: platform_tinker/reports/esa_phase1/Phase1_Project_Report_ZVF.tex.

[3]Source: platform_tinker/reports/esa_phase1/Phase1_Project_Report_ZVF.tex.

[4]Source: platform_hybrid/paper/sections/group_size.tex; source: platform_hybrid/paper/sections/zvf_iter122.tex.

contrast, with diminishing returns that saturate well before the group is small enough to be cheap.

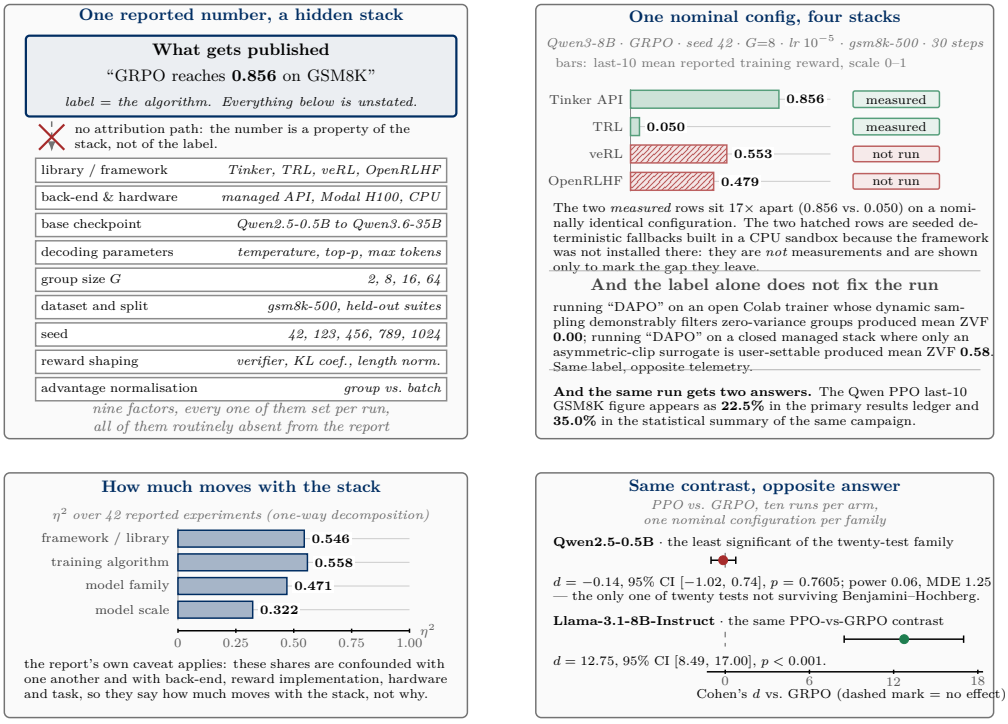
There is a second and less comfortable observation in the background, and it is about attribution rather than mechanism. "GRPO" is not a single algorithm. It is a label that covers implementations in Tinker, TRL, SkyRL, verl, OpenRLHF and a growing number of research forks, each of which chooses its own answer parser, sampling temperature, top-p, KL coefficient, clip range, LoRA target set, precision and evaluation harness. Two runs that both say "GRPO" in their titles may differ in a dozen ways that plausibly move the outcome, and the reported headline number does not distinguish between them. The project's own stack-sensitivity probe makes this concrete: a nominally matched configuration (Qwen3-8B, seed 42, $G = 8$, learning rate 1×10^{-5}, GSM8K-500, 30 steps, visible hyperparameters held equal) was run across four framework paths with results recorded in `experiments/results/framework_comparison.json`[5]. Four separate marginal one-way ANOVAs over 42 experiments give $\eta^2 = 0.546$ for library or framework, $\eta^2 = 0.558$ for training algorithm, $\eta^2 = 0.471$ for model family and $\eta^2 = 0.322$ for model scale — with the important caveat, stated by the source itself, that these factors are confounded with one another and with back-end, reward implementation, hardware and task[6]. Because each value is a marginal per-factor η^2 on a collinear corpus, the four sum to 1.897 and are not a variance decomposition; none estimates a factor's independent contribution (Chapter 4 §4.1; source: platform_hybrid/paper/sections/_shared_methods.tex). The framework factor is nonetheless not a nuisance parameter to be wished away: marginally, outcomes vary with it about as strongly as with the algorithm label.

1.2 A Motivating Example

Consider a single training step with a group size of $G = 8$ on an easy GSM8K prompt that the current policy already solves. All eight sampled completions produce the correct final answer, so each receives reward $r_i = 1$. The group mean is $\mu_g = 1$ and the group standard deviation is $\sigma_g = 0$, so every advantage $A_i = 0$. The compute spent on that group is real — eight full forward-decode passes at whatever temperature the sampler is configured with — and the group supplies no task-learning signal. The KL-regularisation term still nudges the policy toward the reference, but the point stands: no reward-driven learning happens on this group. Now imagine a batch of 16 such prompts in which 12 are already solved and 4 are

[5]Source: platform_hybrid/paper/main_eai_body.tex.
[6]Source: platform_hybrid/paper/main_eai_body.tex.

not: $12/16 = 0.75$ of the batch is "dark" compute[7]. What the Semester 4 work adds to this picture is that 0.75 is not a worst case. It is the neighbourhood of the measured typical regime, ZVF $\approx$ 0.72–0.77, and it is therefore the baseline against which any claim that GRPO is learning efficiently has to be read (Figure 1.2).



Why this motivates the thesis. An RL-for-LLM headline is a property of the whole stack that produced it, and the label — "GRPO", "DAPO" — names only one row of that stack. The evidence shown here is drawn from the project's own record: the η^2 shares and the four-framework spread come from the framework-comparison campaign on a single nominal configuration; the p-values, effect sizes and the two ledger figures come from the Phase-1 results and statistical summary. Three readings are load-bearing. First, the two measured frameworks differ by 17× while the algorithm label is held fixed — though the audit's own reading is that the managed stack had also pinned a different base checkpoint, so this is an under-specification exhibit rather than a clean back-end-only effect. Second, the same PPO-vs-GRPO contrast is null on one model family and enormous on another, so an algorithm claim does not transfer across families. Third, the Qwen arms did not share a back-end (GRPO on the managed API, PPO on a Modal H100 cluster), so the source withholds a directional Qwen algorithm claim and treats those rows as stack-conditioned evidence.

Figure 1.2: The confounding problem this work addresses. A single reported reward figure sits on top of an unstated stack; without fixing and reporting that stack the number is not attributable to the method under test.

Now suppose, having noticed this, that one wants to answer the obvious follow-up question: is the group-relative estimator actually a better choice than a learned critic? The natural experiment is a matched comparison — same model, same training prompts, same verifier, same step budget, same evaluation, with PPO and GRPO as the only difference. The project ran that contrast on Qwen3-8B and recorded the result in its own test family. The Welch t-test on PPO versus GRPO gives $p = 0.7605$, the least significant entry in the report's twenty-test family after Benjamini–Hochberg correction and the only one that does not survive it[8]; the methods chapter carries that Welch statistic as a provenance conflict rather than an estimate (Chapter 4 §4.1). The accompanying effect-size table carries

[7]Source: platform_tinker/reports/esa_phase1/Phase1_Project_Report_ZVF.tex.
[8]Source: platform_hybrid/paper/main_eai_body.tex; source: REPRODUCE.md.

Cohen's $d = -0.14$ with a 95% confidence interval of $[-1.02, 0.74]$, observed power 0.06 and a minimum detectable effect of 1.25 in the units of the outcome[9]. The unit behind those statistics must be read carefully: the $n = 10$ is the final ten steps of a single run per arm, not ten independent runs, and those steps are autocorrelated by construction. The project's own methods note is explicit that single-seed Tinker experiments have "zero statistical power", that "no inferential test can be computed and no confidence intervals can be formed" on them, and that such results are "treated as descriptive only and are not subject to significance testing"[10]. The interval and power figures are therefore reproduced here because they appear in the source's table, but they cannot bear the weight a confidence interval normally carries, and nothing in this section rests on them. Read carefully, that is a failure to detect a difference at this sample size, not a demonstration that the two estimators are equivalent; the confidence interval is wide enough to contain effects in both directions, and the study was in no position to rule out anything but a very large one.

Two further observations from the same study make the example sharper rather than cleaner. First, the same PPO-versus-GRPO contrast on Llama-3.1-8B-Instruct produces $d = 12.75$ with a 95% confidence interval of $[8.49, 17.00]$ and $p < 0.001$[11]. The apparent answer is not merely weak on Qwen; it reverses across model families. Second, the Qwen arms were not executed on a common back-end — the GRPO arm ran through Tinker-managed infrastructure and the PPO arm on a Modal H100 cluster — and the source states this explicitly and declines to draw the algorithmic conclusion it invites: "We therefore treat the PPO/GRPO rows as stack-conditioned evidence, not as an algorithm diagnostic baseline", and "We therefore withhold a directional Qwen algorithm claim"[12]. The same section records an aggregation gap that illustrates why such care is warranted: the Qwen PPO last-10 GSM8K figure is 22.5% in the primary results ledger but 35.0% in the statistical summary, and the source names the ledger value as primary and treats the summary value as evidence of an aggregation defect rather than as a second measurement[13].

The genuinely same-stack evidence in the repository points the same way, and it is worth stating plainly because it is a negative. The same-stack PPO/GRPO run record (Qwen2.5-0.5B, 40 steps, five paired seeds) gives a paired held-out difference of -0.002 with p = 0.374[14]. The E1 audit takes four GRPO-family

[9]Source: platform_hybrid/paper/main_eai_body.tex.
[10]Source: platform_hybrid/paper/sections/_shared_methods.tex.
[11]Source: platform_hybrid/paper/main_eai_body.tex.
[12]Source: platform_hybrid/paper/main_eai_body.tex.
[13]Source: platform_hybrid/paper/main_eai_body.tex.
[14]Source: platform_hybrid/paper/sections/frontier_synthesis_scaling.tex; source: plat-

variants (DAPO, GSPO, Dr. GRPO, AERO) plus the GRPO reference, runs five arms across eight paired seeds for 40 units at a frozen stack fingerprint, and evaluates on 500 held-out GSM8K examples per unit. Every one of the four contrasts is inconclusive: DAPO $+0.00100$, 95% CI $[-0.00450, +0.00675]$, $p = 0.756$; GSPO $+0.00500$, CI $[-0.00125, +0.01200]$, $p = 0.210$; Dr. GRPO -0.00200, CI $[-0.00950, +0.00725]$, $p = 0.673$; AERO -0.00075, CI $[-0.00825, +0.00675]$, $p = 0.858$. The pre-audit aggregator's verdict that the differences "disappear" is itself declared invalid, because the locked exact noncentral paired-t minimum detectable effect of 0.010115893 exceeds the 0.010 margin the study was designed around[15]. The audit's permitted wording is deliberately narrow: across eight paired seeds no arm produced a BH-significant held-out difference from GRPO, and none met the locked exact-power equivalence rule — which is not the same as saying the methods are the same, nor that they fail to improve.

Taken together, these two halves of the example are the whole motivation for what follows. The mechanism half shows that a large fraction of the compute in a GRPO run is not doing what a casual reading of the loss curve suggests it is doing. The attribution half shows that the most natural comparative question one would ask about that mechanism — is this estimator better than the alternative? — cannot be answered from the label "GRPO" and "PPO" alone, and that the honest answer on the project's own evidence is a non-significant, underpowered, back-end-confounded contrast rather than a ranking. Neither observation is a defect of the GRPO estimator; both are defects of what gets measured and what gets reported.

1.3 Problem Statement

The project is organised around four problems, in decreasing order of how often the literature acknowledges them.

Attribution is missing. Comparative claims about RL post-training are routinely made across frameworks that differ in parser, sampler, KL coefficient, clip range, LoRA targets, precision and evaluator. A reported gain from "GRPO" over "PPO" is, on most published evidence, a gain from one complete stack over another complete stack, with no way to say which component produced it. The project's own marginal per-factor η^2 values put the framework factor at a magnitude comparable to the algorithm factor, although they are not a partition of the

form_hybrid/experiments/results/samestack_ppo_grpo.json.

[15]Source: autoresearch/deli-neurips-tmlr-260802/audits/statistical_refutation.md.

variance[16], and its same-stack PPO/GRPO contrast returns a paired difference of -0.002 with $p = 0.374$[17]. The problem is not that these numbers are surprising; it is that no paper in this area is obliged to report the information that would let a reader separate them.

Signal starvation is invisible. Standard training logs report reward, loss, gradient norm, entropy and completion length. None of these surfaces the fraction of a batch whose group advantage is exactly zero. A run can show a rising reward curve and a falling ZVF, or a rising reward curve with a ZVF pinned near 0.8, and the two look identical in a loss plot. The fraction of sampling budget buying no reward-relative gradient is, in the project's own measurements, the largest measured source of zero-advantage sampling on this fixed setup[18], and it is not part of any standard telemetry.

There is no reporting or provenance standard. Even where authors are willing to describe their stack, there is no machine-readable form in which to do it, no validator that checks whether a claim's recorded configuration is internally consistent, and no accepted minimum set of fields that a stack-conditioned result must carry. Consequently, results cannot be pooled, cannot be compared across papers, and cannot be audited after the fact.

Evaluation results are rarely reported with their terminal states. Benchmark suites in the agentic and reasoning evaluation space are frequently multi-scope (a public split versus a private original contract, a benign-utility scope versus an adversarial scope), quota-limited, or gated behind private payloads. Reports tend either to omit such suites or to collapse them into a single number, which conceals how much of the suite was actually graded. The project treats this as a first-class concern rather than a presentational one: a result that is genuinely partial should be recorded as partial, with its coverage denominator and its reopen condition, and a lane that cannot be run at all should be recorded as externally blocked rather than quietly dropped.

1.4 Objectives

The project pursues eight objectives. They are stated as they were pursued, not retrofitted to the results.

O1 — Build a single harness that runs the same experiment across

[16]Source: platform_hybrid/paper/main_eai_body.tex; source: platform_hybrid/paper/sections/_shared_methods.tex.

[17]Source: platform_hybrid/experiments/results/samestack_ppo_grpo.json.

[18]Source: platform_tinker/reports/esa_phase1/Phase1_Project_Report_ZVF.tex.

multiple back-ends. Provide one control surface over Tinker, TRL, SkyRL, verl and OpenRLHF paths so that a configuration can be held fixed while the framework varies, and so that a framework difference is measurable rather than anecdotal.

O2 — Instrument the training loop so that signal starvation is directly observable. Emit per-step telemetry carrying ZVF, gradient utilisation, policy entropy, completion length and KL divergence, and persist the raw per-group reward tensors so that any derived statistic can be deterministically recomputed after the fact rather than trusted from a log line.

O3 — Measure the zero-variance regime on real reward tensors and test the levers that might move it. Characterise ZVF empirically on genuine GRPO reward tensors, and investigate re-baselining and curriculum interventions as ways of converting dark compute into contrast.

O4 — Establish and hold a scientific protocol. Multi-seed runs, matched baselines, held-out evaluation with explicit denominators, deterministic recomputation, pre-registered effect-size and power accounting, and honest reporting of negatives and nulls — including the cases where the answer is that the study was underpowered to answer.

O5 — Scope the most original direction in the field. A literature scan for white space, which in this project identified stack provenance and reporting standards as the unclaimed ground and produced the minimum reporting standard (P5) and the machine-readable stack registry (P6).

O6 — Turn the diagnostic into an intervention. Derive an adaptive group-size controller from the ZVF measurement and evaluate it as a training-time policy rather than a post-hoc observation (P7).

O7 — Evaluate a trained actor against held-out suites under strict governance. Run a single trained checkpoint against fourteen held-out agentic and reasoning suites, each with its own native evaluator at a pinned revision, under receipt, budget and decontamination controls, and record every lane's terminal state.

O8 — Report under a deliberate asymmetry. Where the evidence supports a structural or descriptive claim, state it; where it does not, withhold it. No cross-suite aggregate, no pooling of original-contract and replacement-scope numbers, and no claim of improvement over any baseline anywhere in the work.

1.5 Scope and Non-Goals

The scope of this report is measurement and diagnostics, not leaderboard performance. The controlled studies and the campaign actor use small-to-mid open models — the Qwen2.5, Qwen3, Qwen3.5 and Qwen3.6 families, from roughly 0.5B parameters up to the 35B sparse mixture-of-experts configuration used for the campaign actor. The P1 cross-scale roster is wider, 0.6B to 1T total parameters, but its frontier-scale anchors are single-seed runs through a managed API and are read descriptively (Chapter 6 §6.1; source: platform_hybrid/paper/sections/p1_abstract.tex). Verifiable rewards come from mathematics and code checkers, principally GSM8K exact-match and HumanEval-style unit tests. Held-out evaluation uses suites' own native evaluators at pinned revisions.

Several things are explicitly out of scope. This is not an attempt to set a state-of-the-art result on any benchmark, and no such claim is made anywhere in the report. It is not a study of frontier reasoning models. The held-out sets used in the controlled comparisons are intentionally small — 200 to 500 prompts in most cases — and every denominator is reported so that a reader can see how little is being claimed. Claims about novel algorithms are not made: the length-weighting intervention (P4) and the adaptive controller (P7) were designed during the work and are reported as designed and simulated systems with their evaluation limits attached, not as demonstrated improvements.

Three evidentiary rules apply throughout and are worth stating here because they constrain what the later chapters are allowed to say. First, every number in this report is traceable to a repository file that was actually read; where a figure could not be sourced, the finding is described qualitatively and the figure is attributed to its source rather than reconstructed. Second, original-contract and replacement-scope benchmark results are never pooled and never averaged across suites — the two are different measurements of different things under different licences, and treating them as one column would manufacture a number that no suite reported. Third, no improvement over a baseline or over a base model is claimed unless the source states it with a test; where the source reports a difference without a test supporting it, or reports an inconclusive test, that is what is written here.

1.6 Contribution of the Candidate

This report covers work that spans two semesters and two very different modes of authorship, and the distinction matters for how the contribution should be

assessed. Semester 3 was a group project (Group 6). The multi-framework repository, the baseline integrations, the initial Tinker, SkyRL and GRPO experiments, the literature foundation, and the early benchmark, evaluation and reproducibility scaffolding were produced by that group, of which the candidate was a member. Semester 4 is the candidate's individual continuation, and the new research direction, the experiment expansion, the analysis, the paper writing and the P1–P8 outputs are the candidate's solo work[19] (Figure 1.3).

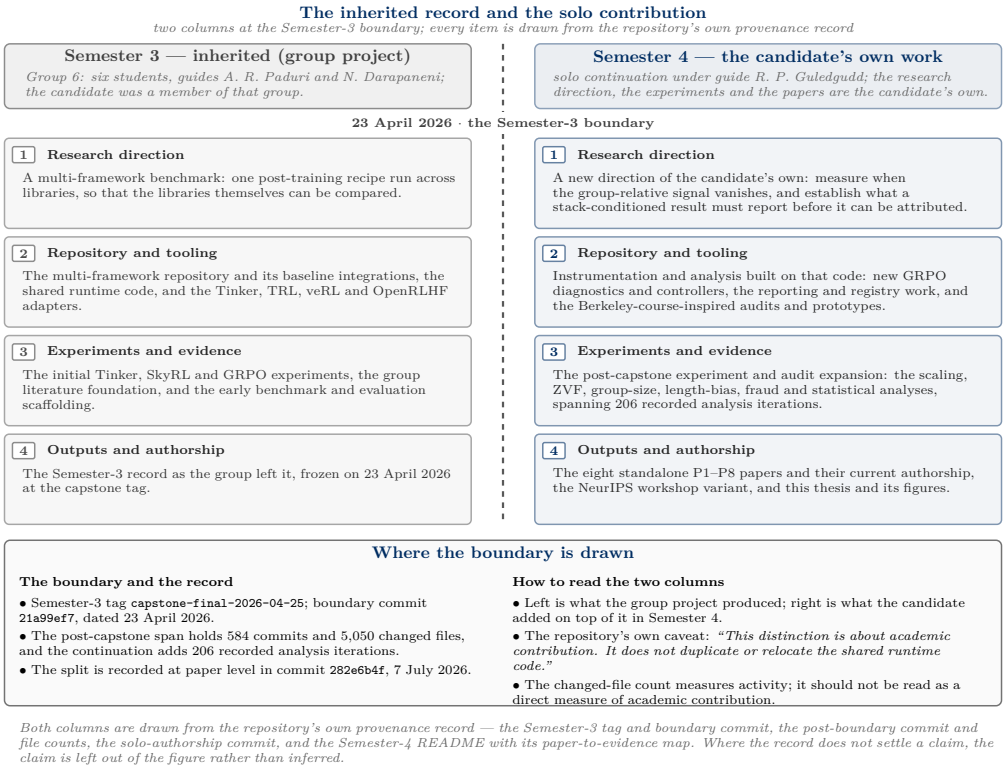


Figure 1.3: Boundary between the inherited Semester-3 group infrastructure and the individual Semester-4 contribution reported in this document.

The boundary is recorded precisely rather than loosely. The inherited state is frozen at the Git tag `capstone-final-2026-04-25`, corresponding to boundary commit `21a99ef766543a1ca86bfbad188445d3596ea73c` of 23 April 2026. At the time the separation was documented, 584 commits had been made after that boundary, touching 5,050 changed files — a figure the provenance record explicitly declines to present as a measure of academic contribution, since it counts mechanical and generated changes alongside substantive ones. The transition to explicit solo authorship is marked by commit `282e6b4f0f158de11b3f5bb4656e2d871f68f5c0` of 7 July 2026[20].

[19]Source: platform_hybrid/sem 4 work/EXPERIMENTS.md; source: platform_hybrid/sem 4 work/README.md.

[20]Source: platform_hybrid/sem 4 work/PROVENANCE.md.

What was added after that boundary is the substance of this report. It includes the eight standalone research papers: P1, a cross-library and cross-scale study of GRPO behaviour; P2, the Zero-Variance Fraction as a descriptive diagnostic; P3, group size, contrast density and the relationship to DPO; P4, length bias and held-out generalisation; P5, the minimum reporting standard for stack-conditioned results; P6, a machine-readable GRPO stack registry; P7, an adaptive group-size controller derived from the signal-starvation diagnostic; and P8, a side study comparing a language model against gradient-boosted trees in credit-card fraud detection[21]. It includes the post-capstone experimental, audit, synthesis and paper iterations — 206 recorded analysis iterations — the Berkeley-course-derived research audits and prototypes, the expanded statistical analyses, and the new GRPO diagnostics and controllers[22]. And it includes the E1–E14 held-out benchmark campaign, in which a single Tinker-trained actor (base `Qwen/Qwen3.6-35B-A3B` at commit `995ad96e…`, with the `arvindcr4/pavlov-portfolio-qwen36-seed809-stepfinal-tinker-cf0ad8c1…` adapter at commit `64444133…`, in bfloat16) was evaluated against fourteen suites under receipt, budget and decontamination governance. Three of those replacement scopes are strictly complete — LAB-Bench public at 1,967/1,967, AgentDojo benign utility at 97/97, and VerilogEval at 129/312 = 41.35% pass@1 — and Omni-MATH is terminal-complete under its native protocol, with all 4,428 dispositions recorded, 4,426/4,428 accepted, and its official accuracy of 2,271/4,428 = 51.31% reproduced. SWE-bench Pro carries a full verified original-contract score of 2/731 = 0.274% pass@1. The remaining lanes are recorded honestly as partial or externally blocked, each with a terminal state and a named reopen condition, and the structural claims were re-verified by deterministic code checks on 19 September 2026 with 11 of 11 checks passing[23].

Two further contributions deserve naming because they are contributions of restraint rather than of result. The first is that the campaign's negative findings were kept rather than discarded. When BankerToolBench was re-run on the base model under a separate authorisation — 100/100 trials, 3 hours 29 minutes, $15.07 charged — the mean reward was 0.0 across every trial, and the audit identified the mechanism as tool-dialogue collapse: agents terminated after 2 to 14 steps with no deliverables, and 27 or more trajectories ended in role-token degeneration of the form `user user assistant …`. That result is reported as a measurement of the base model's inability to sustain an open-ended tool-use loop, not as a measurement of finance reasoning, and it is retained in the record[24]. The second

[21]Source: platform_hybrid/sem 4 work/README.md.

[22]Source: platform_hybrid/sem 4 work/README.md; source: platform_hybrid/sem 4 work/EXPERIMENTS.md.

[23]Source: outputs/E1_E14_FINAL_RESULTS_2026-09-19.md.

[24]Source: outputs/E1_E14_FINAL_RESULTS_2026-09-19.md.

is the discipline applied to the project's own claims: the same-stack audit that invalidated the pre-audit aggregator's "disappears" verdict did so on the strength of the study's own locked minimum detectable effect, and the permitted wording it left in place is narrower than the finding it refuted[25].

The report does not reallocate the shared runtime code. The inherited repository remains where it was; the Semester 4 papers cite their canonical LaTeX sources and evidence directories rather than duplicating them, and the distinction recorded here is a distinction about academic contribution, not about file ownership[26].

1.7 Organisation of the Report

The report is organised in four parts.

Part I — Foundations contains the present chapter, followed by Chapter 2, the literature survey, which situates the work against the RLHF and RLVR literature, the critic-free line of estimators, the group-size and advantage-estimation literature, and the reproducibility and reporting literature on which the provenance argument depends; and Chapter 3, the system requirements specification, which states the functional and non-functional requirements the harness, the telemetry layer and the campaign infrastructure were built to satisfy.

Part II — Design and Implementation contains Chapter 4, the proposed methodology, which sets out the measurement design, the statistical protocol, the evidence rules and the study-to-evidence mapping; and Chapter 5, the implementation details, covering the multi-framework harness, the per-step telemetry and raw tensor persistence, the campaign launcher and its receipt and budget governance, and the registry and validator components.

Part III — Results contains four chapters. Chapter 6 reports the core GRPO studies — the cross-library and cross-scale scaling work, the ZVF diagnostic, the group-size and contrast-density sweep, and the length-bias and held-out generalisation study, corresponding to P1 through P4. Chapter 7 reports the reporting, registry and controller work, corresponding to P5 through P7, including the minimum reporting standard, the machine-readable stack registry, and the simulated behaviour of the adaptive group-size controller. Chapter 8 reports the applied fraud-detection study, corresponding to the archived P8. Chapter 9 reports the E1–E14 held-out benchmark campaign: the actor, the governance model, the strictly completed scopes, the original-contract scores, the incomplete and blocked

[25]Source: autoresearch/deli-neurips-tmlr-260802/audits/statistical_refutation.md.
[26]Source: platform_hybrid/sem 4 work/EXPERIMENTS.md.

lanes, the verification results and the campaign's terminal states.

Part IV — Synthesis contains Chapter 10, which draws the cross-study synthesis, states the threats to validity candidly, gives the conclusions the evidence will bear, and sets out future work and the reopen conditions under which the blocked lanes could be completed.

Three appendices follow. Appendix A gives the complete run registry. Appendix B documents reproducibility and provenance, including the inherited-versus-individual boundary and the environment records needed to reproduce the reported results. Appendix C gives the notation, symbols and abbreviations used throughout.

Chapter 2

Literature Survey

This chapter surveys the prior work on which the project is built and the gap it occupies. It revises the survey of the Phase-1 report[1], retaining that survey's seven-part structure and updating each part with the material the Semester 4 work added: the P1–P8 paper series and the E1–E14 held-out evaluation campaign. Two rules of construction carry over from Phase 1. The first concerns proportion. The contribution of this project is overwhelmingly a contribution of measurement and reporting practice rather than of a new training algorithm, so the literature that establishes measurement practice receives more room here than the literature that establishes methods. The second concerns evidence. A figure is quoted only where a repository source actually states it; where a source states a finding without giving a number, the finding is described qualitatively and the chapter says so. Where the project's own documents state the same quantity differently, the divergence is reported rather than silently reconciled, and Section 2.5 records one such divergence in full.

2.1 From RLHF and PPO to Group-Relative Methods

Policy-gradient post-training of language models begins with the RLHF pipeline. Christiano et al. [1] established the three-stage pattern of supervised fine-tuning, reward-model training on human pairwise preferences, and reinforcement learning against that reward model; Ziegler et al. [2] adapted it to language, collecting approximately 60,000 comparisons and introducing the per-token KL penalty against a frozen reference policy that has been standard ever since. Stiennon et al. [3]

[1] Source: `platform_tinker/reports/esa_phase1/Phase1_Project_Report_ZVF.tex`.

scaled the recipe to a 6.7B summarisation model with 64,832 comparisons, demonstrated transfer to an out-of-distribution summarisation set (CNN/DailyMail), and produced the first empirical evidence of reward over-optimisation: continued optimisation against a frozen reward model eventually degrades the true objective. Ouyang et al. [4] then built InstructGPT, whose three-stage pipeline used roughly 40 contractors to produce about 13,000 supervised fine-tuning prompts, about 33,000 reward-model prompts, and about 31,000 PPO prompts. Two details of that work matter for this project. First, the reward model was 6B rather than 175B because the larger reward model proved unstable during training — a reminder that the reward model is itself a trained component with its own failure modes. Second, the preference data were collected as rankings of $K \in \{4, \ldots, 9\}$ responses, which yields $C(K, 2)$ pairwise comparisons from a single annotation batch; this is precisely the within-prompt comparative structure that group-relative methods would later exploit directly. InstructGPT's headline results were that a 1.3B InstructGPT was preferred to the 175B GPT-3 base model, that the 175B InstructGPT was preferred to GPT-3 in 85% of evaluations, that hallucination on a closed-domain task fell from 41% to 21%, that toxicity was reduced by roughly 25%, and that TruthfulQA performance roughly doubled; inter-annotator agreement was approximately 73%. The same source records four limitations — annotation cost, reward-model over-optimisation, the difficulty of aligning to user intent rather than to annotator preference, and the residual capability costs of alignment[2] (Figure 2.1).

The optimiser for that pipeline was PPO. PPO's clipping mechanism, with $\varepsilon \approx 0.2$, restricts each update's policy ratio, and generalised advantage estimation with its standard defaults ($\lambda = 0.95$, $\gamma = 0.99$) provides the advantage estimate. Recasting language generation as an RL problem makes the state the prompt plus generated prefix, the action the next token, the policy the language model, and the reward sparse and terminal. The practical consequence is a four-model architecture — policy, reference, reward, and value — whose mixed-precision memory footprint is roughly four times that of supervised fine-tuning alone; PPO-ptx mitigates capability regression by mixing pretraining gradients into the update. Zheng et al.'s "Secrets of RLHF" study found that policy constraints, not algorithmic choice, are the single most critical factor in whether RLHF succeeds: without them, training exhibits reward hacking, length explosion, and catastrophic pattern collapse[3]. That finding is the intellectual ancestor of the zero-variance analysis in this project, because it locates the failure mode in the constraint structure rather than in the objective.

[2]Source: `capstone-literature-survey/section_2_2_rlhf.tex`.
[3]Source: `capstone-literature-survey/section_2_3_ppo.tex`.

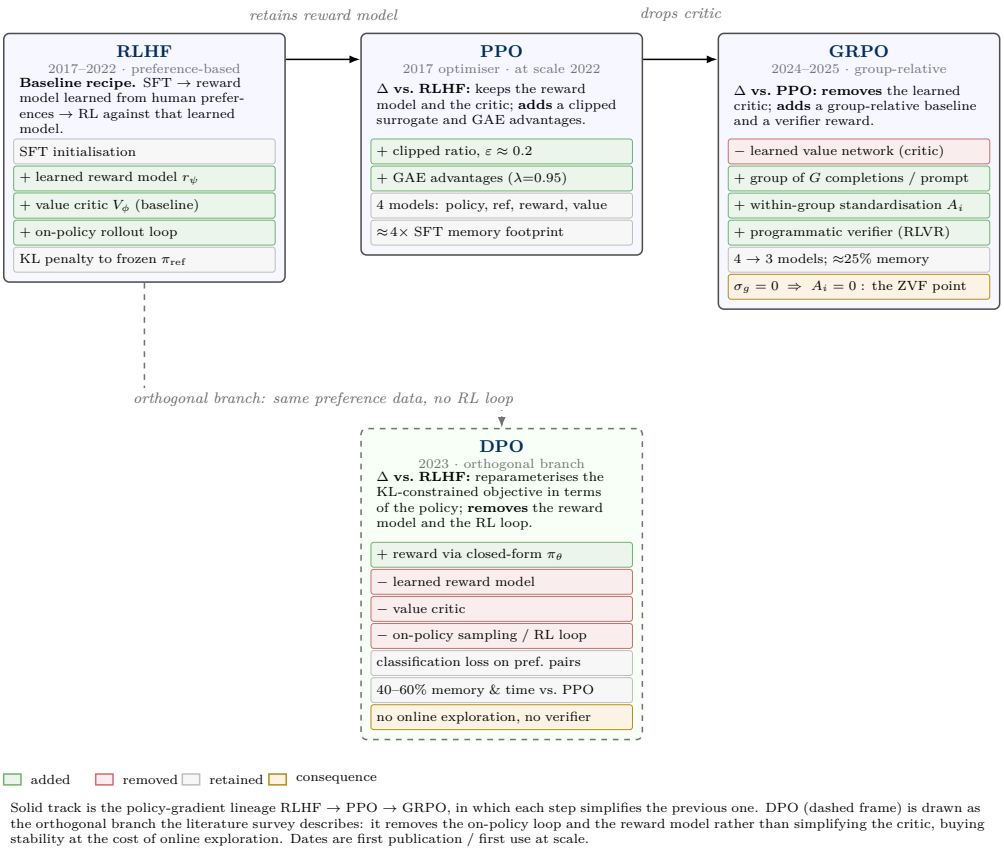


Figure 2.1: Algorithmic lineage from RLHF through PPO and DPO to GRPO and its variants. Each transition removes a component or substitutes a cheaper estimator; GRPO removes the critic and replaces it with a group-relative baseline.

Group Relative Policy Optimization (GRPO), introduced with DeepSeekMath [5], removes the learned value network altogether. For each prompt it samples a group of G completions, scores them with a reward function, and standardises each completion's reward against the group mean and standard deviation; the resulting z-score serves as the advantage for every token of that completion. In the DeepSeekMath configuration $G = 64$, $\beta = 0.04$, and $\varepsilon = 0.2$. Eliminating the value model reduces the number of trained and served models from four to three and cuts memory requirements by approximately 25%. DeepSeekMath-RL 7B reached 51.7% top-1 accuracy on MATH against a 46.8% supervised fine-tuning baseline, 60.9% with self-consistency over 64 samples, GSM8K improved from 82.9% to 88.2%, and CMATH improved from 84.6% to 88.8% despite no Chinese data in the RL training set[4]. The paper's iterative variant retains 10% of historical data in a replay buffer to prevent catastrophic forgetting in the co-evolving reward model. RLOO [6] is the closely related leave-one-out estimator that baselines each sample against the mean of the *other* samples in the group; it already performs a within-group baseline, and is therefore directly relevant to the signal-starvation analysis this project conducts[5].

DeepSeek-R1 [7] then demonstrated that pure outcome-reward RL can elicit chain-of-thought behaviour from a base model without supervised cold-start data. R1-Zero was trained on DeepSeek-V3-Base, a 671B mixture-of-experts model, using rule-based rewards only and no neural reward model. AIME 2024 pass@1 rose from 15.6% to 71.0% over training — the revised *Nature* version reports 77.9% after extended training — with consensus decoding over 64 samples (cons@64) reaching 86.7% and MATH-500 reaching 95.9%. The source states the training configuration as 16 sampled outputs per question, a group size of 512, a learning rate of 3×10^{-6}, a KL coefficient of 0.001, and approximately 10,400 policy update steps. The paper documents spontaneous chain-of-thought, self-verification, and the widely reported "aha moment" as emergent rather than taught behaviours. The full R1 model, trained with a four-stage pipeline that includes a language-consistency reward, reports 79.8 on AIME, 97.3 on MATH-500, and 71.5 on GPQA, against 79.2 / 96.4 / 75.7 for o1-1217. Distillation into six students from 1.5B to 70B produced Distill-Qwen-1.5B at 28.9% AIME and 83.9% MATH-500, and Distill-Qwen-32B at 72.6% and 94.3%, against 47.0% and 91.6% for Qwen2.5-32B-Zero trained from scratch; the 7B and 16B base models failed to show meaningful reasoning gains from direct RL[6].

For this project, the significance of that lineage is narrower than the headline

[4]Source: `capstone-literature-survey/section_2_5_grpo.tex`.
[5]Source: `platform_tinker/reports/esa_phase1/Phase1_Project_Report_ZVF.tex`.
[6]Source: `capstone-literature-survey/section_2_6_deepseek_r1.tex`.

numbers suggest. GRPO's defining operation is a within-group standardisation, and a standardisation is undefined when the group's rewards are all identical. That degenerate case is not an edge case: it is the modal case early in training on a hard task and late in training on an easy one. The rest of this chapter and the empirical chapters that follow are concerned with what that degeneracy does to the training signal, and with how honestly the field currently reports it.

2.2 Direct Preference Optimisation as a Contrast

An orthogonal branch of post-training removes on-policy sampling entirely. Direct Preference Optimisation (DPO) [8] exploits the fact that the RLHF objective with a KL constraint admits a closed-form optimal policy, which permits the reward to be reparameterised in terms of the policy itself. The result is a simple classification loss over a fixed dataset of preference pairs, with no reward model, no value model, and no sampling loop. Reported savings are 40–60% in both GPU memory and wall-clock time relative to PPO. Four limitations are documented in the same source: the training data are static and offline, so there is no online exploration; the loss can displace likelihood on responses outside the preference pairs; and the method has weaker guarantees for long-horizon alignment, where credit must propagate across many steps[7].

The contrast is structurally instructive for this project. DPO cannot suffer zero-variance collapse, because it never standardises within a sampled group — its contrast is supplied by the dataset rather than constructed at training time. But it equally cannot exploit a verifier that is available in the loop, which is the property that makes RLVR attractive on mathematics. GRPO's zero-variance pathology is therefore the price of its on-policy, verifier-in-the-loop design, and measuring that price is exactly what makes the pathology worth quantifying. The DPO family has since diversified into inverse-preference and non-paired variants — IPO, KTO, ORPO, SimPO, and SLiC among them — which the project's cross-library protocol treats as no-rollout baselines[8].

The boundary between the two families is also less sharp than it first appears. Wu et al. prove that GRPO is "secretly DPO" — the group-level advantage constitutes an implicit contrastive objective — and report that 2-GRPO retains 98.1% of the

[7]Source: `capstone-literature-survey/section_2_4_dpo.tex`.

[8]Sources: `capstone-literature-survey/section_2_4_dpo.tex`; `platform_hybrid/paper/sections/related_work_v2.tex`.

performance of 16-GRPO at 12.5% of the rollout budget[9]. That reduction is the theoretical counterpart of the PCD statistic this project develops: if a group is exactly as informative as the correct/incorrect pairs it contains, then the pairwise-contrast density $\mathrm{PCD} = ((G - 1)/G) \cdot E[p(1 - p)]$ is the natural magnitude-aware summary of a group's usable signal, and the group-size question becomes a question about how much contrast a given sampling budget buys[10]. The present project is careful about what it inherits from that argument: the reduction is cited as prior theory, while the empirical claims resting on it are scoped to this project's own measurements, reported in Chapter 6.

2.3 Fixing GRPO's Biases and Collapse

A dense and fast-growing literature targets specific GRPO pathologies. Dr. GRPO [9] removes two normalisation terms from the standard objective — the per-response 1/L factor and the question-difficulty amplification — restoring the sequence-level gradient up to a global scale, and is motivated explicitly as a fix for reward hacking through length[11]. DAPO [10], from ByteDance, changes four things in one release: asymmetric Clip-Higher clipping, dynamic sampling that resamples until the batch carries signal, token-level loss normalisation, and over-long filtering; the reported result is 50 points on AIME 2024 with a 32B model[12]. GSPO (Qwen team, 2025) moves the importance ratio from the token level to the sequence level and was credited with stabilising the Qwen3 series, particularly its mixture-of-experts variants. Production-side variants include VAPO and Dr. SAC, the latter arguing that carefully implemented REINFORCE with a value baseline matches GRPO on several RLHF tasks[13].

A third thread attacks the zero-variance failure mode directly, and this is the thread the project's own diagnostic sits in. GRESO pre-screens prompts before sampling so that uninformative prompts are never paid for. CPPO prunes completions after rollout, reporting up to 7.98× speedup on GSM8K. NGRPO adds advantage calibration and asymmetric clipping so that all-incorrect groups still carry gradient. Scaf-GRPO injects tiered scaffolding hints when learning stagnates, reporting a 44.3% relative improvement on AIME 2024. MO-GRPO addresses vanilla GRPO's collapse to a single reward component under multiple objectives

[9]Source: `platform_hybrid/paper/sections/related_work_v2.tex`.
[10]Source: `platform_hybrid/paper/sections/p7_intro.tex`.
[11]Sources: `platform_hybrid/paper/sections/p4_intro.tex`, `platform_hybrid/paper/sections/p6_intro.tex`.
[12]Source: `platform_hybrid/paper/sections/related_work_v2.tex`.
[13]Source: `platform_hybrid/paper/sections/related_work_v2.tex`.

through variance-based reward re-weighting. SSPO operates at sentence granularity to balance GSPO's stability against token-level GRPO's expressiveness, improving by 3.56 points on mathematics benchmarks. AERO / RL-ZVP takes the opposite approach to filtering: it extracts signal *from* zero-variance groups through entropy-guided advantage shaping[14]. GVPO re-weights by within-group variance[15]. The tail continues with λ-GRPO, MC-GRPO (median-centred advantage), G^2RPO, ConSPO, and Goldilocks RL.

One act of bookkeeping in this literature deserves emphasis, because it is a claim about what may *not* be inferred. AVSPO, query recycling, ConSPO, and Goldilocks RL are described in the project's own related-work source as prior or concurrent interventions, not as evidence for this project's runner, and the paper explicitly claims neither priority nor an intervention effect over them[16]. The present thesis inherits that disclaimer without qualification.

Against this background, the project's own contribution to the area is deliberately meagre in algorithmic ambition and deliberately demanding in evidential standard. The Zero-Variance Fraction (ZVF) — the fraction of groups whose completions are all correct or all incorrect — was measured across libraries, model scales, and training phases in the companion ZVF paper, and that paper stopped at description: ZVF tracks collapse, drifts upward, and is mechanically entangled with accuracy and group size, so it declined to promote ZVF to a predictive or causal statistic[17]. A minimal binomial model of group degeneracy was then proposed, in which the expected ZVF indicator is $h_G(p) = p\hat{}G + (1-p)\hat{}G$ and the usable signal is $S = p(1-p)(1-h_G(p))$; three regularities follow in closed form, each paired with an empirical check drawn from a 368-run Weights & Biases audit that counts all logged runs and is therefore a superset of the benchmark's 70+ core training runs. Because $h_G(p)$ is symmetric under $p \leftrightarrow 1-p$, ZVF assigns the same value to a mastered prompt and to a hopeless one, producing the U-shape measured across seven models; and because ZVF is an existence statistic, adding reward jitter of order 10^{-4} drives measured ZVF to zero while shifting the measured within-group contrast by less than 10^{-6}. The pairwise-contrast density is proposed as the magnitude-aware replacement, and the toy-scale validation reports a gradient-norm correlation of corr $= +0.71$, marked directional only. The word *controller* in that work names the implemented policy, not an established improvement: the decisive static-G = 16 matched-token bakeoff remains unrun[18].

[14]Sources: `platform_hybrid/paper/sections/related_work_v2.tex`, `platform_hybrid/paper/sections/p7_intro.tex`.
[15]Source: `platform_tinker/reports/esa_phase1/Phase1_Project_Report_ZVF.tex`.
[16]Source: `platform_hybrid/paper/sections/related_work_v2.tex`.
[17]Source: `platform_hybrid/paper/sections/p7_intro.tex`.
[18]Source: `platform_hybrid/paper/sections/p7_intro.tex`.

The one genuinely measured GRPO-variant comparison in the repository is a Dr. GRPO versus GRPO control. On Qwen2.5-0.5B at group size 8, 40 steps, and 5 seeds, held-out accuracy was indistinguishable between arms: GRPO 0.987 $\pm$ 0.003 against Dr. GRPO 0.992 $\pm$ 0.003, paired difference +0.5 percentage points, $t = 1.05$, $df = 4$, $p = 0.35$; mean completion length was 4.74 against 4.71 tokens. In a second regime — Qwen2.5-1.5B-Instruct on GSM8K with a 200-token chain-of-thought cap and 3 seeds — GRPO moved from 20.2% to 26.3% and Dr. GRPO from 20.5% to 25.5%; per-seed McNemar tests were significant in 2 of 3 seeds for GRPO ($p = 0.071, 0.036, 0.016$) and 1 of 3 for Dr. GRPO ($p = 0.122, 0.017, 0.122$), and the pooled $p < 10^{-3}$ is expressly rejected as overstating significance. Neither arm inflated length over 30 steps (GRPO 195 $\rightarrow$ 180 tokens; Dr. GRPO 194 $\rightarrow$ 187). The same source contains head-to-head and rank-correlation tables whose values are *projections*, not measurements: the GRPO baseline row is a synthetic baseline produced by a dry run of `platform_hybrid/experiments/variance_mitigation_integration.py`, the file states that it makes no measurement claim for any row of that table, the correlation coefficients (0.78 / 0.66 / 0.63 / 0.31) carry the same caveat, and the configuration proxies under `platform_local/unified/` are explicitly not faithful reimplementations. Those numbers are reported here only to mark them as unusable as evidence[19]. The nearest thing to a headline cross-library effect in the project is also the one that must be read most carefully: holding the visible group size, learning rate, data, seed, and label fixed while changing stacks moved final GSM8K reward from 85.6% to 5.0%, a 17$\times$ swing; the audit that followed found the managed stack had also pinned a different base checkpoint, so the source describes this as an under-specification exhibit rather than a backend-only causal effect[20]. The complementary telemetry fact is that running "DAPO" on an open Colab trainer whose dynamic sampling demonstrably filters zero-variance groups produced mean ZVF 0.00, while running "DAPO" on a closed managed stack where only an asymmetric-clip surrogate is user-settable produced mean ZVF 0.58 — the same label, opposite telemetry[21].

[19]Sources: `platform_hybrid/paper/sections/variance_mitigation_comparison.tex`; `platform_hybrid/experiments/results/drgrpo_vs_grpo.json`; `platform_hybrid/experiments/results/drgrpo_gsm8k_cot.json`.

[20]Source: `platform_hybrid/paper/sections/p6_intro.tex`.

[21]Source: `platform_hybrid/paper/sections/p6_intro.tex`.

2.4 Verifiable Rewards, Frameworks and Reporting Practice

RLVR — reinforcement learning from verifiable rewards — was formalised for open post-training in TÜLU 3 [11], which is the source this project follows in treating a programmatic checker rather than a learned preference model as the reward[22]. The implementation landscape the project unifies is crowded and heterogeneous. On the training side it includes TRL, OpenRLHF, veRL (HybridFlow), NeMo-RL, DeepSpeed-Chat, trlX, Axolotl, and Tinker as a managed back-end; each makes different default choices about advantage normalisation, KL estimators, tokenisation loss masking, rollout batching, and importance-sampling granularity, and the project identifies those differences as the principal source of the cross-library gap it quantifies. On the inference side, vLLM and SGLang supply the rollout backends on which modern frameworks depend, with FlashAttention and LoRA as load-bearing kernels[23]. Tinker is adopted here through its `message_with_tool` interface as the neutral substrate for tool-use extensions, and the same source explicitly cites tool-use and agentic RL settings as concurrent environments in which this project's ZVF measurements will need to be re-evaluated rather than assumed to transfer[24].

The evaluation benchmarks are equally well established and equally awkward. GSM8K comprises 8,500 problems split 7,473 / 1,319, on which a fine-tuned GPT-3 verifier reached 35% at release while current systems exceed 95%; GSM8K-Platinum has since documented label noise in the original set, which is why the source describes the benchmark as necessary but not sufficient. MATH comprises 12,500 problems across seven subjects and five difficulty levels, on which the original results reached 6.9%; MATH-500 is the 500-problem subset introduced with the Lightman process-supervision work, on which current systems sit at roughly 99%. AIME comprises 15 problems per year with integer answers in 0–999; reported AIME 2024 accuracies include GPT-5 at 95.7% and Grok 4 at 94.3%. Because these benchmarks are close to saturation, the source frames chain-of-thought as the action space that makes them interesting at all, and identifies process reward models as the bridge from terminal outcome signals to step-level credit[25].

Reporting practice is where the project's own contribution is anchored, and its

[22]Source: `platform_tinker/reports/esa_phase1/Phase1_Project_Report_ZVF.tex`.

[23]Sources: `platform_hybrid/paper/sections/related_work_v2.tex`; `platform_tinker/reports/esa_phase1/Phase1_Project_Report_ZVF.tex`.

[24]Source: `platform_hybrid/paper/sections/related_work_v2.tex`.

[25]Source: `capstone-literature-survey/section_2_8_mathematical_reasoning_benchmarks.tex`.

lineage is explicit. Model Cards [12] introduced documentation of intended use, out-of-scope use, and per-slice evaluation results; Datasheets for Datasets [13] extended the idea to datasets, covering composition, preprocessing, annotation process, and discriminatory-impact mitigation; Data Statements [14] added the speaker-demographic axis; and Data Cards [15] bridged research and industry practice with schema, provenance, sampling-distribution, and audit-transparency fields. A cross-coupling audit at iteration 109 maps these four works onto the project's legacy 18-field experimental manifest, showing that they cover 12 distinct fields, that the remaining six are either RL-specific (per-step ZVF, group-size schedule) or stack-coverage items with no direct analogue in the prior literature, and that items 9–12 — annotation process and annotator demographics — have the lowest evidence density in the live manifests. The same iteration verified all four entries against CrossRef metadata (4/4 pass on year, five-gram title, and author-family overlap ≥ 0.6) and retrofitted two entries that had been present in the bibliography but never cited[26]. The resulting standard, MIN-REPORT-RL, is an eight-item list aligned at eight items on 2026-07-11, whose eighth item (held-out pass@k) is an evaluation-report field rather than run-start stack identity[27].

The reproducibility literature to which this standard is a deliberate echo is older and better established. Henderson et al. [16] showed that deep-RL results depend on codebase, hyperparameters, and even the choice of random seeds, to the point that the same algorithm from two repositories behaves as two algorithms. Colas et al. [17] and Agarwal et al. [18] (`rliable`) supplied the statistical repair kit — rigorous comparison protocols and small-sample-aware aggregate metrics with interquartile-mean reporting and performance profiles. Jordan et al. [19] argued that most published RL comparisons lack statistical power. Dodge et al. [20] proposed reporting expected validation performance as a function of compute budget; the project's run manifest is the RL-for-LLM analogue, replacing *budget* with *stack*. Pineau et al. [21] formalised the NeurIPS reproducibility criteria, and the project draws the explicit lesson that self-attested checklists have limits, which motivates an artifact-computed badge rather than a declared one[28]. Statistical treatments of evaluation noise (Miller et al.; Zhang et al.) quantify the uncertainty floor against which the project's own within-noise findings should be read, and the evaluation-stack sensitivity documented by the LM Evaluation Harness effort, by

[26]Sources: `platform_hybrid/paper/sections/p5_related.tex`; `platform_hybrid/experiments/results/p5p8/p5_iter109_minreport_coupling.tsv`; `platform_hybrid/experiments/results/p5p8/p5_iter109_crossref_verify.tsv`; `platform_hybrid/experiments/results/p5p8/p5_iter109_bib_audit.tsv`.

[27]Source: `platform_hybrid/paper/sections/p5_related.tex`, `platform_hybrid/paper/sections/p6_intro.tex`.

[28]Sources: `platform_hybrid/paper/sections/p5_related.tex`; `platform_hybrid/paper/sections/related_work_v2.tex`.

audits of reasoning evaluations, and by contamination studies of the data side of the same coin, together establish that RL *training* inherits all of this — the verifier is an evaluator in the loop — plus the additional vectors unique to on-policy optimisation[29].

Three existing resources come closest to the project's registry and are the reason it exists in its present form. HELM aggregates evaluation across a large model and scenario matrix, but its unit is the (model, scenario) cell, not the training stack. The LM Evaluation Harness standardises evaluation by *executing* implementations, whereas the registry here standardises by *describing* stacks it cannot execute — many of them closed — which is precisely why the project required a null-versus-absent convention and an R5 rung that records un-auditability rather than converting it into a false value. Papers with Code catalogued paper–code–leaderboard linkage at scale, stopped at the repository boundary, and was discontinued in July 2025; the source treats its fate as a governance cautionary tale. The project's own claim to novelty is narrow and stated negatively: no existing resource provides the executed loss form at GRPO-variant granularity, per-component implementation status, or an unauditability marker[30].

2.5 Scaling and Layer-Adaptation Context

Compute-optimal pretraining is governed by well-characterised scaling laws (Kaplan et al.; Hoffmann et al., Chinchilla), but RL post-training is much less well characterised, and the sources the project builds on separate cleanly into those that predict and those that describe. Hilton et al. [22] derived compute-efficiency frontiers for RL in games and robotics. For RLHF, Gao et al. [23] characterised the over-optimisation curve of reward models — the point at which additional optimisation against a frozen reward model buys a decreasing true objective — and Rafailov et al. [24] provided scaling laws for DPO and PPO that show a strong dependence on reward-model quality. Nimmaturi et al. [25] derived an empirical scaling law specifically for GRPO, identifying three training phases (slow start, rapid improvement, plateau) and reporting that most of the benefit is exhausted by roughly 80% of a single epoch. Tan et al. [26] treat group size as a budget knob in an empirical scaling study of GRPO post-training. Snell et al. treat inference-time compute as a scaling axis in its own right, and the project's own related-work source cites these as the works against which its measurements should be read[31].

[29] Source: `platform_hybrid/paper/sections/p5_related.tex`.

[30] Sources: `platform_hybrid/paper/sections/p6_related_resources.tex`; `platform_hybrid/paper/sections/p6_intro.tex`.

[31] Source: `platform_hybrid/paper/sections/related_work_v2.tex`.

The project's scaling study covers 70+ runs across seven RL libraries and five model families. Its own documents do not agree on the upper end of the model-range: the P1 paper's introduction states the range as 0.6B–1T, while the P2, P3, and P4 introductions state 0.6B–~671B, and the related-work source states the cross-family data as spanning 0.6B to 235B across five families and four frameworks[32]. This divergence is a reporting defect in the portfolio's own documents and is recorded here rather than harmonised, because harmonising it would mean asserting a range that no single source states. The principal empirical result of the scaling work is a null: across that range, no reliable scaling law is recovered from reward curves[33]. The companion group-size study adds a second bounded negative: a three-seed sweep does not establish an optimum, since the measured arms are statistically indistinguishable, and the measured signal-to-noise ratio grows at only about 52% of the $\sqrt{G}$ ideal, with $G \in \{2, 4, 8, 16\}$ measured and $G \in \{4, 8, 16, 32, 64\}$ explicitly reconstructed rather than run.

Layer adaptation supplies the second half of this section's context, and connects the project's white-box investigation to the parameter-efficient fine-tuning literature. LoRA [27] freezes the pretrained weights and injects trainable low-rank matrices into the attention projections, achieving a 128× reduction in trainable parameters at $d = 4096$ with rank $r = 16$, which is on the order of 0.1% of the model's parameters. QLoRA [28] extends this by quantising the frozen base to NF4 with double quantisation at 0.37 bits per parameter — approximately 3 GB for a 65B model, permitting 65B fine-tuning on a single 48 GB GPU. The motivating arithmetic is that Adam at mixed precision costs roughly 16N bytes, exceeding 100 GB for a 7B model. Prefix tuning, prompt tuning, and adapter layers are the earlier members of the same family, and LoRA's advantages for RL specifically are that the frozen base doubles as the reference policy and that the trainable parameter count is small enough for modest hardware[34]. The project's layer-adaptation pillar asks a narrower and orthogonal question: where inside the network GRPO-style adaptation concentrates, and whether the layers that will matter can be identified early enough to freeze the rest. The reported finding is moderate concentration of per-layer adaptation, with the important negative that step-1 predictability does *not* survive scaling. This connects directly to Mukherjee et al. [29], who found that RL fine-tuning updates only 5–30% of model parame-

[32]Sources: `platform_hybrid/paper/sections/p1_intro.tex`, `platform_hybrid/paper/sections/p2_intro.tex`, `platform_hybrid/paper/sections/p3_intro.tex`, `platform_hybrid/paper/sections/p4_intro.tex`, `platform_hybrid/paper/sections/related_work_v2.tex`.

[33]Sources: `platform_hybrid/paper/sections/frontier_synthesis_scaling.tex`, `platform_hybrid/paper/sections/p1_conclusion.tex`.

[34]Source: `capstone-literature-survey/section2_7_peft.tex`.

ters across seven algorithms and ten model families, suggesting that RL activates a sparse subnetwork[35]. The length-bias pillar completes the scaling picture with its own bounded null: under a 200-token GSM8K generation cap, GRPO and Dr. GRPO produce held-out gains that are indistinguishable at the project's seed budget, and neither arm shows length inflation; the source is careful to state that the 200-token cap is part of the estimand rather than a nuisance detail, so the null bounds the capped, short-horizon regime and cannot adjudicate uncapped reasoning or long agent trajectories[36].

2.6 Process-, Tree- and Stability-Aware Variants

Beyond the group-baseline fixes, a parallel line of work changes *what* signal drives the update. Process reward models supply denser, step-level feedback instead of a single terminal reward, which reduces the chance that a whole group collapses to identical terminal rewards. Let's Verify Step by Step [30] showed that step-level PRMs substantially outperform outcome reward models on GSM8K and MATH; Math-Shepherd derives process rewards from automatically generated process annotations, avoiding costly human labels; PRM800K and PRMBench supply step-level evaluation data; OmegaPRM uses Monte-Carlo tree search to supervise process rewards without human-in-the-loop; ReasonEval introduces reasoning-specific reward models; and Implicit PRM learns process rewards implicitly from outcome labels, with VersaPRM extending process rewards to multi-domain tasks. Tree-of-thought search, self-consistency, and critic-style post-hoc verifiers all either benefit from or substitute for process supervision[37]. The same source documents the counter-current that matters most here: VerifyBench and "A Sober Look" warn that reward hacking and evaluation artefacts can inflate perceived gains on reasoning benchmarks, which is the project's stated motivation for insisting on binary verifiable rewards in its own mathematics work.

Stability-aware variants constrain the update itself. GSPO's sequence-level importance ratio, discussed above, is the clearest instance, credited with stabilising the Qwen3 mixture-of-experts series. SSPO's sentence-granularity objective is explicitly framed as balancing GSPO's stability against token-level GRPO's expressiveness. Entropy-guided advantage shaping in AERO / RL-ZVP regulates the update when the group carries no reward contrast, and GRESO's pre-screening is a sampling-side stability control. On the reasoning-RL side, GRPO-Lead, LCPO,

[35]Sources: `platform_tinker/reports/esa_phase1/Phase1_Project_Report_ZVF.tex`; `platform_hybrid/paper/sections/related_work_v2.tex`.

[36]Source: `platform_hybrid/paper/sections/p4_intro.tex`.

[37]Source: `platform_hybrid/paper/sections/related_work_v2.tex`.

and LIMR tune reasoning depth through length-aware rewards, Search-R1 trains reasoning agents to interleave search with chain-of-thought, and TTRL performs test-time RL against self-generated verifiers[38].

The Phase-1 survey's judgement on this body of work is retained without amendment, because the Semester 4 evidence has not changed it. These methods propose *levers*; this project first builds the *instrumentation* needed to tell whether any such lever converts wasted rollouts into held-out accuracy — ZVF, group utilisation, pairwise-contrast density, and the collapse decomposition. The nulls reported in the results chapters are exactly the kind of evidence that instrumentation is designed to produce, and they constitute a standing caution against adopting a variant on the strength of a single-seed improvement. The project applies that caution to its own proposals: the adaptive group-size controller is described as an implemented policy rather than an established improvement, and its decisive matched-token comparison remains unrun[39].

2.7 Positioning the Portfolio

The portfolio's positioning against prior art is a per-pillar statement of what is claimed and, more importantly, what is not. Table 2.1 reproduces the Phase-1 positioning and updates it with the Semester 4 evidence (Figure 2.2).

Table 2.1: Positioning of the eight portfolio papers (P1–P8) against closest prior art and the narrowed axis each claims.

Pillar	Closest prior art	Narrowed axis claimed
P1 (scaling)	whole-model scaling and saturation laws; Liu et al. on sparse RL subnetworks	Per-layer adaptation is moderately concentrated; step-1 predictability does not survive scaling; no reliable reward-curve scaling law recovered

[38]Source: `platform_hybrid/paper/sections/related_work_v2.tex`.

[39]Sources: `platform_tinker/reports/esa_phase1/Phase1_Project_Report_ZVF.tex`; `platform_hybrid/paper/sections/p7_intro.tex`.

Pillar	Closest prior art	Narrowed axis claimed
P2 (ZVF)	DAPO dynamic sampling, GVPO, AERO/RL-ZVP, GRESO, CPPO, NGRPO	*Measure* ZVF as an existence statistic; show it is entangled with accuracy and group size; propose PCD as the magnitude-aware replacement
P3 (group size)	integer-G sweeps; MC-GRPO; Tan et al. [26] budget-knob treatment	Trainability is non-monotone in G; a three-seed sweep finds its arms statistically indistinguishable and does not establish an optimum; measured SNR grows at ~52% of the $\sqrt{G}$ ideal
P4 (length bias)	Dr. GRPO and λ-GRPO length fixes	A bounded null: under a 200-token cap, GRPO and Dr. GRPO held-out gains are indistinguishable with no length inflation
P5 (reporting standard)	Datasheets, Model Cards, Data Statements, Data Cards	An eight-item RL-run reporting standard targeting the stack rather than the label
P6 (registry)	static run-config registries; HELM; LM Evaluation Harness; Papers with Code	Machine-readable stack records with per-component implementation status and an unauditability marker
P7 (controller)	heuristic adaptive-G (G^2RPO-A); DAPO dynamic sampling	Closed-form h_G(p) prediction of ZVF, with a controller described as implemented, not established

Pillar	Closest prior art	Narrowed axis claimed
P8 (fraud study)	tuned gradient-boosted trees as the tabular fraud-detection baseline	A measurement-discipline side study comparing a fine-tuned LLM with a gradient-boosted tree on synthetic credit-card fraud, with no RL claim (Chapter 8); P8 was archived and absorbed into the portfolio

The two crowded pillars are P2 and P4, where the literature is dense and the project accordingly narrows its claim to measurement rather than to a new algorithm. The widest open ground remains the systems and reporting side, where no GRPO-specific machine-readable reporting standard exists — the gap Section 2.4 establishes from three directions: HELM's unit is the (model, scenario) cell rather than the training stack, the LM Evaluation Harness standardises evaluation by executing code while the registry must describe stacks it cannot execute, and Papers with Code stopped at the repository boundary and was discontinued in July 2025[40].

The Semester 4 held-out campaign extends this positioning from method-comparison to benchmark practice. A single Tinker-trained actor — base `Qwen/Qwen3.6-35B-A3B` with the `pavlov-portfolio-qwen36-seed809-stepfinal` adapter, in bfloat16 — was evaluated against fourteen held-out agentic and reasoning suites under receipt, budget, and decontamination governance. The results are reported per suite and never pooled or averaged across suites. SWE-bench Pro recorded 2/731 = 0.274% pass@1 under its original contract, with all eighteen non-native outcomes retained in the denominator. VerilogEval recorded 129/312 = 41.35% pass@1 (67/156 code-completion and 62/156 spec-to-RTL), with a 129/311 sensitivity figure retained non-canonically. LAB-Bench's public split was completed at 1,967/1,967 across eight categories, with per-category accuracies ranging from 0.787 for TableQA to 0.000 for CloningScenarios. AgentDojo's benign-utility scope completed at 97/97. Omni-MATH closed with all 4,428 dispositions recorded and the official accuracy of 2,271/4,428 = 51.31% reproduced, with two parser exclusions verified to have no score effect. The remaining lanes

[40]Sources: `platform_hybrid/paper/sections/p6_related_resources.tex`; `platform_tinker/reports/esa_phase1/Phase1_Project_Report_ZVF.tex`.

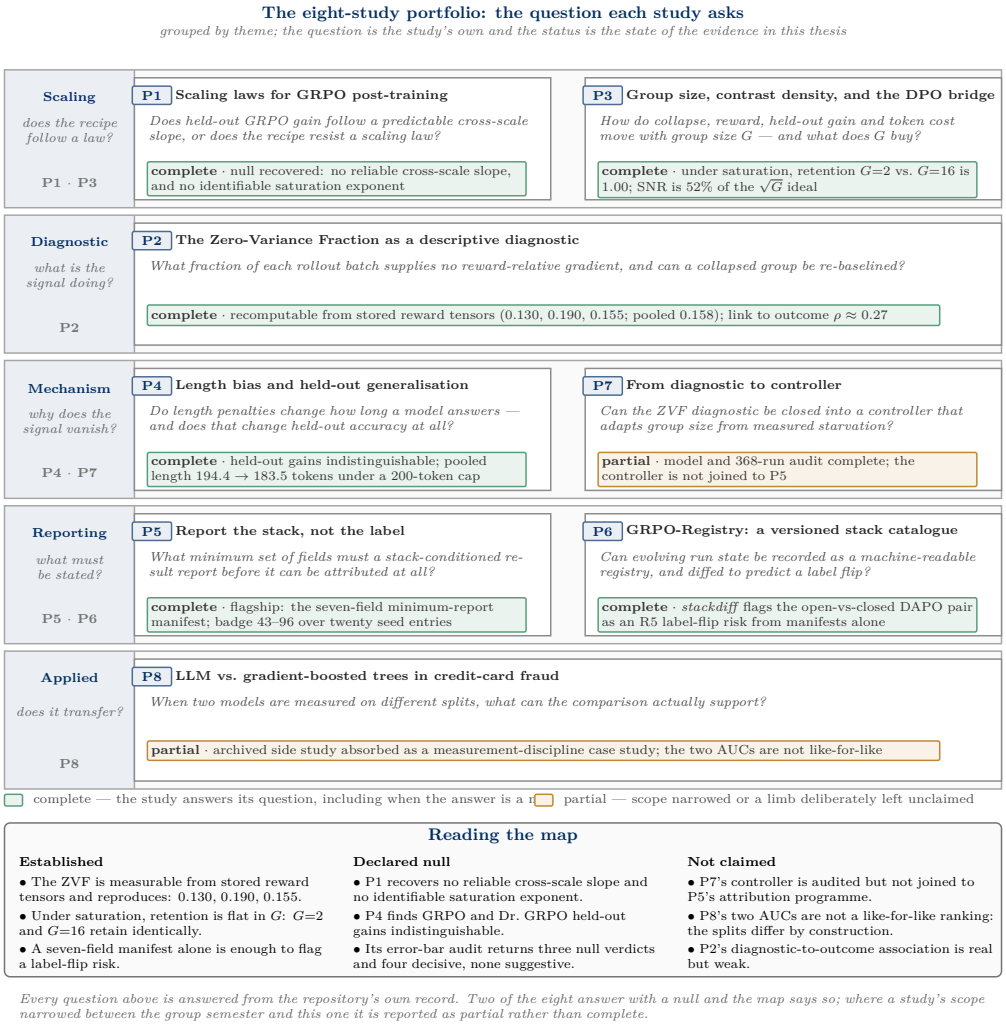


Figure 2.2: The eight studies P1–P8 grouped by theme, with the question each one addresses.

are recorded as partial, quota-blocked, or externally blocked: E2 FrontierSWE at 1/17 tasks with a normalised replay of 0.8628, E4 BankerToolBench at 1/100 tasks with a recovery metric of 0.3115, E5 APEX-Agents at 7/480 natively scored with a prefix mean of 0.050505, E9 MLE-bench at 40/75 competitions natively graded with a null suite score, E13 BALROG at 13/255 episodes, and six lanes (E3, E7, E8-original, E10-original, E12, E14-original) closed as externally blocked with no provider response and named reopen conditions. Structural claims were re-verified by deterministic code checks on 19 September 2026 at 11/11 PASS[41]. No claim of improvement over a baseline or over the base model is made anywhere in the campaign, and the agentic-suite results in particular — E4's mean reward of 0.0 across 100 base-model trials, with 27 or more trajectories ending in role-token degeneration — are reported as evidence about tool-dialogue collapse rather than as evidence about the tasks the suites nominally measure[42].

Two gaps in this survey should be stated plainly, since the chapter's purpose is to locate the work rather than to flatter it. First, the primary papers for the fourteen evaluation suites are not surveyed here at the depth the algorithmic literature receives; the suites are discussed as instruments, sourced to the campaign's own results document, and their native protocols are documented in Chapter 9 and its receipt trail rather than in this chapter. Second, the survey's treatment of tool-use and agentic RL is inherited from the project's own related-work source and is explicitly flagged there as a concurrent setting in which the ZVF measurements will need to be re-evaluated rather than assumed to transfer[43]. What the chapter does establish is that the project's contribution is a measurement and reporting contribution, that the literature it sits in is dense enough that claim-narrowing rather than claim-extension is the honest move, and that the specific gap the portfolio occupies — machine-readable, per-component, executed-loss-form disclosure for group-relative RL runs — is not filled by any of the resources surveyed above.

[41]Source: `outputs/E1_E14_FINAL_RESULTS_2026-09-19.md`.
[42]Source: `outputs/E1_E14_FINAL_RESULTS_2026-09-19.md`.
[43]Source: `platform_hybrid/paper/sections/related_work_v2.tex`.

Chapter 3

System Requirements Specification

This chapter states what the Tinker RL Lab system must do, what qualities it must hold while doing it, and how each requirement is verified. It inherits the specification of the Phase-1 report and extends it across the Semester-4 additions: the multi-framework benchmark rosters, the Zero-Variance Fraction (ZVF) diagnostic and the adaptive group-size controller derived from it, the minimum reporting standard and the machine-readable GRPO stack registry, and the E1–E14 held-out evaluation campaign with its fail-closed completion gate. It is written as a specification rather than as a retrospective description, but every requirement below corresponds to a component that exists in the repository, and every functional requirement is paired with the concrete test that checks it. Where a requirement is not satisfied — the cloud capacity for two evaluation lanes, the white-box gradient path at the largest model scale — the shortfall is named in §3.6 and §3.8 rather than written around.

3.1 Overview and Actors

The central object of the Phase-1 specification was the **run**: one configuration, one seed, one framework, one training budget, producing per-step telemetry and a persisted set of raw group reward tensors[1]. Semester 4 introduces two further first-class objects. The first is the **lane**: a single benchmark suite evaluated under a named scope — an original contract with the suite's owner, or a declared replacement scope — carrying its own provider, payload, licence, quota and terminal state. The second is the **receipt**: an immutable, hash-bound record that a

[1]Source: platform_tinker/reports/esa_phase1/Phase1_Project_Report_ZVF.tex.

declared step actually happened, against declared inputs, at a declared revision. Requirements that concern runs, lanes and receipts are distinguished below where the distinction matters.

Four actors interact with the system, and the fourth is new in Semester 4.

The experimenter declares work. Under the original design this meant declaring a run through a single configuration object and nothing else: the model, the dataset, the reward function, the decoding parameters (temperature, top-p, maximum tokens), the group size G, the learning rate, the KL coefficient β and the seed all live in one file, and the framework is a switch rather than a rewrite[2]. Under the campaign design the same actor declares a lane through a sealed resource request: the lane specification fixes an immutable task bundle, a pinned native runtime, a native grader, a budget and an applicable licence in advance, and those declared inputs are then checked for internal consistency before any launch is permitted[3].

The back-end executes the training loop and, under the managed path, is the Tinker service; under the white-box path it is a local or rented GPU. For evaluation, the corresponding actor is the **provider grader**: the benchmark owner's own native evaluator at a pinned revision. The system is permitted to record a suite score only when that evaluator produced it, or when a declared replacement harness reproduces it under a scope that is labelled as a replacement[4].

The auditor is a downstream script or an independent reader who recomputes published diagnostics from stored raw artefacts. In Semester 4 this role acquires force: the audit suite of §3.4 is executed as a gate with a non-zero exit status rather than as advisory tooling, and the completion gate for the campaign lanes is fail-closed by construction, in that a missing or malformed declaration stops the procedure instead of degrading it.

The campaign operator holds launch authority for paid work. Spend on any lane requires a bound authorization receipt issued by the operator; the system will not mint one for itself, and lanes are recorded as blocked when the receipt or the technical chain behind it is absent[5].

A fifth party, the **external access holder**, is an actor only in the negative sense: several lanes depend on a private payload, a hosted evaluation service or a quota grant that only the provider can issue. The specification requires that such a lane be recorded with an explicit terminal state and a named reopen condition rather than silently dropped, which is why the campaign's terminal-state vocabulary

[2]Source: platform_tinker/reports/esa_phase1/Phase1_Project_Report_ZVF.tex.
[3]Source: zvf-program/flagship/e1_e14_completion_gate.py.
[4]Source: zvf-program/flagship/e1_e14_completion_gate.py.
[5]Source: outputs/PES_Phase2_Review_2026-09-12/finish/Pending_Experiments.md.

exists as a first-class output[6].

3.2 Functional Requirements

FR-1 — Single configuration object. One declarative file shall fully determine a run: model, dataset, reward function, decoding parameters, group size, learning rate, KL coefficient, seed and output location. No framework-specific parameters may be required from the experimenter to obtain a valid run on any supported back-end[7].

FR-2 — Back-end-agnostic execution through thin adapters. A configuration shall be executable on multiple frameworks through adapters that map the shared configuration onto each framework's own interface, while sharing the reward grader and the decoding configuration rather than re-implementing them. Two framework rosters are in scope and must not be conflated. The cross-RL-library roster contains TRL, Stable-Baselines3, CleanRL, Tianshou, PufferLib, rl_games and d3rlpy; the cross-launcher LLM-RL roster contains Tinker, TRL, SkyRL, veRL, OpenRLHF, Atropos and the Hugging Face reference launchers[8]. Completed-run parity across the second roster is a requirement the system does **not** yet meet: of its seven entries, only Tinker and TRL produced completed runs in this release, and the veRL and OpenRLHF entries in the comparison artefact are dry-run placeholders[9].

FR-3 — Per-step signal-starvation measurement. The training loop shall compute, at every step, the fraction of prompt-groups whose within-group reward variance is zero, together with the decomposition of that fraction into all-correct and all-wrong groups. These are the ZVF and the collapse decomposition, and they are derived quantities of the reward tensors rather than proxies[10].

FR-4 — Per-step telemetry. Each step shall emit a structured record carrying reward, ZVF, gradient utilisation, the all-correct and all-wrong collapse counts, policy entropy, completion length and KL divergence, written both to a machine-readable log and to the experiment tracker[11]. Semester-4 controller work extends this record with the per-step counterfactual comparison between the adaptive

[6]Source: outputs/PES_Phase2_Review_2026-09-12/finish/Pending_Experiments.md.
[7]Source: platform_tinker/reports/esa_phase1/Phase1_Project_Report_ZVF.tex.
[8]Source: platform_hybrid/paper/sections/_shared_methods.tex.
[9]Source: platform_hybrid/paper/sections/_shared_methods.tex; source: platform_hybrid/experiments/results/framework_comparison.json.
[10]Source: platform_tinker/reports/esa_phase1/Phase1_Project_Report_ZVF.tex.
[11]Source: platform_tinker/reports/esa_phase1/Phase1_Project_Report_ZVF.tex.

schedule and its matched fixed-group-size control[12].

FR-5 — Raw tensor persistence. The system shall persist the raw per-group reward tensors for every run, so that any derived diagnostic can be recomputed deterministically after the fact rather than trusted from a logged value[13].

FR-6 — Sweeps under a fixed budget. The runner shall support sweeps over group size, seed and baseline-versus-intervention arms, executed under a fixed wall-clock or token budget so that arms are comparable in compute rather than in step count[14].

FR-7 — White-box gradient path. For models small enough to train on a single GPU, the system shall record per-layer LoRA gradient norms alongside the usual telemetry, so that the relationship between reward spread and gradient magnitude can be examined directly rather than inferred. This path is scoped to the small-model configurations and is not a requirement at the large managed scale, where per-layer gradients are not exposed[15].

FR-8 — Per-run provenance record. Every run shall emit a machine-readable provenance record binding the configuration, the grader or verifier version, and the rollout hashes, so that a published number can be tied to the exact inputs that produced it[16]. In the campaign, the equivalent obligation is stronger: a lane declares its provider package and grant documents against published schema versions, each carrying a 64-character hash, and the declared inputs must be internally consistent and available locally before the lane is admissible[17].

FR-9 — Minimum-reporting manifest and audit. A run submitted as a stack-conditioned result shall carry the seven-field minimum-report manifest — loss form, reference KL, sampler backend including base-checkpoint revision and hash, telemetry, group-size schedule, held-out split, and decontamination — plus the eighth evaluation item, a held-out pass@k report; the resource shall emit a completeness badge on a 0–100 scale and shall distinguish an unreported field from a field reported as absent[18].

FR-10 — Registry query surface. The stack registry shall be queryable: a schema plus a set of entry documents and a query interface that returns entries by stack, field or status, so that a reader can ask what a given labelled method

[12]Source: platform_hybrid/experiments/results/p5p8/controller_cf_per_step.tsv.
[13]Source: platform_tinker/reports/esa_phase1/Phase1_Project_Report_ZVF.tex.
[14]Source: platform_tinker/reports/esa_phase1/Phase1_Project_Report_ZVF.tex.
[15]Source: platform_tinker/reports/esa_phase1/Phase1_Project_Report_ZVF.tex.
[16]Source: platform_tinker/reports/esa_phase1/Phase1_Project_Report_ZVF.tex.
[17]Source: zvf-program/flagship/e1_e14_completion_gate.py.
[18]Source: platform_hybrid/paper/paper_P6_registry.tex; source: platform_hybrid/experiments/results/p5p8/minreport_audit_summary.json.

actually did rather than what its name implies[19].

FR-11 — Variant-delta records and stack-diff. For each named method variant the registry shall store an explicit delta against the GRPO reference — for example DAPO's asymmetric clipping, dynamic sampling, token-level loss, overlong-reward shaping and removed KL term; Dr. GRPO's removal of the length and standard-deviation normalisations; GSPO's sequence-level ratio — and the stack-diff procedure shall return a flip-risk verdict on the 0–5 scale when two entries differ in label but not in substance[20].

FR-12 — Adaptive group-size controller. The system shall implement a controller that adjusts group size in response to the measured signal, with an escalation asymmetry, hysteresis to prevent oscillation, and a callback that diverts degenerate groups away from the fixed schedule, and it shall record the counterfactual comparison against the best static recipe at matched and at unequal rollout counts[21]. The controller is specified as a training-time policy, not as a promotion mechanism: the requirement is to measure the trade, not to claim a win.

FR-13 — Lane harness with evidence classes. Each evaluation lane shall be driven by a harness that records per-lane results against a declared evidence class — exact complete, partial exact, partial recovery, externally blocked, or local-setup-ready-provider-input-required — and shall keep the class assignment tied to the source status rather than to the operator's optimism[22].

FR-14 — Fail-closed completion gate. A declared lane package shall be validated, not executed, by the completion gate. A valid package proves only that the declared immutable inputs are internally consistent and available locally; accepting a package as a benchmark result is explicitly out of scope for the gate. Signature verification against provider trust roots shall be performed, and the gate shall refuse rather than warn[23].

FR-15 — Scope-separation contract in reporting. Original-contract and replacement-scope results shall never be pooled, no cross-suite aggregate shall be computed, and every reported accuracy shall carry its coverage denominator. This is a requirement on the outputs, not only on the analysis code[24].

FR-16 — Blind-review packaging with an unskippable audit guard. The

[19]Source: platform_hybrid/registry/schema.json; source: platform_hybrid/registry/query.py.

[20]Source: platform_hybrid/paper/paper_P6_registry.tex; source: platform_hybrid/registry/provenance/.

[21]Source: platform_hybrid/paper/paper_P7_zvf_controller.tex; source: platform_hybrid/experiments/results/p5p8/controller_cf_summary.json.

[22]Source: zvf-program/flagship/e1_e14_completion_gate.py.

[23]Source: zvf-program/flagship/e1_e14_completion_gate.py.

[24]Source: outputs/E1_E14_FINAL_RESULTS_2026-09-19.md.

export path shall produce an anonymised review package whose build runs the audit suite, and the guard shall detect and refuse a package built with the audits skipped[25].

3.3 Non-Functional Requirements

NFR-1 — Attribution. Every comparative claim shall be made with the whole stack held fixed except the factor under test, and framework, algorithm, model family and scale shall be reported as confounded where they are confounded[26].

NFR-2 — Reproducibility. Multi-arm claims shall rest on at least three seeds; single-seed results shall be labelled descriptive or exploratory and shall not be used to support a directional claim[27].

NFR-3 — Recomputability. Every published derived diagnostic shall be reproducible from stored raw artefacts by a documented procedure, with no dependence on a logged summary value[28].

NFR-4 — Honesty and denominator discipline. Held-out figures shall be reported with their denominator n; negative and null results shall be reported as such; where an analysis was underpowered, that shall be stated rather than left implicit[29].

NFR-5 — Isolation. Concurrent experiments shall run in separate processes rather than threads, following a tracker thread-safety defect observed in Phase 1[30].

NFR-6 — Budget. Managed-compute spend shall be recorded per run and counted against a declared envelope; even where the standing authorization removes the cumulative cap and sets the total to unlimited, per-run resource and timeout bounds remain in force and paid launches remain bound to a lead-issued authorization receipt[31].

NFR-7 — Fail-closed behaviour. A validation boundary shall default to refusal. The gate neither invokes an adapter nor accepts a package as a result, and a package that cannot be verified is not scored[32].

[25]Source: platform_local/run_all_audits.py; source: platform_local/export_guard_audit.py.
[26]Source: platform_hybrid/paper/sections/_shared_methods.tex.
[27]Source: platform_tinker/reports/esa_phase1/Phase1_Project_Report_ZVF.tex.
[28]Source: platform_tinker/reports/esa_phase1/Phase1_Project_Report_ZVF.tex.
[29]Source: platform_hybrid/paper/main_eai_body.tex.
[30]Source: platform_tinker/reports/esa_phase1/Phase1_Project_Report_ZVF.tex.
[31]Source: outputs/PES_Phase2_Review_2026-09-12/finish/Pending_Experiments.md.
[32]Source: zvf-program/flagship/e1_e14_completion_gate.py.

NFR-8 — Least privilege and no credential retention. The system shall hold no key material in its evidence records: credential checks record only liveness and scope, and the resulting receipts state explicitly that no key material is stored[33].

NFR-9 — Auditability under lost-source conditions. Execution sources for anything declared sealed shall live in tracked paths, so that a lost working directory cannot orphan a sealed artefact; where loss nonetheless occurs, the sealed requests and receipts shall remain sufficient to pin a faithful re-implementation[34].

NFR-10 — Traceability of external artefacts. Externally supplied payloads and hosted results shall be bound by hash equality and validated against published trust roots and schema versions, and an unreported field shall be represented as a null rather than as a zero or a false value[35].

3.4 Test-Case Specification

The test cases below are the concrete checks attached to the requirements. TC-1 through TC-5 are inherited from the Phase-1 specification and are restated here because later chapters depend on them; TC-6 onward are the Semester-4 additions, and TC-6 is the suite that carries the largest share of the functional requirements.

TC-1 — Recompute ZVF and the collapse decomposition from stored tensors. Recompute the published ZVF and collapse figures from the persisted per-group reward tensors of the main configuration, using the documented analysis script. The published values are a ZVF of approximately 0.72–0.77 and an all-correct fraction of 0.65–0.71; recomputation must match[36]. Verifies FR-3 and FR-5.

TC-2 — Recompute the detector metric from raw tensors. Recompute the Phase-3 detector's area under the ROC curve from raw tensors under five-fold stratified cross-validation. The published figure is an AUROC of 0.84 ± 0.01 against 0.43 for the reward-only comparator[37]. Verifies FR-5 and NFR-3.

TC-3 — Re-run a matched-budget arm comparison. Re-run the baseline and the curriculum arm at a matched token budget over three seeds; the Phase-1 result is that both arms average +0.028 on the held-out set with no curriculum

[33]Source: outputs/PES_Phase2_Review_2026-09-12/finish/Pending_Experiments.md.
[34]Source: outputs/E1_E14_FINAL_RESULTS_2026-09-19.md.
[35]Source: zvf-program/flagship/e1_e14_completion_gate.py; source: platform_hybrid/paper/paper_P6_registry.tex.
[36]Source: platform_tinker/reports/esa_phase1/Phase1_Project_Report_ZVF.tex.
[37]Source: platform_tinker/reports/esa_phase1/Phase1_Project_Report_ZVF.tex.

advantage[38]. Verifies FR-1, FR-2 and FR-6.

TC-4 — Recompute headline diagnostics. Recompute each headline diagnostic quoted in a results chapter directly from stored artefacts, with the recomputation recorded next to the published value[39]. Verifies NFR-3.

TC-5 — Provenance verification. Verify a run's provenance record end to end, from configuration to rollout hashes to the grader version[40]. Verifies FR-8.

TC-6 — Execute the submission audit suite as a gate. Run `python platform_local/run_all_audits.py`. The suite executes nine audits — `claim_issues`, `sync_issues`, `anon_issues`, `strength_issues`, `package_issues`, `workflow_issues`, `export_guard_issues`, `caveat_issues` and `scientific_issues` — emits one `METRIC <name>=<count>` line per audit, and returns exit status 0 only when every audit reports zero issues[41]. Individual checks that this test enforces include the presence of scope and caveat language in the review documents (`readme_missing_key_results_scope_header`, `readme_missing_heldout_language`, `checklist_missing_preliminary_key_results_label`), the absence of overstated checkpoint claims (`misleading_checkpoint_availability_claim`, `submission_overstates_checkpoint_release`), the presence of every reviewer caveat (`heldout_scope`, `tool_eval_protocol`, `codegen_subset`, `capacity_confound`, `moe_diagnostics`, `budget_and_splits`, `replication_release`), the absence of banned marketing formulations[42], the survival of the technical caveats into the anonymous build (`anonymous_missing: <label>` for the seven needles), and the presence of the audit call and its skip flag in the export script[43]. Verifies FR-16, NFR-4 and NFR-7.

TC-7 — Structural checks on the evaluation script and its result files. For each held-out evaluation script, assert by parsing the source — not by reading its documentation — that it exposes a seed argument, locks the dataset split to the test split, samples with a non-zero temperature, consumes a checkpoint path, and writes the declared metadata keys including the split, the sampling flag, the seed and the model source. For each result file, assert that the schema version is 2, that the evaluation status is one of completed or failed, that the required configuration and summary fields are present, that the recorded split is the test split, that the attempted count equals the sum of correct and incorrect, and that a failed evaluation carries a failure reason and a completed evaluation carries non-

[38]Source: platform_tinker/reports/esa_phase1/Phase1_Project_Report_ZVF.tex.
[39]Source: platform_tinker/reports/esa_phase1/Phase1_Project_Report_ZVF.tex.
[40]Source: platform_tinker/reports/esa_phase1/Phase1_Project_Report_ZVF.tex.
[41]Source: platform_local/run_all_audits.py; source: utils/audit_utils.py.
[42]Source: platform_local/claim_strength_audit.py.
[43]Source: platform_local/paper_sync_audit.py; source: platform_local/export_guard_audit.py.

zero attempts[44]. Verifies FR-8 and NFR-4.

TC-8 — Build the report from source. Compile the main report and its supplementary through the full LaTeX chain and flag an empty journal field in the bibliography[45]. Verifies FR-16.

TC-9 — Dry-run the completion gate against a declared lane. Present a provider package to the gate and require it to validate the declaration without executing anything: the expected outcomes are acceptance with the declared evidence class, or a specific refusal naming the missing element. A package must never be reported as a benchmark result by this procedure[46]. Verifies FR-13, FR-14 and NFR-7.

TC-10 — Registry query and null discipline. Query the registry for a labelled entry and its delta record, and confirm that a field absent from the source is returned as unreported rather than as a zero value, and that a same-label pair with divergent behaviour triggers the flip-risk verdict[47]. Verifies FR-9, FR-10, FR-11 and NFR-10.

3.5 Requirements Traceability Matrix

Table 3.1: Requirements traceability matrix: each functional requirement mapped to the artefacts that implement it and the audit check that verifies it.

Requirement	Implemented in	Verified by
FR-1 Configuration object	`platform_local/unified/launcher.py`; run configurations under `platform_hybrid/experiments/`	TC-3
FR-2 Back-end-agnostic execution	`platform_local/trl_integrations/`; `platform_hybrid/experiments/results/framework_comparison.json`	TC-3; parity shortfall recorded in §3.2

[44]Source: platform_local/scientific_audit.py.
[45]Source: platform_local/scientific_audit.py.
[46]Source: zvf-program/flagship/e1_e14_completion_gate.py.
[47]Source: platform_hybrid/paper/paper_P6_registry.tex; source: platform_hybrid/registry/query.py.

Requirement	Implemented in	Verified by
FR-3 Per-step ZVF	Training-loop telemetry (Phase-1 design)	TC-1
FR-4 Per-step telemetry	Telemetry writer; `platform_hybrid/experiments/results/p5p8/controller_cf_per_step.tsv`	TC-1, TC-4
FR-5 Raw tensor persistence	Persisted per-group reward tensors	TC-1, TC-2
FR-6 Sweeps	Sweep runner	TC-3
FR-7 White-box gradients	Single-GPU training path	Not automatable in the audit suite; checks in Chapter 5 §5.5 (per-layer gradient profile)
FR-8 Provenance record	`zvf-program/flagship/e1_e14_completion_gate.py`; run provenance records	TC-5, TC-7, TC-9
FR-9 Minimum-report manifest	`platform_hybrid/experiments/results/p5p8/minreport_audit_summary.json`	TC-10
FR-10 Registry query	`platform_hybrid/registry/schema.json`, `query.py`, `entries/`	TC-10
FR-11 Variant deltas, stack-diff	`platform_hybrid/registry/provenance/`	TC-10
FR-12 Adaptive controller	`platform_hybrid/experiments/results/p5p8/controller_cf_summary.json`	TC-4

Requirement	Implemented in	Verified by
FR-13 Lane harness	`zvf-program/e1_wave10/`, `zvf-program/e5_successor27/`, `zvf-program/e2_core/`, `zvf-program/e13_balrog/`, `zvf-program/e6e9/check_quota_status.py`	TC-9
FR-14 Completion gate	`zvf-program/flagship/e1_e14_completion_gate.py`	TC-9
FR-15 Scope separation	`outputs/E1_E14_FINAL_RESULTS_2026-09-19.md`	TC-6, TC-9
FR-16 Blind-review packaging	`platform_local/run_all_audits.py`, `export_guard_audit.py`	TC-6, TC-8
NFR-1 Attribution	Method sections and comparison artefacts	TC-3, TC-10
NFR-2 Reproducibility	Seed management in the runner	TC-3
NFR-3 Recomputability	Documented analysis procedures	TC-1, TC-2, TC-4
NFR-4 Honesty	Caveat and claim-strength checks	TC-6, TC-7
NFR-5 Isolation	Process-level execution	Operationally checked; not in the audit suite
NFR-6 Budget	Spend receipts	TC-9
NFR-7 Fail-closed behaviour	Gate constants and suite exit status	TC-6, TC-9
NFR-8 Least privilege	Credential-gate receipts	TC-9
NFR-9 Auditability	Tracked execution paths	TC-9
NFR-10 External traceability	Hash equality and trust-root validation	TC-9, TC-10

Two rows deserve comment. FR-7 has no automated check because the white-box gradient path is a measurement procedure rather than an invariant; its evidence is the result data itself. NFR-5 is likewise checked operationally rather than by

the suite, because process isolation is a property of how runs are launched and is not visible in any post-hoc artefact.

3.6 Hardware Requirements

The system requires four compute surfaces. The **managed training back-end** executes GRPO at the scale of the campaign actor, a mixture-of-experts model of roughly 35 billion total parameters with about 3 billion active per token, evaluated in bfloat16 with a LoRA adapter at a pinned revision[48]. The requirement here is not a particular accelerator but the service contract: the back-end exposes sampling, training and checkpointing, and does **not** expose per-layer gradients, which is why FR-7 is scoped to the small-model path[49].

The **white-box GPU** requirement is a single CUDA device with at least 24 GB of memory, met in Phase 1 by a cloud notebook instance with an L4-class GPU, sufficient for the 1.5-billion-parameter configuration and for a scaled re-test at 3 billion parameters on reduced sequence lengths[50]. The **batch scoring and held-out evaluation** surface is a serverless GPU path used for large scoring sweeps and for the hosted sampler bridge that the tool-use lane depends on; that bridge is a required component, and its availability is a precondition for that lane rather than an assumption[51].

The **evaluation-lane compute** is the Semester-4 addition with the sharpest unmet requirement. Two lanes are gated on cloud capacity that is currently below the required level: one has requested eight standard on-demand vCPUs in its region (a documented minimum of four, for a single four-vCPU builder instance) against a live quota of one, and the other needs sixteen against a live quota of one, with quota increase requests filed and still open[52]. A third lane requires a local container build for which the recipe reconstruction needs at least 30 GiB of free disk against 31 GiB available, which is marginal rather than comfortable[53]. A fourth lane requires Cloud project resources whose binding must be minted by the project owner; that binding is a hard prerequisite the system cannot satisfy on its own[54]. The **orchestration workspace** is an ordinary developer workstation

[48]Source: outputs/E1_E14_FINAL_RESULTS_2026-09-19.md.
[49]Source: platform_tinker/reports/esa_phase1/Phase1_Project_Report_ZVF.tex.
[50]Source: platform_tinker/reports/esa_phase1/Phase1_Project_Report_ZVF.tex.
[51]Source: outputs/UNBLOCK_CARRYOUT_2026-09-21.md.
[52]Source: outputs/E1_E14_FINAL_RESULTS_2026-09-19.md; source: outputs/PES_Phase2_Review_2026-09-12/finish/Pending_Experiments.md; source: outputs/PES_Phase2_Review_2026-09-12/finish/e9_completion/status_2026-09-19.json.
[53]Source: outputs/UNBLOCK_CARRYOUT_2026-09-21.md.
[54]Source: outputs/UNBLOCK_CARRYOUT_2026-09-21.md.

used for orchestration, analysis and figure generation, and carries the audit suite and the report toolchain.

3.7 Software Requirements

The runtime is Python 3.11 with PyTorch and the Hugging Face stack (Transformers, PEFT, Datasets) for model handling, tokenisation and adapters; the RL back-ends are TRL, veRL through its HybridFlow path, and OpenRLHF, driven through the shared launcher, with the Tinker client SDK for the managed path[55]. Experiment tracking is a weights-and-biases project; telemetry is stored as JSONL and as persisted tensors; analysis is carried out by the documented scripts named in the Phase-1 specification[56].

Semester 4 adds five software components to the requirement set. First, the audit suite: `platform_local/run_all_audits.py` together with the nine audit modules it imports and the shared data model in `utils/audit_utils.py`, which defines the issue, result and suite types and the `METRIC` line format; the suite requires a Python interpreter with no third-party dependencies beyond the standard library for its structural checks[57]. Second, the completion gate and the per-lane harnesses under `zvf-program/`, which require Ed25519 signature verification and JSON Schema validation against the published provider-package, provider-grant and trust-root schema versions[58]. Third, the registry: a schema, an entry store and a query interface, together with the provenance records under `platform_hybrid/registry/`[59]. Fourth, the report toolchain: a LaTeX distribution capable of the multi-pass build with BibTeX, plus a markdown assembler that concatenates the chapter files and, when available, renders the document through pandoc[60]. Fifth, the judgement-layer harness under `zvf-program/jev_lab/`, which records its own receipts and is used only for classification, faithfulness checking and value ordering[61].

[55]Source: platform_tinker/reports/esa_phase1/Phase1_Project_Report_ZVF.tex.
[56]Source: platform_tinker/reports/esa_phase1/Phase1_Project_Report_ZVF.tex.
[57]Source: platform_local/run_all_audits.py; source: utils/audit_utils.py.
[58]Source: zvf-program/flagship/e1_e14_completion_gate.py.
[59]Source: platform_hybrid/registry/query.py.
[60]Source: platform_local/scientific_audit.py; source: outputs/PES_Phase2_Third_Review_2026-09-24/thesis/assemble_thesis.py.
[61]Source: outputs/E1_E14_FINAL_RESULTS_2026-09-19.md.

3.8 Constraints and Assumptions

Rewards are binary and verifiable. The specification assumes a programmatic checker — exact match for the mathematics tasks and unit tests for the code subset. Nothing in the requirement set addresses learned reward models, and none of the diagnostics should be read as applying to them[62].

Held-out sets are small. Several evaluation sets have n between 8 and 20, with the consequence that learning-gain magnitudes are noise-limited by construction. This is a constraint on what the results can mean, not a defect of the analysis: the small-n figures are reported with their denominators and are not used to support comparative claims[63].

The managed back-end is a black box at the layer level. Per-layer gradients are not exposed on the managed path, so claims about the relationship between reward spread and gradient magnitude are confined to the small-model white-box configuration and are directional only[64].

Group-size differences must not be attributed to sampling noise. The comparison protocols share a deterministic prompt schedule, so that method-to-method differences in ZVF reflect training dynamics rather than independent draws of prompts[65].

Framework parity is incomplete. The requirement in FR-2 covers two seven-library rosters, but only two of the seven cross-launcher entries produced completed runs in this release, and the remaining comparison entries are dry-run placeholders. Comparative framework claims are therefore constrained to the rosters and the completed runs actually recorded, and the repository states the dry-run status explicitly[66].

The scale study reports descriptive limits rather than a scaling law. The cross-scale analysis spans more than seventy runs across seven libraries and five model families, and its central result is a negative one: no reliable positive cross-scale slope is recovered and no saturation exponent is identifiable. The specification inherits that boundary — the requirement is to report identifiability limits, not to assert a law[67].

External dependencies are assumed unavailable until proven otherwise.

[62]Source: platform_tinker/reports/esa_phase1/Phase1_Project_Report_ZVF.tex.
[63]Source: platform_tinker/reports/esa_phase1/Phase1_Project_Report_ZVF.tex.
[64]Source: platform_hybrid/paper/paper_P7_zvf_controller.tex.
[65]Source: platform_tinker/reports/esa_phase1/Phase1_Project_Report_ZVF.tex.
[66]Source: platform_hybrid/paper/sections/_shared_methods.tex; source: platform_hybrid/experiments/results/framework_comparison.json.
[67]Source: platform_hybrid/paper/paper_P1_scaling.tex.

Payload grants, hosted evaluations and quota increases are treated as absent until a response arrives; a lane with no provider response is closed as externally blocked with a reopen condition rather than left pending indefinitely[68].

Arithmetic and string lookups stay in code. Whatever judgement layer is used for triage, classification or value ordering, verification-type checks are performed by deterministic code. This rule was banked after the judgement layer returned an incorrect answer to a parity probe in a healthy window, and it constrains the architecture: no requirement in this chapter may be verified by a language model where a deterministic check is possible[69].

Named variants are labels, not specifications. The assumption that a method name identifies a method is false on the available evidence: entries sharing a label diverge substantially in measured behaviour, which is why the registry's delta records and flip-risk verdicts are specified as requirements rather than as documentation niceties[70].

[68]Source: outputs/E1_E14_FINAL_RESULTS_2026-09-19.md.
[69]Source: outputs/E1_E14_FINAL_RESULTS_2026-09-19.md.
[70]Source: platform_hybrid/paper/paper_P6_registry.tex.

Part II

Design and Implementation

Chapter 4

Proposed Methodology

This chapter sets out the methodology of the work rather than its results. It states four things in order: the design commitment to attribution and the four de-confound pillars that operationalise it (§4.1); the per-step telemetry that makes a run's internal state observable at the granularity at which the mechanism acts (§4.2); the formalisation of the Zero-Variance Fraction as a descriptive diagnostic rather than a metric (§4.3); the eight-study portfolio P1–P8 and the question each study asks (§4.4); and the design of the E1–E14 held-out evaluation campaign, including its single trained actor, pinned native evaluators, frozen sampling, receipt model, terminal-state taxonomy and decontamination handling (§4.5). Throughout, the reporting discipline is part of the method: figures are given only where a repository file supports them, and where a lane produced no admissible result, it is recorded as blocked rather than written around (Figure 4.1).

4.1 The Design Commitment to Attribution

The premise of the project is that an RL-for-LLM result is not identified by the name of its algorithm. Two runs both labelled GRPO may differ in loss form, reference-policy handling, sampler, precision, base-checkpoint identity, data order and group-size schedule, and the label alone does not say which of those differences produced the outcome. The project's response is a design commitment rather than a post-hoc covariate adjustment: hold the whole experimental stack fixed and vary exactly one factor at a time, so that an observed difference is attributable to the factor that was varied rather than to the several things that changed alongside it[1]. This commitment is the methodological spine of the whole thesis, and it has a practical corollary that shapes every experiment reported later: a study is defined

[1]Source: platform_tinker/reports/esa_phase1/Phase1_Project_Report_ZVF.tex.

Pillar 1 · Factor isolated: the advantage estimator

VARY — the advantage estimator only. PPO and GRPO are run on one and the same stack, so the estimator is the single difference between the two arms.

FIX — model checkpoint, dataset slice, decoding path, reward grader, framework, schedule and seed set.

Measured outcome. *Paired* $\Delta = -0.002$, $p = 0.374$*: the estimator effect is small and not statistically significant.*

Pillar 2 · Factor isolated: signal starvation

VARY — the internal signal state of the batch: per-step Zero-Variance Fraction and its collapse decomposition, logged at every step rather than summarised after the run.

FIX — the training stack is untouched; the diagnostic observes, it does not intervene.

Measured outcome. *Portable across nine library back-ends;* 0.16 *mean ZVF on Qwen3-8B / GSM8K at* $G = 8$.

Pillar 3 · Factor isolated: group size

VARY — the group size $G \in \{2, 4, 8, 16\}$ across seeds, holding model, data, reward and schedule fixed.

FIX — model, data, reward grader, learning-rate schedule and evaluation split.

Measured outcome. *Mean ZVF falls* $0.838 \rightarrow 0.631$ *(*$G = 2 \rightarrow 16$*); held-out retention is* ≈ 1.00.

Pillar 4 · Factor isolated: held-out behaviour

VARY — the method, Dr. GRPO versus GRPO, compared pre-RL against post-RL on data the training loop never saw.

FIX — the pre-RL checkpoint and the held-out split, evaluated side by side under frozen sampling.

Measured outcome. *Held-out gain small and not significant; length drifts* $194.4 \rightarrow 183.5$ *tokens under the cap.*

Vary exactly one factor; hold every other factor fixed.

Held fixed throughout the programme: LoRA rank 32; Adam with $\beta_1 = 0.9$, $\beta_2 = 0.95$; learning rate 1×10^{-4}; seeds $\{42, 123, 456, 789, 1024\}$; one shared reward grader and one shared decoding path across back-ends; Welch / Mann–Whitney, Cohen's d with Hedges–Olkin intervals, 95% bootstrap CIs (10,000 resamples) and a declared multiple-comparison family, all fixed in advance of the comparisons they support.

Figure 4.1: The four de-confound pillars. Each holds the whole stack fixed and varies exactly one factor.

by the single axis it varies and is only interpretable to the extent that every other axis was genuinely pinned (Figure 4.2).

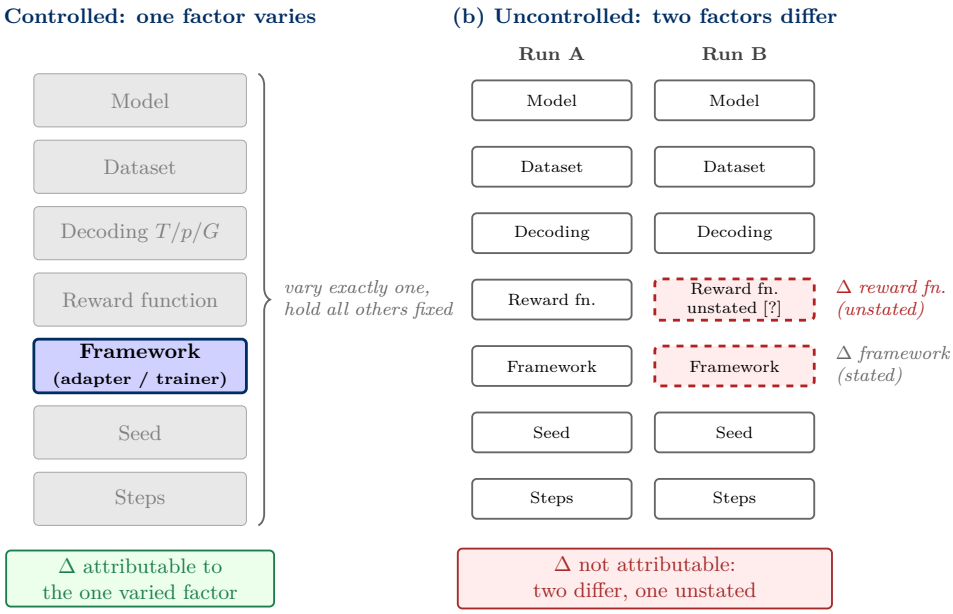


Figure 4.2: The attribution design. Exactly one factor of the training stack is varied while the remainder is held fixed, so a measured difference cannot be attributed to an unstated framework difference.

The commitment is implemented in a unified launcher, `platform_local/unified/launcher.py`, which consumes a single configuration object and dispatches it to thin per-framework adapters — `trl_integrations/`, `verl/`, `platform_modal/openrlhf/`, `platform_hybrid/skyrl/` and `platform_tinker/atropos/` — with the reference GRPO loop implemented once in `platform_tinker/tinkerrl/grpo.py`[2]. The reward grader and the decoding path are shared code. That sharing is the load-bearing part of the design: because the grader and the sampler are the same objects across back-ends, a difference in outcome between two frameworks is attributable to the framework, and not to a divergent reward parser or a differently configured decoder masquerading as an algorithmic difference. The commitment also bounds what a result may claim. A completed run supports a statement about that stack, at that revision, with that grader; it does not license a statement about "GRPO" in the abstract, and the project's reporting standard (§4.4, P5) is built around that distinction.

Four pillars carry the commitment, each holding the remainder of the stack fixed while varying one factor[3]:

[2]Source: platform_tinker/reports/esa_phase1/Phase1_Project_Report_ZVF.tex.

[3]Source: platform_hybrid/sem 4 work/submissions/mtech-final-review/build_review_pptx.py, slide 7.

Table 4.1: The four de-confound pillars, the single factor each isolates, and the design that holds the remainder fixed.

Pillar	Factor isolated	Design
1 · PPO versus GRPO, same stack	Advantage estimator only	Identical model, data and stack
2 · Zero-Variance Fraction	Signal starvation and its confounders	Per-step ZVF measured across nine libraries
3 · Trainability and group size	Contrast density against the variance knob	$G \in \{2, 4, 8, 16\} \times$ seeds, held out
4 · Held-out generalisation	Whether GRPO adds real capability	Dr. GRPO versus GRPO, pre-to-post, McNemar

Pillar 1 asks whether a reported GRPO advantage survives when the estimator is the only thing that changes; on the measured comparison the paired difference is small and not statistically significant (paired $\Delta = -0.002$, $p = 0.374$). The run record for that comparison is `Qwen/Qwen2.5-0.5B` over 40 steps, and the prose source it is attributed to names no model at all[4]. The Qwen3-8B row in the project's own effect-size table is carried as a provenance conflict and explicitly marked not estimable, so no Pillar-1 statistic is quoted here under that label[5], while the same-stack Welch test for that comparison is recorded in the historical twenty-test family at $p = 0.7605$ and flagged there as a provenance conflict rather than an estimate[6]. Pillar 2 turns a run's internal signal state into a first-class measurement; it is formalised in §4.3. Pillar 3 varies the group size G while holding model, data, reward and schedule fixed, and is the axis on which the preference-density argument of P3 rests. Pillar 4 asks the question the other three cannot answer — whether any of this changes held-out behaviour — by comparing a pre-RL checkpoint with its post-RL successor on data the training loop never saw.

Held fixed across the experiment programme are the training protocol constants: LoRA rank 32; Adam with $\beta_1 = 0.9$ and $\beta_2 = 0.95$; learning rate 1×10^{-4} for the headline configuration; and the seed set {42, 123, 456, 789,

[4]Sources: platform_hybrid/paper/sections/frontier_synthesis_scaling.tex; platform_hybrid/experiments/results/samestack_ppo_grpo.json.

[5]Source: platform_hybrid/paper/sections/_shared_methods.tex.

[6]Source: platform_hybrid/paper/sections/_shared_methods.tex.

1024}[7]. The instrumented 79-run roster described in that source spans the Qwen3, Qwen3.5, Llama-3.x, DeepSeek-V3.1 and Nemotron-120B families, from 0.6B to approximately 671B total parameters, including mixture-of-experts anchors at the frontier end of that range[8]; the wider P1 cross-scale roster adds a 1T anchor (Kimi-K2) and so spans 0.6B to 1T (Chapter 6 §6.1; source: platform_hybrid/paper/sections/p1_abstract.tex). The statistical apparatus is fixed in advance of the comparisons it supports: Welch t-tests and Mann–Whitney U tests for two-sample contrasts; Cohen's d with Hedges–Olkin intervals; 95% bootstrap confidence intervals with 10,000 resamples; rliable aggregate metrics (interquartile mean, median and optimality gap); and Bonferroni and Benjamini–Hochberg corrections over a declared test family[9]. Power was computed rather than assumed: at 80% power the minimum detectable effect for an equal-n comparison of five seeds against five is d_min = 2.024, improving to d_min = 1.530 when a five-seed group is compared against a pooled group of fifteen[10]. Those figures are a standing reminder of how coarse a five-seed design is, and the project reports several null results that are null precisely because the design could not have detected anything smaller.

Two limitations belong in this section rather than in a later discussion. First, the framework roster is broader than the framework evidence: although seven RL libraries are named in the study design, only Tinker and TRL produced completed runs, and the veRL and OpenRLHF entries in `framework_comparison.json` are dry-run placeholders rather than measurements[11]. Cross-library claims are therefore narrower than the roster implies. Second, the variance decomposition that motivates the whole attribution commitment is itself four separate marginal one-way ANOVAs, each run against the same 42 experiments with valid performance observations: library/framework $\eta^2 = 0.546$, training algorithm $\eta^2 = 0.558$, model family $\eta^2 = 0.471$, model scale $\eta^2 = 0.322$[12]. These four values sum to 1.897. A set of shares that sums above one is not a partition of a common variance budget, and on a corpus where the factors are collinear a marginal η^2 absorbs the variance of each factor's correlates. The decomposition is therefore reported here as evidence that framework and algorithm vary a great deal across this corpus, which is what motivates holding them fixed — not as an estimate of any factor's independent contribution, which a marginal one-way analysis on a non-orthogonal design cannot identify. The historical twenty-test multiple-comparison family that

[7]Source: platform_hybrid/paper/neurips_2026_variants/paper_P8_workshop.tex.
[8]Source: platform_hybrid/paper/neurips_2026_variants/paper_P8_workshop.tex.
[9]Source: platform_hybrid/paper/sections/_shared_methods.tex.
[10]Source: platform_hybrid/paper/sections/_shared_methods.tex.
[11]Source: platform_hybrid/paper/sections/_shared_methods.tex.
[12]Source: platform_hybrid/paper/sections/_shared_methods.tex.

accompanies it pre-dates the Tier-A/B/C partition, silently includes single-seed rows whose Welch statistics are mathematically undefined, and reduces to a single surviving comparison under the restricted family; it is retained as a reproducibility audit trail and explicitly not as a corrected survival claim[13].

4.2 Per-Step Telemetry

Attribution is only as good as the run's observability. Every training step therefore emits a record containing the Zero-Variance Fraction; gradient utilisation; the collapse decomposition that separates all-correct from all-wrong zero-variance groups; and, alongside them, reward, loss, entropy, completion length and the KL divergence to the reference policy[14]. Records are written to JSONL and mirrored to Weights & Biases, so a run can be analysed locally without a network dependency and reviewed remotely without a local environment. The reason ZVF and its decomposition are logged as separate fields rather than collapsed into a single starvation number is that the same aggregate fraction can describe opposite situations: a group of completions that are all correct and a group that are all wrong both contribute zero within-group variance, and the remedy for each is different. Logging the decomposition preserves that distinction at the point of collection instead of trying to recover it afterwards (Figure 4.3).

The quantity being logged is the group-relative advantage actually used by the runner,

A_i = (r_i − μ_g) / (σ_g + ε), μ_g = (1/G) Σ_j r_j, σ_g = sqrt((1/G) Σ_j (r_j − μ_g)^2)

[15]. When σ_g = 0 every A_i is zero, so the whole group contributes no gradient through the advantage (surrogate) term. The parenthetical qualification matters and is preserved throughout this thesis: it is the advantage term that vanishes, not necessarily the total update, because the KL penalty and any auxiliary loss or clipping term are unaffected[16].

Two further design choices distinguish this telemetry from conventional training curves. The first is that the raw per-group reward tensors are retained, not only the summaries computed from them. This makes a published number auditable by recomputation: any downstream diagnostic, such as `p2_collapse_analysis.py`, can reproduce a reported statistic from the stored tensors rather than trusting

[13]Source: platform_hybrid/paper/sections/_shared_methods.tex.
[14]Source: platform_tinker/reports/esa_phase1/Phase1_Project_Report_ZVF.tex.
[15]Source: platform_tinker/reports/esa_phase1/Phase1_Project_Report_ZVF.tex.
[16]Source: platform_tinker/reports/esa_phase1/Phase1_Project_Report_ZVF.tex.

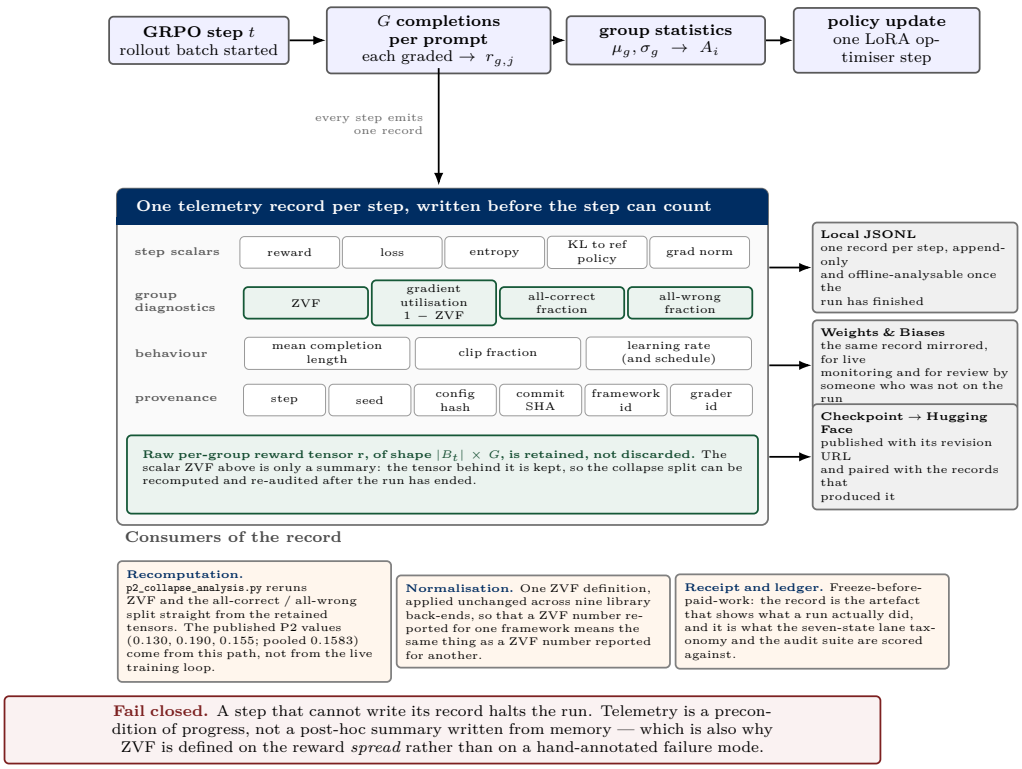


Figure 4.3: Per-step telemetry recorded by the unified harness, and the destinations each record flows to.

a logged summary[17]. The convention has been exercised. Study P2 recomputed three stored GSM8K group tensors from their raw values and obtained 0.130, 0.190 and 0.155, pooling to 0.1583 — figures reproduced from the tensors rather than read back from a log line[18]. The second is that the logging layer is treated as part of the apparatus and is expected to fail. An early Weights & Biases integration used the library's thread-unsafe global run object from multiple worker threads, which produced repeated "Run is finished" summary-write failures; the fix was to give each run its own process in `campaign.py`[19]. The defect is recorded because telemetry that silently drops writes is indistinguishable from telemetry showing a healthy run, and the remedy — process-level isolation per run — is a structural property of the campaign rather than a patch.

4.3 Formalising the Zero-Variance Fraction

The Zero-Variance Fraction is the share of prompt groups in a batch whose reward variance is zero. Formally, for a batch B_t of prompt groups at step t, each group

[17]Source: platform_tinker/reports/esa_phase1/Phase1_Project_Report_ZVF.tex.
[18]Source: platform_hybrid/paper/sections/p2_abstract.tex.
[19]Source: platform_tinker/reports/esa_phase1/Phase1_Project_Report_ZVF.tex.

g contributing K sampled completions with rewards r_{g,1}, …, r_{g,K},

ZVF_t = (1 / |B_t|) Σ_{g ∈ B_t} 1[Vaŕ(r_{g,1}, …, r_{g,K}) ≤ ε]

where Vaŕ is the unbiased sample variance and $\varepsilon = 10^{-6}$ for rewards normalised to [0, 1][20]. A prompt-set formulation used elsewhere in the paper sources writes the same quantity as ZVF_t = (1 / |P_t|) Σ_{p ∈ P_t} 1[Var_{c ~ C(p)} r(c) = 0], with C(p) the set of sampled completions for prompt p[21]. The reference implementation is deliberately small, so that the diagnostic cannot acquire hidden behaviour:

```
def zvf(rewards_2d: np.ndarray, eps: float = 1e-6) -> float:
    # rewards_2d shape: [num_groups, K]
    var_per_group = rewards_2d.var(axis=-1, ddof=1)
    return float((var_per_group <= eps).mean())
```

[22]. Gradient utilisation is then defined as its complement, GU_t = 1 − ZVF_t[23]. The complement is a descriptive accounting identity, not a claim about optimiser behaviour. Its scope must be read against the runner actually used, and that reading is stronger than the source's generic caveat suggests. The objectives the source disclaims — those with KL, auxiliary losses, clipping or token masking — are not the objectives this project measured: the group-size sweep's own driver is annotated "no KL/clip", the project records that its runner "omits the PPO ratio/clip, the frozen reference policy, and the completion-only token mask", and the loss is a single advantage-weighted term summed over tokens[24]. For that objective ZVF_t = 1 does entail a zero step gradient through the policy term, and the escapes that remain are named rather than implied: optimiser momentum carrying stale updates from earlier steps, and the per-batch and per-token normalisers. The quoted sentence from paper_P8_workshop.tex applies to objectives that carry those terms, and is not the licence for the measured configuration (Figures 4.4 and 4.5).

Three properties make the definition more than a restatement of the training loop. First, it is non-tautological under a simple null: if completions were independent Bernoulli draws with per-prompt success probability p_g, then E[ZVF_t] = E_{p_g}[p_g^K + (1 − p_g)^K], a quantity that depends on the model's accuracy distribution and the group size rather than on the observed batch[25]. The

[20]Source: platform_hybrid/paper/sections/appendix_zvf_formalization.tex.
[21]Source: platform_hybrid/paper/neurips_2026_variants/paper_P8_workshop.tex.
[22]Source: platform_hybrid/paper/sections/zvf_pipeline_spec.tex.
[23]Source: platform_hybrid/paper/neurips_2026_variants/paper_P8_workshop.tex.
[24]Sources: platform_hybrid/experiments/modal/modal_groupsize_zvf_sweep.py; platform_hybrid/paper/paper.tex.
[25]Source: platform_hybrid/paper/sections/appendix_zvf_formalization.tex.

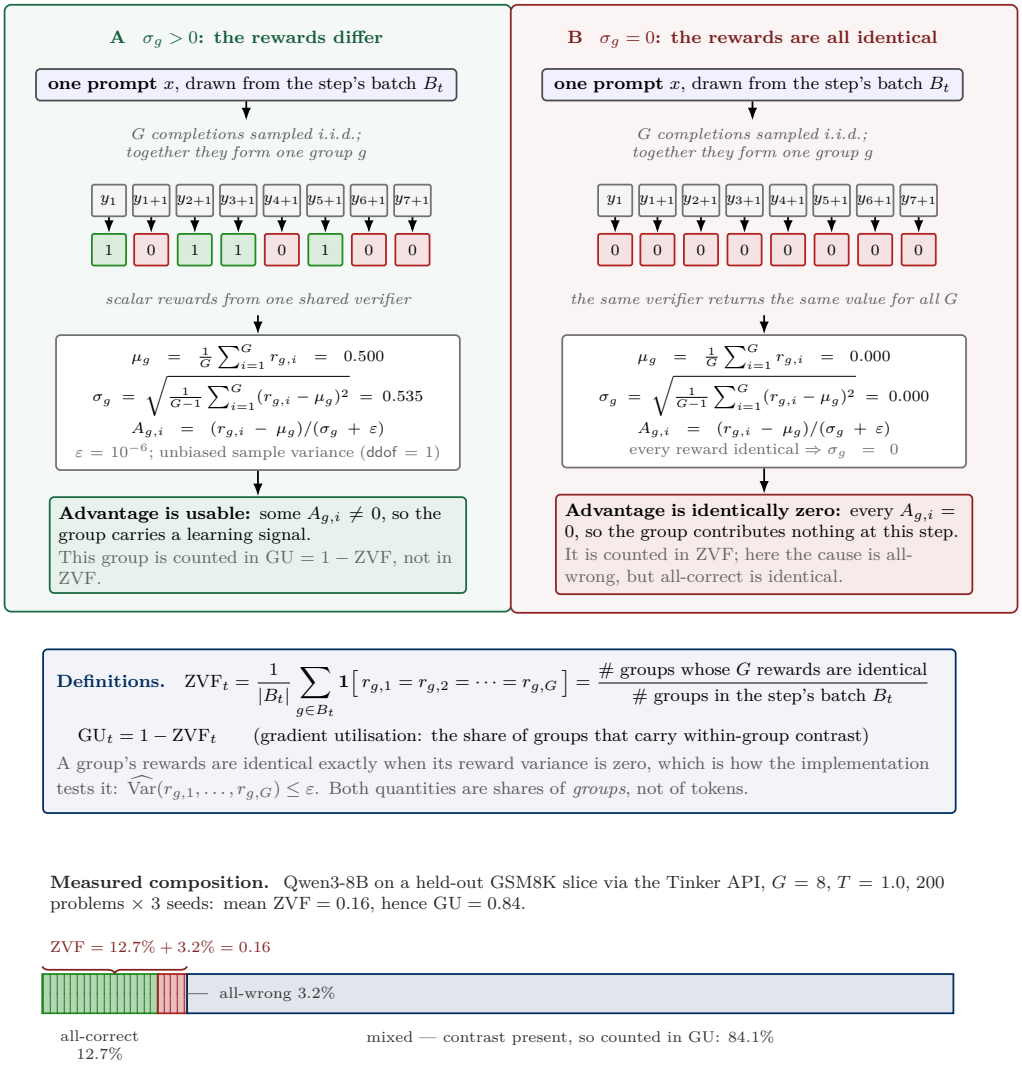


Figure 4.4: Definition of the Zero-Variance Fraction. One prompt produces G completions with scalar rewards; when the within-group reward standard deviation is zero the advantage is identically zero and the sample contributes no gradient. ZVF is the fraction of such groups.

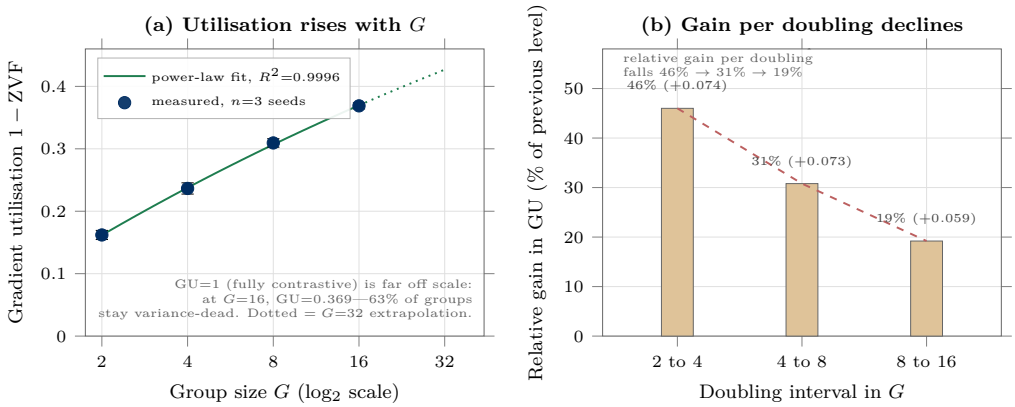


Figure 4.5: Gradient utilisation, $1-\text{ZVF}$, as a function of group size G. Increasing G reduces the zero-variance fraction with saturating benefit, which is why larger groups are not uniformly better.

identity pass@G − p^G = 1 − ZVF follows algebraically under the same i.i.d.-Bernoulli model and is therefore reported as an algebraic remark rather than as evidence[26]. Second, it was validated against that null on instrumented runs: a Qwen2.5-0.5B sweep over G ∈ {2, 4, 8, 16}, three seeds and 40 steps — twelve runs and 480 logged steps — found per-step observed ZVF correlating at r = 0.92 with the Bernoulli-null prediction p^G + (1 − p)^G, with mean ZVF of 0.84, 0.76, 0.69 and 0.63 at G = 2, 4, 8 and 16 respectively[27]. Third, at frontier scale the same instrument was run on Qwen3-8B against held-out GSM8K through the Tinker managed API at G = 8, temperature 1.0, over 200 problems and three seeds, yielding a mean ZVF of 0.16 composed of 12.7% all-correct groups, 3.2% all-wrong groups and 84% mixed groups; the raw correlation with the mechanical null is approximately r = 0.93 with rank $\rho \approx 0.65$, and the i.i.d. null over-predicts (approximately 0.28 against the observed 0.16), which is the expected direction when the reward signal is real rather than noise[28].

That the quantity is portable across libraries is a separate, weaker claim, and the project states its limits explicitly. A normaliser at `scripts/zvf_compute_cross_framework.py` maps each framework's reward and mask fields onto the common definition — TRL's `rewards` with `batch_size / group_size` and `completion_mask`; Tinker's `rollout.reward` with `rollout.group_id` and `rollout.eos_mask`; OpenRLHF's `reward_score` with `prompt_index` and `attention_mask`; veRL's `data_source/reward` with `uid` and `response_mask` — and emits JSONL records of the form `{step, zvf, batch_reward_mean, n_`

[26]Source: platform_hybrid/paper/sections/p2_abstract.tex.
[27]Source: platform_hybrid/paper/sections/appendix_zvf_formalization.tex.
[28]Source: platform_hybrid/paper/sections/appendix_zvf_formalization.tex.

`groups, K}`[29]. The scope correction attached to that parser is that it normalises reward logs and does not make its runner canonical GRPO; because the project does not bundle raw per-group rollout JSONL for every framework, no fresh four-way pointwise ZVF overlay is claimed, only the narrower implementation-level portability claim[30]. Sensitivity to the threshold ε is bounded: the reported ε-sensitivity is at most 0.4%[31]. Boundary conditions are declared where the diagnostic does not apply: dense or continuous reward signals, K = 1, SFT-saturated baselines and non-outcome-reward RL all fall outside the setting in which the definition is meaningful[32].

The methodological point of this section is that ZVF is a diagnostic and not a metric, and the project defends that position with its own negative evidence rather than with a general caution. Four observations support it. First, the same ZVF value can describe mastery or incapacity: in a 368-run Weights & Biases audit across seven models, ZVF sits at 0.95–0.97 at both reward extremes and at 0.25–0.29 at mid-range rewards of 0.35–0.50, so a high value is as consistent with a solved task as with an unsolved one[33]. Second, the pooled association between ZVF and performance, Pearson $r = -0.769$ ($p = 0.0008$) with Spearman $\rho = -0.784$ ($p = 0.0005$) over $N = 15$ experiments spanning five model families, is confounded by task type: tool-use experiments reach ZVF = 100% at approximately 0% reward while GSM8K experiments average ZVF = 8.5%, and within GSM8K the association weakens to $r = +0.40$ ($p = 0.09$, $N = 19$) and then to $r = +0.13$ ($p = 0.79$) when restricted to the seven model-varying runs[34]. A statistic whose sign is set by which tasks are included in the pool is not a score. Third, ZVF does not track final accuracy ($r \approx 0.09$, $n = 12$), and P2 reports only a weak association with final held-out outcome ($\rho \approx 0.27$) while deliberately declining to promote ZVF to a causal or incrementally predictive statistic[35]. Fourth, no step-level p-value is computed on the 480 autocorrelated steps of the instrumented sweep, because treating correlated within-run steps as independent samples would manufacture significance[36]. One further boundary is recorded for the same reason: a variance-mitigation table in the paper sources is a declared simulation projection and not evidence, and its rows should be read as synthetic rather than as inde-

[29]Source: platform_hybrid/paper/sections/zvf_pipeline_spec.tex.
[30]Source: platform_hybrid/paper/sections/zvf_pipeline_spec.tex.
[31]Source: platform_hybrid/paper/sections/zvf_pipeline_spec.tex.
[32]Source: platform_hybrid/paper/sections/appendix_zvf_formalization.tex.
[33]Source: platform_hybrid/paper/sections/p7_abstract.tex.
[34]Source: platform_hybrid/paper/neurips_2026_variants/paper_P8_workshop.tex.
[35]Source: platform_hybrid/paper/sections/appendix_zvf_formalization.tex; platform_hybrid/paper/sections/p2_abstract.tex.
[36]Source: platform_hybrid/paper/sections/appendix_zvf_formalization.tex.

pendent library deployments[37]. The result of these choices is a diagnostic that is useful for describing where within a batch the learning signal is and is not, and that is explicitly not proposed as a quantity to be maximised, ranked or reported as a headline score.

4.4 The Eight-Study Portfolio

Semester 3 delivered the shared code and the initial experimental foundation; the Semester 4 programme — the research direction, the expanded experiments, the analysis and the P1–P8 papers — is the present author's solo continuation, and the roster records 206 analysis iterations behind the numbers reported in this thesis[38]. The portfolio is organised as eight studies, each asking one question and each carrying its own claim boundary. The questions and boundaries are stated here because they are methodological commitments: a study's question determines which comparisons it may make, and the boundary states what it declines to conclude.

P1, "Scaling Laws for GRPO Post-Training of Language Models: A Cross-Library, Cross-Scale Study", asks whether GRPO post-training outcomes scale with model scale in a way that can be fitted and extrapolated[39]. Its answer is a conservative negative one: across more than seventy runs and seven RL libraries over five model families from 0.6B to 1T parameters, the cross-scale slope is flat over roughly 2.4 orders of magnitude; four of five frontier-scale traces have their saturation rate pinned at the fit boundary and are unidentifiable; the internal three-phase geometric check matches only 2 of 12 anchors; and the Nemotron-120B trace shows a zero-reward step fraction of 0.55 against at most 0.067 elsewhere[40]. The frontier anchors — Qwen3-235B-MoE, DeepSeek-V3.1, Kimi-K2 and Nemotron-120B — were run through the Tinker managed API in a single-seed, descriptive-only regime, and the paper's claim boundary is a descriptive limits-and-identifiability audit rather than a positive scaling law; matched multi-seed cross-scale runs are an explicit non-claim[41].

P2, "The Zero-Variance Fraction: A Descriptive Diagnostic for Signal Starvation in GRPO", asks how much of a training batch carries usable group-relative signal,

[37]Source: platform_hybrid/paper/sections/p2_abstract.tex; platform_hybrid/paper/sections/zvf.tex.

[38]Source: platform_hybrid/sem 4 work/README.md.

[39]Source: platform_hybrid/sem 4 work/README.md; platform_hybrid/paper/sections/p1_abstract.tex.

[40]Source: platform_hybrid/paper/sections/p1_abstract.tex.

[41]Source: platform_hybrid/paper/sections/p1_abstract.tex.

and whether that fraction predicts the final outcome[42]. The study characterises ZVF descriptively, establishes the algebraic identity of §4.3 as algebraic rather than empirical, reports the weak association with final held-out outcome ($\rho \approx 0.27$), and marks the variance-mitigation table as a simulation projection; the retired roster item that previously carried a causal framing now lives under `archive/absorbed/R02_main_zvf/`[43].

P3, "Group Size in GRPO: Contrast Density and the Bridge to DPO", asks how trainability varies with the group size G and whether GRPO admits an all-pairs preference interpretation[44]. Two results anchor it. The first is that trainability varies non-monotonically with G, with intermediate sizes often outperforming both the smallest and the largest settings, although the data do not justify a universal optimum; measured equivalence tests extend to $G = 16$, retained held-out accuracy is essentially flat across $G \in \{2, \ldots, 16\}$ on the near-ceiling arithmetic task with a retention ratio of approximately 1.00 between $G = 2$ and $G = 16$ (a ratio, not an accuracy above 100%), and effective signal-to-noise rises only sublinearly with G, at approximately 52% of the $\sqrt{G}$ ideal; no directly trained matched-budget $G = 32$ cell exists, and the $G = 32$ values come from an explicitly labelled reconstructed token-budget grid and are hypotheses rather than measurements[45]. A 505-task conditional utility audit using $(1 - \mathrm{ZVF})/\sqrt{G}$ selects $G = 4$ over $G = 5$ by 0.00177 with a 95% bootstrap interval of [0.00082, 0.00270], conditional on cohort and objective[46]. The second is that GRPO admits an all-pairs preference interpretation resembling DPO, offered as an interpretation and not as a proven equivalence[47].

P4, "Length Bias and Held-Out Generalization in GRPO and Dr. GRPO", asks whether the verbosity trap associated with GRPO appears in the measured regime and whether either method changes held-out accuracy[48]. The headline is a carefully scoped negative: in the short-horizon GSM8K chain-of-thought regime with a 200-token generation cap, the mean Qwen3-8B GRPO gain over its pre-RL checkpoint on held-out data is small and not statistically significant, and GRPO and Dr. GRPO are indistinguishable at the seed budget used; neither inflates length within the capped regime, with pooled mean completion length drifting from 194.4 to 183.5 tokens. The 200-token cap bounds the claim, and the absence of the verbosity signature is the expected outcome when length is already controlled rather

[42]Source: platform_hybrid/paper/sections/p2_abstract.tex.
[43]Source: platform_hybrid/paper/sections/p2_abstract.tex.
[44]Source: platform_hybrid/paper/sections/p3_abstract.tex.
[45]Source: platform_hybrid/paper/sections/p3_abstract.tex.
[46]Source: platform_hybrid/paper/sections/p3_abstract.tex.
[47]Source: platform_hybrid/paper/sections/p3_abstract.tex.
[48]Source: platform_hybrid/paper/sections/p4_abstract.tex.

than a positive finding[49].

P5, "Report the Stack, Not the Label: RL-for-LLM Results Are Stack-Conditioned", asks how much of a reported result is attributable to the stack rather than to the algorithm label, and what a reportable stack must contain[50]. Three exhibits carry the argument: a nominally matched-configuration backend audit spanning final training rewards from 5.0% to 85.6%, a seventeen-fold same-label swing, in which the managed backend also silently pinned a different base checkpoint — undisclosed bundling being part of the exhibit rather than a footnote to it; a "DAPO" comparison in which true dynamic sampling on an open trainer produced a mean ZVF of 0.00 while the same label approximated on a closed stack through an asymmetric-clip surrogate produced a mean ZVF of 0.58; and a twelve-cell head-to-head of four labelled algorithms on one fixed stack landing within a 0.034 band (mean last-ten reward 0.710–0.744)[51]. The study proposes an eight-item minimum reportable stack — loss form; reference-policy and KL handling; sampler, backend and precision including base-checkpoint identity by revision or hash; per-step ZVF and gradient-utilisation trajectory; group-size schedule; a disjoint held-out split; decontamination plus a parser-robustness probe; and held-out pass@k curves alongside pass@1 — and formalises an RL-reproducibility threat model with flip-risk grades R0–R5[52]. This is the standard this thesis is written to.

P6, "GRPO-Registry: A Machine-Readable Catalog of Group-Relative RL Stacks and Their Variant Deltas", asks whether stacks can be compared mechanically rather than through prose[53]. It defines seven-field stack records corresponding to the MIN-REPORT-RL manifest block, a per-component variant-delta status, three variant-delta records verified against the DAPO, Dr. GRPO and GSPO papers, twenty seed stack entries, and a reference CLI with a 0–100 MIN-REPORT auditor badge spanning 43–96 across the seed entries and a `stackdiff` command that maps entries to an R0–R5 flip-risk verdict, flagging the open-versus-closed "DAPO" pair as R5[54]. The study is explicitly a resource paper rather than a second ranking, and it cites the seventeen-fold exhibit only to motivate the schema[55].

P7, "From Diagnostic to Controller: A Signal-Starvation Theory of GRPO and an Adaptive Group-Size Intervention", asks whether the diagnostic of §4.3 can be turned into a control signal[56]. It models usable learning signal per prompt as

[49]Source: platform_hybrid/paper/sections/p4_abstract.tex.
[50]Source: platform_hybrid/paper/sections/p5_abstract.tex.
[51]Source: platform_hybrid/paper/sections/p5_abstract.tex.
[52]Source: platform_hybrid/paper/sections/p5_abstract.tex.
[53]Source: platform_hybrid/paper/sections/p6_abstract.tex.
[54]Source: platform_hybrid/paper/sections/p6_abstract.tex.
[55]Source: platform_hybrid/paper/sections/p6_abstract.tex.
[56]Source: platform_hybrid/paper/sections/p7_abstract.tex.

$S = p(1 - p)(1 - h_G(p))$ with $h_G(p) = p\hat{}G + (1 - p)\hat{}G$, yielding three testable consequences: that the expected ZVF indicator equals $p\hat{}G + (1 - p)\hat{}G$; that larger G lowers ZVF; and that gradient magnitude tracks $p(1 - p)$[57]. The empirical checks reproduce the predicted U-shape in the 368-run audit; a model-by-group-size grid confirms the monotone G effect at interior accuracy (Qwen3-32B: 0.33, 0.18 and 0.08 at G = 4, 8 and 16); and a toy-scale open-backward-pass experiment finds $\mathrm{corr}(\|\nabla\|, p(1 - p)) = +0.71$ on a 0.5B model with synthetic arithmetic, reported as directional only[58]. The study promotes pairwise-contrast density, $\mathrm{PCD} = (G - 1)/G \cdot E[p(1 - p)]$, as the control signal, and validates it by falsification: a micro-jitter intervention collapses batch ZVF from 0.158 to 0.000 while leaving PCD unchanged, showing that the two quantities are not measuring the same thing[59]. Adaptive group size matches the best fixed-recipe held-out gain (+0.575) at 186 rollouts, while DAPO-style dynamic sampling drives ZVF to 0.00 at a 45% rollout cost increase; in the 2,560-observation controller trace, 1,723 of 1,867 escalation events (92.3%) occur on all-correct groups and only 144 on all-wrong groups[60]. The methodological commitment that governs this study is stated at its head: adaptive group size is not promoted, and no result establishes prospective superiority over static G = 16, Dr. GRPO or dynamic sampling[61].

P8 occupies the eighth slot with a study outside the RL programme, "LLM vs. XGBoost in Credit-Card Fraud: Sensor and Scribe, Not Scorer"[62]. Its question is whether a large language model should be used as the fraud scorer or as a component around one, and it is answered with tabular baselines reported at AUC 0.7955, F1 0.356, precision 0.723 and recall 0.236 for the boosted-tree model, against a Qwen3.5-4B scorer at accuracy 0.792 with AUC 0.483, i.e. chance-level ranking, at roughly eighty-five times lower cost as a triage component[63]. The roster evolved after the semester-4 title list was fixed, and the present position is recorded honestly rather than smoothed over: the fraud study is parked outside the ZVF thesis, contributes no RL evidence, and is retained for its measurement discipline; the active P8 slot in the current paper roster is an exploratory workshop artifact[64]. No result from P8 is used anywhere in this thesis to support an RL claim.

[57]Source: platform_hybrid/paper/sections/p7_abstract.tex.
[58]Source: platform_hybrid/paper/sections/p7_abstract.tex.
[59]Source: platform_hybrid/paper/sections/p7_abstract.tex.
[60]Source: platform_hybrid/paper/sections/p7_abstract.tex.
[61]Source: platform_hybrid/paper/sections/p7_abstract.tex.
[62]Source: platform_hybrid/sem 4 work/README.md.
[63]Source: platform_hybrid/sem 4 work/submissions/mtech-final-review/README.md; platform_hybrid/paper/archive/absorbed/P08_fraud/sections/p8_abstract.tex.
[64]Source: platform_hybrid/paper/PAPERS_README.md.

4.5 The Design of the E1–E14 Held-Out Evaluation Campaign

The evaluation campaign is the applied half of the programme and is deliberately narrow in a way that makes its results interpretable: it evaluates one actor against fourteen held-out benchmark suites under a receipt, budget and decontamination regime, instead of training many actors and comparing them. The single actor is the base model `Qwen/Qwen3.6-35B-A3B` at commit `995ad96e…`, with adapter `arvindcr4/pavlov-portfolio-qwen36-seed809-stepfinal-tinker-cf0ad8c1…` at commit `64444133…`, in bfloat16[65]. One actor fixes the object of study: every reported per-suite figure describes the same frozen policy, so a difference between two suites is a property of the suite and not of a model swap. The design consequence is that the campaign cannot support model-comparison claims at all, and it makes none (Figure 4.6).

Evaluators are the suites' own scorers, used natively rather than reimplemented, and they are pinned by revision. The surviving native runners live under `zvf-program/flagship/`, including `e1_swe_bench_pro_full_eval.py`, `e11_verilog_eval_decontamination.py`, `eval_pavlov_sdab.py`, `eval_pavlov_xlam.py`, `mle_bench_eval.py` and `eval_apex_agents.py`[66]. The evaluation-side revisions pinned for the decontamination audit are LAB-Bench `998a8e0a40cf116c80e1b0e7a805ebb5fb9fa838`, AgentDojo `089ed468cf3ed0322acc66b0211f26d9d90dbf60` and Omni-MATH `40ba231d8f16e29ecd40e6407e2c8640145a8f62`, with the training-side envelope pinned at API-Bank-RLVR `bf67c42626f02c305514b1df16dcabc5fc616333` (597 rows, MIT) and SWE-Gym `bb94ed9e39bbeb96a7fcbfb533b80f25a7fd59cb` (2,438 train rows, MIT)[67]. Using the native scorer is a substantive choice and not a convenience: when the Omni-MATH lane produced two rows whose judged outputs were truncated before any equivalence verdict, the native scorer's omission of missing-verdict rows was adjudicated as correct official behaviour rather than a parser bug, and the lane closed with no re-judge because re-judging would deviate from the official scorer and generate new evidence with no effect on the score[68].

Sampling is frozen rather than merely recorded: temperature 0.1, top-p 0.95, one sample per row, a maximum of 128 response tokens, with the service seed fixed where the provider supports it and otherwise recorded in the receipt[69]. Prompt and

[65]Source: outputs/E1_E14_FINAL_RESULTS_2026-09-19.md.
[66]Source: zvf-program/flagship/.
[67]Source: outputs/public_portfolio_2026-09-05/decontamination/README.md.
[68]Source: outputs/E1_E14_FINAL_RESULTS_2026-09-19.md.
[69]Source: zvf-program/flagship/PAVLOV_EXPERIMENT_PROTOCOL_2026-08-09.md.

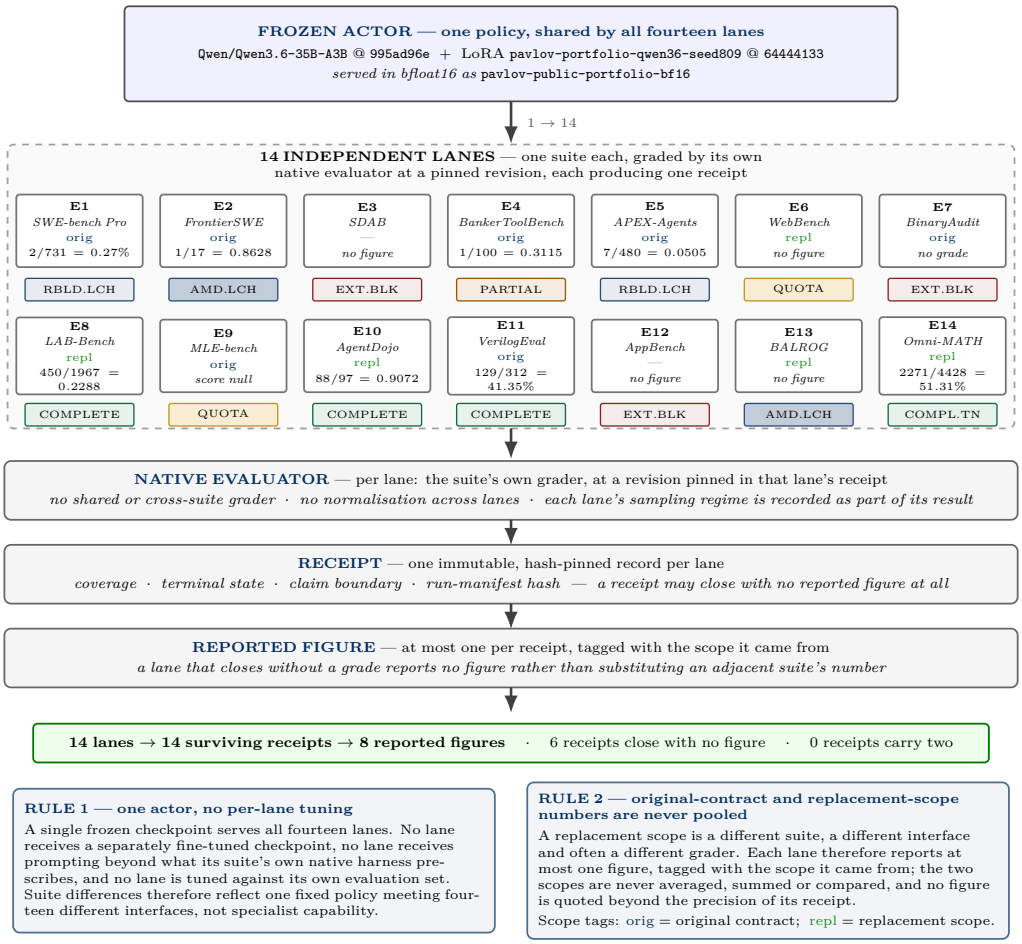


Figure 4.6: Design of the E1–E14 held-out evaluation campaign: one frozen actor, each suite graded by its own native evaluator at a pinned revision, with original-contract and replacement-scope results kept separate.

response limits are 1,200 and 128 tokens respectively, and the same source requires the strict verifier implementation to be hashed before paid work, with a perfect call defined as a verifier score of exactly 1.0 and no parser, target, split or verifier change permitted mid-campaign[70]. Freezing the sampler addresses the failure mode in which a benchmark score quietly improves because decoding became more favourable between runs; it also fixes the comparison between the base reference and the adapter, which is evaluated side by side on untouched manifests[71].

The receipt model is freeze-before-paid-work, and it is the part of the campaign design that carries the most methodological weight. Before any paid phase a run must record an immutable base-model revision, tokenizer revision, container or environment digest and adapter revision; the selection slice and the disjoint hold-out are re-created from a pinned dataset revision and sealed, are write-once, are not mounted in training, and the word "holdout" is reserved for a suite or manifest only once its split, task-ID, license, runtime and decontamination receipts are complete[72]. The selection slice is used for arm decisions and is therefore unsuitable for confirmatory claims; the final evaluation manifests are never used to choose an arm, alter a prompt, or re-run a failed suite with a more favourable split[73]. Tracking is verified before spend rather than after: an online Weights & Biases run must be initialised and its Tinker run identifier recorded before the first paid step, every step records loss, reward, configuration, seed, split identity and cumulative cost, and every checkpoint is published to Hugging Face with a unique repository or revision URL, with a failure of any of these stopping the run fail-closed[74]. If a receipt is missing, the phase is blocked rather than silently shortened[75]. At handoff the protocol requires four separate evidence classes to be reported apart: completed receipts; operational and provenance receipts; prospective gates and budget projections; and admissible final results, the last existing only if every gate passed[76].

That last distinction has teeth in practice, and the campaign's disposition record shows it. The portfolio-level protocol defines a predeclared equal-domain macro average over sixteen domains, a conjunctive coverage gate, an integrity gate, a global improvement gate requiring the lower bound of a one-sided 95% interval for the portfolio difference to exceed zero, a domain no-regression guard with a predeclared five-percentage-point tolerance, and a safety guard bounding any

[70]Source: zvf-program/flagship/PAVLOV_EXPERIMENT_PROTOCOL_2026-08-09.md.
[71]Source: zvf-program/flagship/PAVLOV_EXPERIMENT_PROTOCOL_2026-08-09.md.
[72]Source: zvf-program/flagship/PAVLOV_EXPERIMENT_PROTOCOL_2026-08-09.md.
[73]Source: zvf-program/flagship/PAVLOV_EXPERIMENT_PROTOCOL_2026-08-09.md.
[74]Source: zvf-program/flagship/PAVLOV_EXPERIMENT_PROTOCOL_2026-08-09.md.
[75]Source: zvf-program/flagship/PAVLOV_EXPERIMENT_PROTOCOL_2026-08-09.md.
[76]Source: zvf-program/flagship/PAVLOV_EXPERIMENT_PROTOCOL_2026-08-09.md.

violation increase by two percentage points — and it states that a cost cap may not be met by dropping domains or suites, and that a suite or domain which is missing, underpowered, contaminated or outside the cost cap must be labelled incomplete or blocked, "not negative and not positive"[77]. The portfolio training phase was never launched, so the campaign does not report a portfolio result; the recorded outcome is a set of per-lane evaluations of the one adapter, and the protocol's own instruction that a partial subset is not a portfolio result and may not be relabelled as one is honoured throughout[78].

Lane dispositions are recorded in a seven-state taxonomy rather than as pass or fail: COMPLETE, for lanes with verified full-coverage results and no open gate (E8 public, E10 benign, E11); COMPLETE_TERMINAL_NOTE, for a lane complete under the native protocol with a recorded terminal note (E14 replacement); CLOSED_EXTERNAL, for lanes with no local or paid path, closed by written closure records (E3, E7, E8-private, E10-private, E12, E14-private); PENDING_QUOTA, for lanes gated on cloud quota resolution alone (E6, E9); AMENDMENT_ACCEPTED_LAUNCH_PENDING (E2, E13); REBUILD_READY_LAUNCH_PENDING (E1 wave10, E5 successor27); and CLOSED_PARTIAL (E4)[79]. The taxonomy exists so that a lane which has not been measured is not silently absorbed into a percentage. Carried alongside it is the project's evidence rule, stated in the ledger of record and repeated in every results artefact: original-contract and replacement-scope numbers are never pooled[80]. The rule is not cosmetic. The original-contract E1 lane and its replacement scope are different suites with different task populations, and a table that added them would produce a number that corresponds to no benchmark.

Decontamination is handled as a declared deterministic procedure with its own stated limits. The check hashes complete UTF-8 fields and then normalised fields with SHA-256; for fields of at least eight tokens it hashes each consecutive eight-token sequence and computes exact set Jaccard similarity for every training and evaluation pair sharing a shingle; and it tests whether an entire normalised evaluation field is a token-bounded contiguous substring of a training field[81]. Against the pinned training envelope, the per-suite inventories were 1,967 LAB-Bench tasks over 13,113 text fields with 5,105 eligible for eight-token near checks, 97 AgentDojo tasks over 71,966 fields with 9,382 eligible, and 4,428 Omni-MATH tasks over 13,284 fields with 9,394 eligible; the count of near pairs at or above

[77]Source: zvf-program/flagship/PAVLOV_EXPERIMENT_PROTOCOL_2026-08-09.md.
[78]Source: zvf-program/flagship/PAVLOV_EXPERIMENT_PROTOCOL_2026-08-09.md.
[79]Source: outputs/PES_Phase2_Review_2026-09-12/finish/Pending_Experiments.md.
[80]Source: outputs/PES_Phase2_Review_2026-09-12/finish/Pending_Experiments.md.
[81]Source: outputs/public_portfolio_2026-09-05/decontamination/README.md.

0.80 was zero in all three suites, and the maximum eight-gram Jaccard similarity was 0 for LAB-Bench and AgentDojo and 0.13414634146341464 for Omni-MATH, below the declared 0.80 threshold, with all 1,338 candidate pairs sharing an eight-gram examined[82]. Short exact matches of zero to three normalised tokens were retained and enumerated rather than dropped — the longest examples being seven characters in the LAB and Omni inventories and at most seventeen raw characters in AgentDojo, such as "Paris" and "Conference Room A" — and the audit states plainly that these collisions do not demonstrate duplication of a task but do make an unqualified "zero overlap" claim incorrect[83]. The audit's scope limits are recorded with equal care: it does not reconstruct the exact 512 consumed training records, because the dirty training transform and consumed task inventory were not recorded; it establishes a content comparison against the verified historical source envelope rather than proving global pretraining cleanliness, semantic disjointness, character-edit disjointness or image cleanliness; ASCII normalisation loses mathematical punctuation and non-ASCII text; and fields shorter than eight tokens fall outside the near-match and containment rule[84]. The audit itself is receipted: it re-verifies Weights & Biases run `bsv8vx04` against `training_identity_verified.json`, matches the 23,733 original source-field document hashes to the historical E11 receipt, adds 2,507 decoded nested API-Bank strings to reach 26,240 source documents, and records 43,977,748 bytes downloaded against a 500 MiB cap, with the checker not having enumerated the roughly 2.58 billion possible document pairs[85].

Two features of the campaign's execution are methodological rather than incidental. The first is that a lane stops when its receipts are incomplete, and the record says why. A single inventory pass on 2026-09-21 checked every blocked lane live and found concrete, named failures: E1 lacked W&B authentication, a sealed reservation and a live actor endpoint; E2 awaited an owner IAM binding that only the project owner can create; E5 lacked both a bound authorization receipt and any real native runtime adapter in the repository; E13's request remained a draft with launch authorisation false; E6 and E9 were refused by cloud quota in both regions[86]. Recording a lane as blocked-with-reason, rather than reporting a shortened run, is the same principle as the terminal-state taxonomy applied to execution. The second is that the campaign's largest lane is reported as a negative result. E1 on SWE-bench Pro produced 2/731 resolutions, or 0.274% pass@1, with 731 terminal generations, 713 native evaluations, and 14 generation failures and

[82]Source: outputs/public_portfolio_2026-09-05/decontamination/README.md.
[83]Source: outputs/public_portfolio_2026-09-05/decontamination/README.md.
[84]Source: outputs/public_portfolio_2026-09-05/decontamination/README.md.
[85]Source: outputs/public_portfolio_2026-09-05/decontamination/README.md.
[86]Source: outputs/UNBLOCK_CARRYOUT_2026-09-21.md.

4 artifact losses retained in the denominator rather than excluded[87]. A separate authorized rerun of the base model on the E4 finance suite completed 100 of 100 trials with a mean reward of 0.0, and the audit of all one hundred trajectories attributed the result to a mechanism rather than to reasoning: agents terminate after two to fourteen steps with a median of about six and no deliverables, so the verifier short-circuits to zero, and 27 or more trajectories end in role-token degeneration; the source's own reading is that this zero measures tool-dialogue collapse in the base model and not finance reasoning[88]. Reporting those outcomes as findings about the actor, with the mechanism identified, is the behaviour the whole apparatus is designed to produce.

The campaign also carries its integrity accounting in the same register. Eleven of eleven deterministic checks passed on 2026-09-19, covering exact divisions, receipt presence, sealed-artifact hash equality and lost-source absence probes; the recorded spend was $5.6301 against the $50 additional cap counted at that date, with later budget-mode records moving to an unlimited mode whose per-run resource and timeout bounds remain in force[89]. A judgment layer was added for classification, faithfulness and value-ordering questions and then partly invalidated by an external service outage: both blocker-classification batteries returned degenerate distributions after the incident, their receipts are retained as incident evidence, and the per-lane blocker column therefore rests on document evidence and not on model judgment[90]. A design rule was banked from the healthy window and is followed in this thesis: arithmetic and string lookups stay in code, and model judgment is reserved for classification, faithfulness and value ordering, because a parity probe was answered incorrectly even while the service was healthy[91].

What the campaign does not claim follows directly from the design. It does not claim improvement over any baseline anywhere, because no lane carries a test establishing it: the campaign's role in this thesis is to fix the evaluation protocol, the receipt discipline and the disposition taxonomy that any future improvement claim would have to satisfy. It does not claim that the fourteen suites constitute a portfolio result. It does not pool original-contract with replacement-scope figures, and it does not average across suites. Its one trained actor means it says nothing about model comparisons. And where the record is incomplete — a lost execution source, an unanswered external access request, a run whose budget exceeded the authorised envelope — the incompleteness is written into the ledger rather than

[87]Source: outputs/E1_E14_FINAL_RESULTS_2026-09-19.md.
[88]Source: outputs/E1_E14_FINAL_RESULTS_2026-09-19.md.
[89]Source: outputs/E1_E14_FINAL_RESULTS_2026-09-19.md.
[90]Source: outputs/E1_E14_FINAL_RESULTS_2026-09-19.md.
[91]Source: outputs/E1_E14_FINAL_RESULTS_2026-09-19.md.

resolved by inference.

Chapter 5

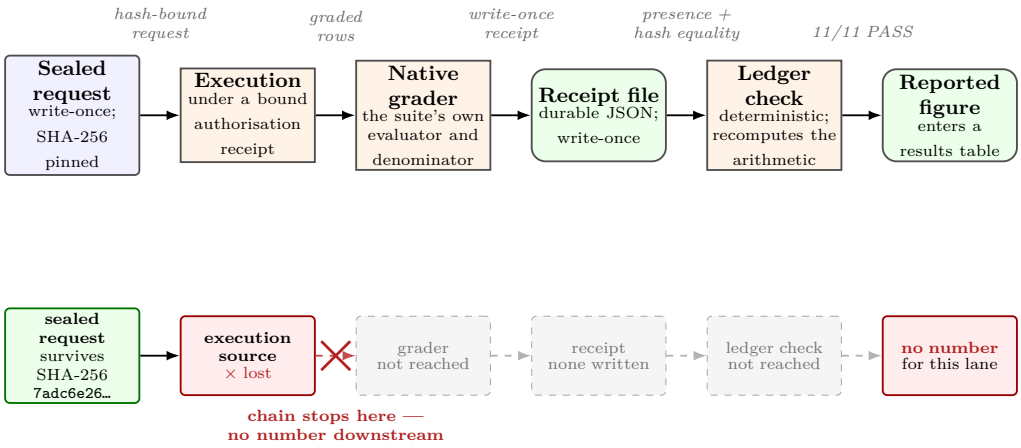
Implementation Details

This chapter describes what was built. Its organising claim is that the engineering contribution of the project is a **dispatch and evidence layer** rather than a training algorithm: the algorithmic content of GRPO is inherited from the literature, but the work of making six reinforcement-learning frameworks and six compute backends addressable through one Tinker-style interface — and of making every number that comes out of them traceable to a sealed, hash-pinned artefact — is where the implementation effort sits. The chapter proceeds from the abstraction (the adapter architecture), through the concrete training loop, to the two engineering episodes that most shaped the codebase (the parallel logging defect and the Colab white-box path), and finally to the evaluation harness and the recovery re-implementations undertaken after an execution-source loss in the final phase of the project (Figure 5.1).

5.1 Two Rosters, One Dispatch Surface

A reader approaching the project's comparison material will encounter two different framework rosters, and conflating them is the single most likely misreading of the implementation. The paper's methods note states the distinction explicitly: the cross-launcher LLM-RL scaffold referenced in the framework-gap discussion is a seven-library roster — Tinker, TRL, SkyRL, veRL, OpenRLHF, Atropos, and HF reference launchers — that targets LLM post-training launchers specifically, while the cross-RL-library evidence reported elsewhere uses a different roster. Critically for this chapter, the same note records that "only Tinker and TRL produced completed runs in this release; veRL and OpenRLHF entries in the `framework_comparison.json` artefact are dry-run placeholders"[1]. The veRL and

[1]Source: platform_hybrid/paper/sections/_shared_methods.tex.



Example lane: the E1 wave-10 recovery (16 tasks, $16) was sealed and locally validated, then the `.codex-run/` deletion removed its execution source — so no number is reported for that lane.

Fail-closed. A break at any link means the lane reports **no number**: the chain is never repaired by inference, and the gap is carried in the ledger of record with a named reopen condition rather than filled in. The break above is itself receipted — the deterministic check `codex_run_execution_sources_absent` records the four lost paths as duly absent (probed 4, still present: none) — and the lane stays at `REBUILD_READY_LAUNCH_PENDING` with its values in no results table. The same rule runs in the opposite direction at E9 MLE-bench, which reports coverage of 40/75 (53.33%) and keeps the suite score null, because coverage without complete grading is not a score.

checks: `LEDGER_CODE_CHECK_2026-09-19.json` (11 of 11 PASS, 0 failed, schema `ledger-code-check-v1`)

Figure 5.1: The evidence chain from sealed request through execution and native grading to a deterministic ledger check. A break anywhere in the chain means no number is reported for that lane.

OpenRLHF entries in that artefact carry peak and last-ten-average reward values, but they are the output of the harness resolving a launch plan, not the output of a training run, and they must never be pooled with real results or read as framework comparison evidence.

Atropos occupies a different position again. Rather than being a cell in the dispatch matrix, it is integrated through a dedicated package under `platform_tinker/atropos/`, a `tinker_atropos` module containing `config.py`, `api.py`, `trainer.py`, `types.py`, `dataset.py`, `stats_utils.py`, and an `environments/` directory with per-task environment servers[2]. Its integration model is the Atropos managed-server pattern: a three-terminal workflow in which `run-api` starts the Atropos API, `launch_training.py --config configs/default.yaml` starts Tinker training against a `TINKER_API_KEY`, and an environment server — a built-in one or any external Atropos environment pointed at the same config file — serves rollouts. The README's quickstart example runs a fifty-step training run on Llama-3.1-8B-Instruct against a GSM8K environment, and the environment contract is that the `env` and `openai` sections are consumed by the environment server while the `tinker` section is consumed by the trainer and ignored by the environment[3]. This separation is what allows an arbitrary existing Atropos environment to be reused against the Tinker backend without modification to the environment.

5.2 The Adapter Architecture: What versus Where

The dispatch surface is built on one orthogonal decomposition, stated in the backends module docstring: "A `Backend` knows **where** to run one experiment cell (local GPU, Modal, Colab, vast.ai, GCP, HF Spaces); the framework dimension (trl/tinker/verl/openrlhf/skyrl) is the **what**. `plan()` resolves a cell into a concrete `LaunchPlan` without spending compute, so `--dry-run` and the matrix test work uniformly"[4]. The `LaunchPlan` dataclass carries `backend`, `framework`, `command`, `driver_file`, `output`, `env`, and `notes`, with a `format()` method for printing; the default `_execute` runs `subprocess.run(plan.command, shell=True, check=True)`. Because plan resolution is separated from execution, the entire matrix can be validated without a GPU, an API key, or a dollar of spend — a property that the dry-run placeholders discussed above depend on, and which is also the

[2]Source: platform_tinker/atropos/.
[3]Source: platform_tinker/atropos/README.md.
[4]Source: platform_local/unified/backends/base.py.

reason those placeholders must be labelled as such (Figure 5.2).

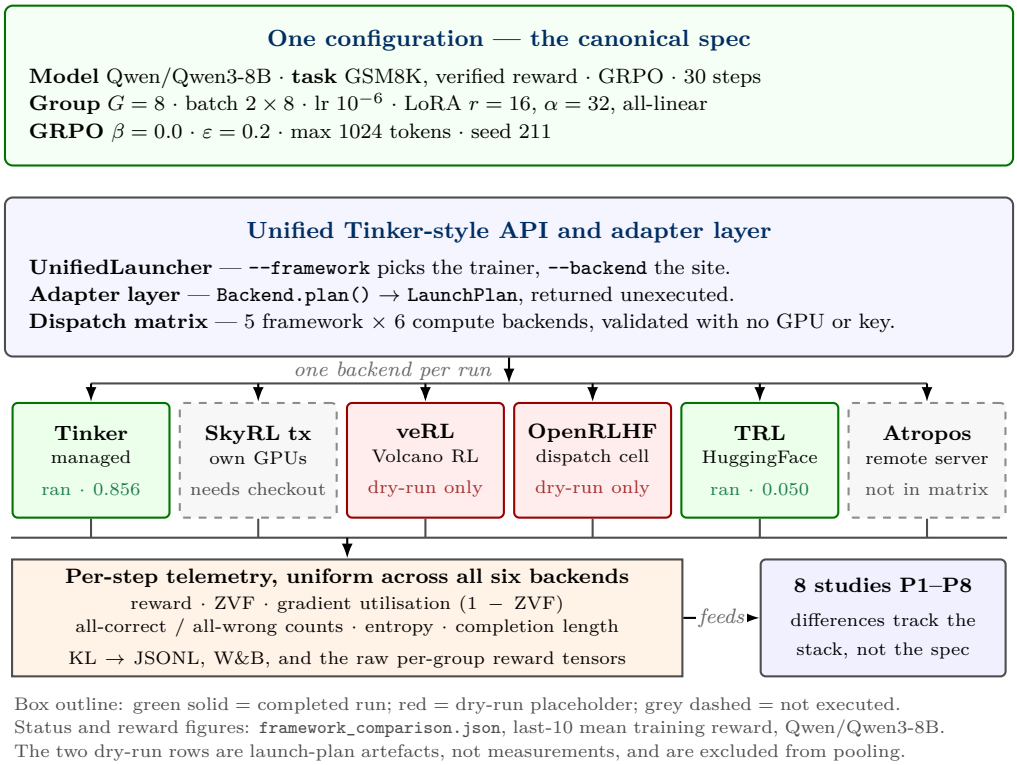


Figure 5.2: Six reinforcement-learning backends behind one Tinker-style API. A single task, reward and decoding configuration drives any backend, so differences in outcome are attributable to the stack.

The matrix itself ships as data. `platform_local/unified/matrix.json` declares schema `tinkerrl-framework-backend-matrix-v1` with thirty cells — five frameworks (trl, tinker, verl, openrlhf, skyrl) crossed with six backends (local, modal, colab, vast, gcp, hfspaces), five cells per row and per column — and a `canonical_experiment` block that pins the single configuration every cell is meant to instantiate: `Qwen/Qwen3-8B` on GSM8K with GRPO, thirty training steps, `num_generations` 8, per-device batch size 2, gradient accumulation 8, learning rate `1e-06`, LoRA rank 16 and alpha 32 over all linear modules, `beta` 0.0, `epsilon` 0.2, maximum completion length 1024, and seed 211[5]. Each cell is a concrete plan; a representative one reads `{"framework":"trl","backend":"local","command":"python -m platform_local.unified --backend local --framework trl --model Qwen/Qwen3-8B","driver_file":"platform_local/trl_integrations/trainer.py","output":"./checkpoints/","env":[],"notes":"in-process framework dispatch"}`.

The launcher implements the dispatch. `UnifiedLauncher` holds three registries: `FRAMEWORKS` mapping `skyrl`, `tinker`, `verl`, `openrlhf`, and `trl` to human-readable

[5]Source: platform_local/unified/matrix.json.

labels; BACKENDS mapping local, modal, colab, vast, gcp, and hfspaces to labels; and ALGORITHMS covering grpo, ppo, reinforce, dapo, and dpo[6]. Dispatch is outer-on-backend and inner-on-framework, which is why the per-framework handlers can each be written against the framework's own idioms. _run_tinker shells to python -m platform_tinker.tinkerrl.grpo_cli --preset <spec.task>; _run_openrlhf delegates to platform_modal/openrlhf/trainer.py:run_openrlhf_training({}, output_dir="./checkpoints/openrlhf"), which drives openrlhf.cli.train_ppo_ray --advantage_estimator group_norm; _run_trl generates a resumable GRPO training script through trl_integrations.trainer.generate_trl_train_script into a temporary directory and executes it; _run_verl imports verl.config and verl.trainer directly; and _run_skyrl requires a SKYRL_CHECKOUT environment variable and emits uv run --extra gpu --extra tinker -m skyrl.tinker.api[7]. For the local and colab backends the launcher runs dispatch_framework() in-process, a path the code comments identify as the one that breaks the Colab entry's self-recursion — an implementation detail worth recording because it indicates that the backend and framework dimensions are not perfectly orthogonal in practice, and the launcher carries framework-specific knowledge precisely at the points where they intersect.

Fairness across frameworks is addressed by configuration auditing rather than by assertion. The platform_hybrid/experiments/framework_config_dumps/ supplement ships one exact hyperparameter dump per framework for the canonical Qwen3-8B + GSM8K matched-configuration run, and the accompanying README recomputes the field partition from those dumps: of 44 tracked leaf fields, 25 are identical across all four frameworks (LoRA rank, alpha, target modules and dropout; GRPO group size, rollouts per group and max new tokens; sampling temperature, top-p, top-k and do-sample; optimizer name, learning rate, weight decay, betas and epsilon; reward type; and the task, seed and source-git-sha metadata), while 11 are Tinker-managed — serialised as null # managed_by_tinker in the Tinker dump — covering kl.beta, kl.surrogate, importance-sampling granularity, reference-model offload and recompute, tokens per optimizer step, reward broadcast precision, gradient checkpointing, micro-batch size, gradient accumulation, and tensor-parallel size[8]. The same README records two disclaimers that matter more than the identical count: model and model_sha256 are *not* identical, because the Tinker run uses Qwen/Qwen3-8B-Base while TRL, veRL and OpenRLHF use the instruction-tuned Qwen/Qwen3-8B, and model_sha256 is

[6]Source: platform_local/unified/launcher.py.
[7]Source: platform_local/unified/launcher.py.
[8]Source: platform_hybrid/experiments/framework_config_dumps/README.md.

a placeholder rather than a verified checksum. Remaining fields are framework-specific and documented rather than harmonised — KL beta is 0.04 for TRL and veRL against 0.02 for OpenRLHF, importance-sampling granularity is token-level for TRL and veRL against sequence-level for OpenRLHF, and reference-model placement differs across all four (`on_device`, `managed`, `separate_process`, and `sharded_ray`). The appendix's original "byte-identical" framing is retracted on the strength of this partition; the honest statement is that a majority of tracked fields are matched, a known Base-versus-Instruct confound survives, and eleven fields are not under user control at the Tinker API level at all.

5.3 The Tinker GRPO Loop

The reference implementation of the training loop lives in `platform_tinker/tinkerrl/grpo.py` and is exposed through `grpo_cli.py`. The loop follows the standard four-stage shape recorded in the Phase-1 report under "The Tinker GRPO Loop": sample a group of `G` completions per prompt, compute group-relative advantages, form the loss, and take an optimiser step[9]. Advantages are normalised within the group to mean zero and unit standard deviation, and the loss is the clipped surrogate with a KL penalty to a frozen reference policy,

$$L(\theta) = -E[\min(\rho\ A,\ \mathrm{clip}(\rho,\ 1-\varepsilon,\ 1+\varepsilon)\ A)] + \beta\ \mathrm{KL}(\pi_\theta \parallel \pi_\mathrm{ref})$$

where ρ is the importance ratio and ε the clipping range. The implementation uses the Tinker SDK's `ServiceClient` with `create_lora_training_client`, `forward_backward_custom` with `loss_type_input="logprobs"` so that per-token log-probabilities flow into a custom loss, `optim_step` with `AdamParams`, and `sample()` for rollouts; resumption uses `create_training_client_from_state_with_optimizer`[10].

The single most consequential property of this loop for the project's research questions is the degenerate case. When the group's reward standard deviation σ`_g` is zero, every advantage `A_i` is zero, the surrogate term vanishes, and only the KL term survives — so the step is a pure pull toward the reference policy with no learning signal at all. This is the mechanism behind the zero-variance-fraction diagnostic that runs through the project; it is recorded in the Phase-1 report as the observation that motivated the collapse analysis, and it is the reason `zero_loss` and `zero_reward` are first-class outputs of the training CLI rather than derived quantities. Zero reward is counted as an average reward equal to zero; zero loss is

[9]Source: platform_tinker/reports/esa_phase1/Phase1_Project_Report_ZVF.tex.
[10]Source: platform_tinker/tinkerrl/grpo_cli.py.

counted as a step loss below `1e-6`[11].

The CLI exposes eight presets — `tooluse_synth`, `tooluse_baseline`, `tooluse_heldout`, `tooluse_xlam`, `gsm8k`, `math100`, `pavlov_xlam`, and `pavlov_portfolio` — each resolving a dataset factory and a reward function through a `DATASET_FACTORIES` registry and a `REWARD_MAP` (`ExactMathReward` for GSM8K, `StrictToolCallReward` for `pavlov_xlam`, `PavlovNonXLAMReward` for `pavlov_portfolio`). Tracking and campaign gates are enforced before a run starts, through `config.validate_tracking()` and `config.validate_campaign_gate()`, and the CLI refuses to launch without a `TINKER_API_KEY`. Each seed's result is printed as a structured record containing `run_id`, `sampler`, `avg_first5`, `avg_last10`, `peak_reward`, `zero_loss`, `zero_reward`, `heldout_reward`, and `reward_trace`, so that a run's evidence is emitted at the point of execution rather than reconstructed afterwards. The eight presets replaced an earlier collection of standalone `grpo_*.py` scripts, with `legacy_main` retained to translate the old flag surface[12].

5.4 The Parallel Runner and a Real Logging Defect

Sweep execution is handled by `parallel_sweep.py` and `campaign.py`, with the runner's design shaped directly by a defect found during development. The defect is documented in the group-size sweep findings and is worth quoting in full, because it is the clearest illustration in the project of why logging infrastructure deserves the same scrutiny as training code:

> The threaded runner trains all G in parallel correctly, BUT `wandb` uses a thread-unsafe global `run` — with 4 threads calling `wandb.init(reinit=True)`, the G8/G16 **summary** writes failed ("Run is finished") after another thread finished its run. Per-step `wandb.log` mostly survived; the summary dict didn't. **Fix for next parallel batch:** isolate each run in its own PROCESS (multiprocessing / subprocess per config), not a thread — or use `wandb-core` service mode with explicit per-run objects. Training itself was unaffected (all step data is in the run log). [13]

[11] Source: platform_tinker/tinkerrl/grpo_cli.py.
[12] Source: platform_tinker/tinkerrl/grpo_cli.py.
[13] Source: platform_hybrid/experiments/results/p3_groupsize/FINDINGS.md.

Two aspects of this record are instructive. First, the failure was silent with respect to training: gradients were correct, steps completed, and per-step metrics were written. Only the run summary — the aggregated dictionary from which a results table would be built — was lost, which is precisely the class of failure that corrupts downstream reporting while leaving the primary artefact intact. Second, the remedy adopted was structural rather than retry-based: isolating each run in its own process removes the shared mutable global entirely, whereas retrying the summary write would have left the race in place. The finding also notes the alternative of `wandb-core` service mode with explicit per-run objects, which would preserve in-process execution at the cost of depending on service-mode semantics that the project chose not to rely on.

The same findings file documents the group-size sweep that exposed the defect, and its numbers are reported here with the qualifications the source attaches to them. The sweep ran `G` $\in$ `{2, 4, 8, 16}` on Qwen3.5-4B for six steps on GSM8K at seed 0 under the W&B group `p3-groupsize-sweep`, with total token counts of 9,663, 19,964, 34,084, and 68,055 respectively. At `G = 2` the run showed a 50% batch-level collapse with a `+0.000` gain; at `G = 4` a 17% collapse with a `+0.125` gain. The source flags the `G = 4` gain explicitly as supported by one example at n = 8 on a single seed and therefore *suggestive rather than proven*, and records that the independent verification of the sweep returned the verdict SUSPECT[14]. These are not results this thesis claims; they are the observations that motivated the group-size investigation and that the later controlled work supersedes, and they are reported in the register in which they were recorded.

5.5 The Colab White-Box Path (Per-Layer Gradient Profile)

The per-layer gradient profile — numbered P1 in the Phase-1 series, and distinct from the P1 cross-scale study of Chapter 6 — asks a question the managed Tinker backend cannot answer. Per-layer gradient norms require access to the model's internals during the backward pass, and the managed service exposes no such interface. The implementation therefore leaves the managed path entirely and runs on a Colab L4 GPU in `platform_hybrid/experiments/openings/p1_layer_profile.py`, using Qwen2.5-1.5B-Instruct with LoRA adapters on the q/k/v/o projections, five GRPO-style advantage-weighted steps, and per-layer

[14]Source: platform_hybrid/experiments/results/p3_groupsize/FINDINGS.md.

LoRA gradient norms recorded at each step[15]. The white-box path is thus a deliberate architectural exception: the project accepts the loss of managed-scale training in exchange for gradient visibility, and confines the exception to a question that only gradient visibility can address.

The first-pass result appeared strong. Across 28 layers, the top-k layer set identified at step 1 overlapped the final top-k set perfectly, with a measured overlap of 1.0; the top 25% of layers accounted for a 0.476 share of gradient mass; layer 0 carried a gradient norm of 63 against 6–17 elsewhere, and a secondary mid-late band appeared at layers 17–23. A scaled re-test disciplined this result. The re-run used `p1_layer_profile_scaled.py` on Qwen2.5-3B-Instruct, with 24 real GSM8K problems, ten steps, `G = 4`, 36 layers, and seeds {0, 1}. The step-1-to-final top-k overlap fell to 0.111 — chance level — and the top-25% concentration measured 0.39 with a standard deviation of 0.013. The source's verdict is unambiguous: the original predictability finding "was an artifact of the toy 1.5B/hardcoded-arithmetic setup"[16]. A later four-seed pass at the same configuration (seeds {0, 1, 2, 3}) supersedes the two-seed figure and gives a mean overlap of 0.083 ± 0.048, again approximately chance, with concentration 0.39[17].

A companion freeze test asked whether the layer profile could be exploited: if a small band of layers dominates the gradient, freezing the rest should preserve the benefit at lower cost. It did not. The full 28-layer configuration trained 2.18M parameters and produced a mean held-out gain of `+0.0625` (seed 0 `+0.125`, seed 1 0.0). Freezing to a hot band comprising layer 0 plus layers 12–27 — 17 layers, 1.32M parameters, or 0.61 times the full count — produced `+0.000`. The source concludes that "aggressive freezing loses the benefit… The freeze lever does not work at this scale", and records that the comparison is underpowered at n = 8 held-out examples over 2 seeds[18]. Both P1 outcomes are negatives: the apparent structure did not survive scaling, and the intervention it suggested did not survive testing. They are reported as findings rather than suppressed, because a layer-importance claim that collapses under a scaled re-test is exactly the kind of result the project's evidence discipline exists to surface.

[15]Source: platform_hybrid/experiments/results/p1_layerfreeze/FINDINGS.md; platform_tinker/reports/esa_phase1/Phase1_Project_Report_ZVF.tex.

[16]Source: platform_hybrid/experiments/results/p1_layerfreeze/FINDINGS.md.

[17]Source: platform_hybrid/experiments/results/p1_layerfreeze/scaled_4seed_result.json.

[18]Source: platform_hybrid/experiments/results/p1_layerfreeze/FREEZE_FINDINGS.md.

5.6 Software Modules

The module inventory recorded in the Phase-1 report remains the accurate map of the codebase. The unified launcher, `platform_local/unified/launcher.py`, maps one configuration onto a chosen backend; `platform_tinker/tinkerrl/grpo.py` is the reference GRPO loop against the managed backend; `platform_local/trl_integrations/` and the `verl/` integration hold the per-framework adapters, with `platform_modal/openrlhf/` and `platform_hybrid/skyrl/` completing the roster, and `platform_tinker/atropos/` carrying the Atropos integration described above. Analysis code is separated from training code by intent: `p2_collapse_analysis.py` recomputes zero-variance fraction and collapse statistics from stored reward tensors, so the diagnostic is reproducible from evidence rather than from a live run; `p1_layer_profile.py` implements the white-box gradient probe; `parallel_sweep.py` and `campaign.py` drive sweeps; and `p8_detector.py` is an exploratory method-trace classifier evaluated by row-wise cross-validated AUROC[19].

The p8 detector deserves an explicit caveat in an implementation chapter. It is an exploratory artefact evaluated by row-wise cross-validation, and its AUROC is a measure of separability on the traces it was given, not a benchmark result against an external suite; it should not be read alongside the E-lane scores reported later in this thesis. Beyond these modules, the `platform_local/unified/` package carries `canonical.py` for the canonical configuration and `peft_utils.py` for adapter handling, and the backends package implements one module per backend (`local.py`, `modal.py`, `colab.py`, `vast.py`, `gcp.py`, `hfspaces.py`) behind the common `base.py` interface[20].

5.7 The E1–E14 Evaluation Harness

Evaluation is governed by a completion gate, `zvf-program/flagship/e1_e14_completion_gate.py`, schema `e1-e14-completion-gate-v1`. The gate maintains five evidence classes — `COMPLETE_EXACT`, `PARTIAL_EXACT`, `PARTIAL_RECOVERY`, `BLOCKED_EXTERNAL`, and `LOCAL_SETUP_READY_PROVIDER_INPUT_REQUIRED` — and maps lane source statuses onto them, with unknown states failing closed rather than defaulting to a classification. Its `LANE_SPECS` table binds each lane to a replacement scope, a task count, and the runner or adapter that produces its evidence: E2 FrontierSWE at 17 tasks (`pavlov_frontier_swe_eval_adapter.`

[19]Source: platform_tinker/reports/esa_phase1/Phase1_Project_Report_ZVF.tex.
[20]Source: platform_local/unified/.

py, frontier_swe_eval.py), E3 SDAB at 80, E4 BankerToolBench at 100 (run_harbor_via_tinker_bridge.py), E5 APEX-Agents at 480, E7 BinaryAudit at 46 (modal_non_e11_readiness.py), E9 MLE-bench at 75 (e9_mle_bench_streaming.py), E12 AppBench at 6, and lanes E6, E8, E10, E13 and E14 with their respective runners. The gate's own header states its scope precisely: it neither invokes an adapter nor accepts a package as a benchmark result[21] (Figure 5.3).

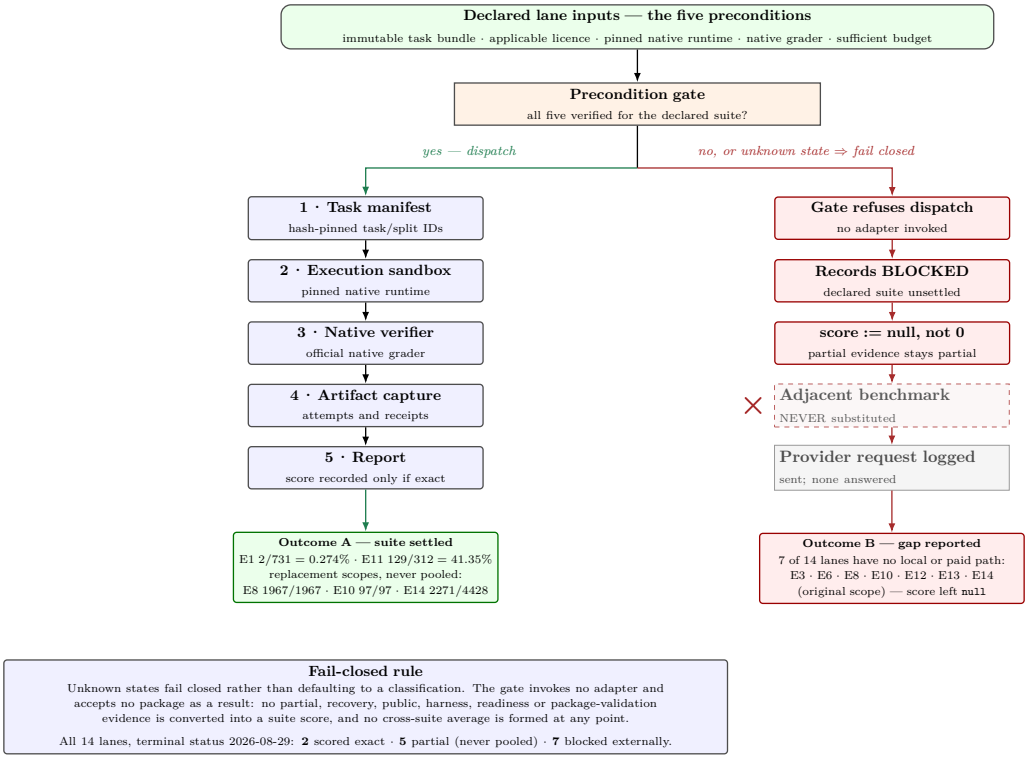


Figure 5.3: Evaluation harness internals. The harness is fail-closed: when a precondition cannot be satisfied the lane records BLOCKED rather than substituting an adjacent benchmark.

The results the gate classifies are recorded in outputs/E1_E14_FINAL_RESULTS_2026-09-19.md, verified by eleven of eleven deterministic checks on 2026-09-19 and bound to surviving receipts. The single Tinker-trained actor is base Qwen/Qwen3.6-35B-A3B at commit 995ad96e… with adapter arvindcr4/pavlov-portfolio-qwen36-seed809-stepfinal-tinker-cf0ad8c1… at commit 64444133…, in bfloat16[22]. The document's opening constraint governs how everything below may be read: original-contract and replacement-scope numbers are never pooled, and per the project evidence rule no cross-suite average is formed at any point.

Three replacement scopes are strictly complete. E8 covers the LAB-Bench public

[21]Source: zvf-program/flagship/e1_e14_completion_gate.py.
[22]Source: outputs/E1_E14_FINAL_RESULTS_2026-09-19.md.

split at 1967/1967 across eight categories, with per-category accuracy of TableQA 0.787, FigQA 0.514, ProtocolQA 0.250, SuppQA 0.183, LitQA2 0.176, DbQA 0.131, SeqQA 0.033, and CloningScenarios 0.000. E10 covers AgentDojo's benign utility scope at 97/97. E11 covers VerilogEval at 312/312 in two native framings, yielding 129/312 = 41.35% pass@1 canonically, composed of 67/156 on code completion and 62/156 on spec-to-RTL, with a 129/311 sensitivity figure retained non-canonically. E14's Omni-MATH lane reached 4426/4428 accepted (99.95% parser coverage) with the official accuracy reproduced at 2271/4428 = 0.5131043831902395; the two remaining rows were resolved as judge outputs truncated before any equivalence judgement, with no score impact because both rows already count in the denominator as not-correct.

Within the original contracts, E1 returned 2/731 = 0.274% pass@1 over 731 terminal generations and 713 native evaluations, with 14 generation failures and 4 artifact losses retained in the denominator; E2 returned 1/17 tasks with a replay-normalised value of 0.8628; E4 returned 1/100 with a recovery metric of 0.3115; E5 returned 7/480 native-scored with a prefix mean of 0.050505; E7 attempted 1/46 with a verifier reward of 0.0; and E9 graded 40/75 competitions (53.33% coverage) with the suite score left null, its merged-vLLM arm reported separately at one valid grade and an H&M value of 0.02132[23]. Counted spend against the additional cap was $5.6301 of $50, with the E5 component of $80 separately recorded as unreserved.

One further result from this phase belongs in the implementation record because of what it demonstrates about the harness. Under separate authorization, the E4 BankerToolBench suite was re-run from scratch on the base model through the Modal bridge's `/v1/responses` path at `--n-concurrent 2` under a cap of $55.91445263. All 100 of 100 trials completed, with one `NonZeroAgentExitCodeError`, and the mean reward was 0.0 — every trial in the zero bucket. The audited mechanism is that agents terminated after 2–14 steps (median about 6) without producing output in `deliverables/`, causing the verifier to short-circuit to zero on a "No deliverables found" condition, and that 27 or more trajectories ended in role-token degeneration of the form `user user assistant …` repeated. The source's own reading is that the 3.6B base model cannot sustain the opencode tool-use loop, so this zero measures tool-dialogue collapse rather than finance reasoning; the run cost $15.07 of the cap over 3h29m, against 16.35M prompt and 2.18M completion tokens[24]. Reporting a mean reward of 0.0 with its mechanism attached, rather than reporting a low score and moving

[23]Source: outputs/E1_E14_FINAL_RESULTS_2026-09-19.md.
[24]Source: outputs/E1_E14_FINAL_RESULTS_2026-09-19.md.

on, is the standard this chapter's evidence rules are meant to enforce.

5.8 The Bridge and Sampler Path

Runners reach a Tinker-trained model through `zvf-program/flagship/modal_tinker_openai_bridge.py`, a Modal-deployed OpenAI-compatible bridge whose own header states that "the deployed bridge is infrastructure, not a benchmark result. Its default paid-inference budget is zero"[25]. The bridge declares app name `pavlov-tinker-openai-bridge`, model alias `pavlov-qwen36-tinker` over model id `Qwen/Qwen3.6-35B-A3B`, and pins its sampler at `tinker://cf0ad8c1-1f1b-5ff3-8bd7-2a0bf232657b:train:0/sampler_weights/seed809_final`, with the Hugging Face side pinned by commit `64444133c55d88c3f1bf0df8a2f5d7ac646125c8`. Runtime limits are 16,384 maximum completion tokens against a 65,536-token model context, with an 8,192-token context prefix reserved; `DEFAULT_MAX_USD` is `Decimal("0.00")`, so every paid path must be opted into explicitly at launch. The pinned dependency set is `fastapi` 0.116.1, `huggingface-hub` 1.27.0, `tinker` 0.30.0, `transformers` 5.5.4, and `wandb` 0.21.0[26].

The architectural point of the bridge is that it converts a Tinker sampler into an OpenAI-compatible endpoint. A harness written against the OpenAI chat-completions API — which most agentic benchmark harnesses are — can therefore be pointed at a Tinker-trained actor without modification, and the E4 rerun described above used exactly this path in its `/v1/responses` form. The same pattern appears in the actor-side configuration of the E1 recovery wave, where the actor is addressed as an OpenAI-compatible chat-completions service at `http://127.0.0.1:18015/v1` (container port 8000) under model name `pavlov-public-portfolio-bf16`, with `max_tokens` 8192, temperature 0, seed 809, top-p 0.95, and `chat_template_kwargs {"enable_thinking": false}`[27]. The bridge and the local actor endpoint are two deployments of the same interface contract, and keeping that contract stable across deployment modes is what allows one harness to serve both.

[25]Source: zvf-program/flagship/modal_tinker_openai_bridge.py.
[26]Source: zvf-program/flagship/modal_tinker_openai_bridge.py.
[27]Source: zvf-program/e1_wave10/README.md.

5.9 Recovery Re-implementations under zvf-program/

In the final phase of the project an execution-source loss — deletion of the `.codex-run` directory — removed the finish-era recovery and build code, confined in scope to the E1 wave10 controller, the E9 restored dependencies, and the E6 validator, while the replacement-lane native runners survived under `zvf-program/flagship/`[28]. The project's lost-source policy is that sealed requests, validation receipts and test logs pin any faithful re-implementation. Four such re-implementations are worth describing, because together they show what the evidence discipline looks like when it is expressed as code rather than as convention. None of them produced a benchmark number; all four are launch-ready or design-complete artefacts whose value is that their safety properties are testable offline.

E1 wave10. The wave-10 recovery package rebuilds the sealed v6 controller for sixteen tasks — nine `google__gson`, five `hashicorp__terraform`, and two `immutable-js` — selected from `resource_request_v6.json` at SHA256 `7adc6e2616f61bc7ef4d08533422a5df83ed1253f2fea507ce831890b2bb972a`. Four safety properties are enforced in code. Selection is sealed-only: the driver may act only on the hash and the embedded sixteen-task id list. Lane evidence stays behind a boundary, in that requests are built only by `public_swe_multilingual_native.build_actor_request` and receipts use the `collect_attempts` schema. The v6 process-group contract is enforced with unconditional finalisation on both success and failure, with the result JSON read only after `killpg(pgid, 0)` reports ESRCH, and with success requiring four conditions jointly — worker exit zero, direct worker reaped, the original group absent, and the controller's remaining deadline not yet expired. Finally, failures are never fabricated: `GENERATION_FAILED` is reserved for genuinely completed responses that lack a valid unified diff, receipts are write-once and immutable, and there is no automatic retry, resume, or replay. The package carries an eight-item pre-dispatch checklist, and the reservation records `public0919-e1-actor11` at 8 USD over 3600 seconds and `public0919-e1-runtime11` at 8 USD over 10800 seconds with status `RESERVED_NOT_DISPATCHED`[29]. The re-implementation verifies 10 of 10 targeted regressions offline, matching the lost suite's record: `test_process_group.py` at 4/4 and `test_cleanup_regressions.py` at 6/6[30].

[28] Source: outputs/E1_E14_FINAL_RESULTS_2026-09-19.md.
[29] Source: zvf-program/e1_wave10/README.md.
[30] Source: outputs/E1_E14_FINAL_RESULTS_2026-09-19.md.

E2 driver. The CORE-Bench orchestration driver (`e2_core/driver.py`, schema `e2-orchestration-driver-v1`) schedules a target-helper GPU chain on GCP with fixed integer offsets — `TARGET_CUTOFF_OFFSET` 1590, `TARGET_BUDGET_END_OFFSET` 1800, `HELPER_DELETE_OFFSET` 2610, `HELPER_BUDGET_END_OFFSET` 2700, `HELPER_TAIL_SECONDS` 600, `A_LE_H_PLUS` 210, and `DELETE_ONSET_ALLOWANCE` 30 — under an envelope of 3 USD target hold and 1 USD helper hold on an `n1-standard-4` instance with 4 vCPU, 15 GiB memory, an NVIDIA Tesla T4, and a 50 GiB boot disk. The driver defaults to validation-only and refuses `--execute` without a fresh owner IAM binding receipt and a sealed v14-style request, which is fail-closed behaviour at the point where money would otherwise be spent. Twelve of twelve offline tests pass, covering schedule arithmetic, exact early DELETEs, cutoff behaviour, `UNVERIFIED/LATE` reporting, the no-retry property, and the IAM gate[31]. The companion `launch_gate.py` is deliberately trivial — it verifies only that the amendment acceptance record and the surviving CORE-Bench harness entry points exist, exiting zero when they do — which keeps the gate's claim small enough to be true[32].

E5 successor27. The successor-27 controller (schema `e5-successor27-controller-v1`) rebuilds the expired sealed packet for twenty-seven Tau3 banking tasks that were never started, anchored to inventory SHA256 `ad80e45e5da84fbcf988f572e2ad837a7573a2c70381af6874906988dd498368` and old-request SHA256 `a7739b1ea3b0234ade4100ae940da7149cb016759471b5c7df3d938496caf135`, with trial seed 626729, native seed 300, and a total of 97 tasks in scope. The identity model is `Qwen/Qwen3.6-35B-A3B` at revision `995ad96eacd98c81ed38be0c5b274b04031597b0`, served as `pavlov-public-portfolio-bf16`, under fresh identifiers (`public0919-e5-actor11`, `public0919-e5-native11`, run id `E5-native-20260919-11`); the prior `public0912-e5-` namespace is barred from reuse. Concurrency is bounded at three; the host wall is 21600 seconds, the actor function 21540 seconds, native dispatch stops 200 seconds before the actor provider deadline, and the readiness window is 800 seconds[33]. The controller exposes offline `verify`, `plan` and `gate` subcommands, and launch requires a lead-issued bound authorization receipt plus a runtime adapter; the offline halt test passes 52 of 52, and the live `verify` and `plan` paths return `PLAN_READY_NOT_EXECUTED` while `gate` and `launch` correctly refuse[34].

E13 supervisor. The hosted supervisor for the BALROG lane (`e13_balrog/ho`

[31]Source: zvf-program/e2_core/driver.py; outputs/E1_E14_FINAL_RESULTS_2026-09-19.md.

[32]Source: zvf-program/e2_core/launch_gate.py.

[33]Source: zvf-program/e5_successor27/controller.py.

[34]Source: outputs/E1_E14_FINAL_RESULTS_2026-09-19.md.

`sted_supervisor.py`) is a design artefact, written but not deployed. Its central invariant is an absolute deadline D = t0 + 3600 seconds that is never rebased on process start, reconnection, or lease renewal, with a short renewable lease (default 5 seconds) issued through HMAC-authenticated, strictly sequenced renewal records that reject replays, and a 30-second early-cleanup reserve that stops honouring renewals at D − 30. The Modal API call is stubbed behind `--dry-run` and live execution is deliberately unimplemented, guarded by a `NeverExecutedError` raised if live execution is ever attempted. Renewal and receipt HMAC domains are direction-separated (`e13-supervisor-domain/renewal-v1` and `e13-supervisor-domain/receipt-v1`), so a supervisor receipt replayed as a renewal is rejected rather than honoured — a defence against the specific failure mode in which a legitimate signature is reused in the wrong direction. Terminal statuses are `CLEANUP_VERIFIED`, `CLEANUP_UNVERIFIED`, and `CLEANUP_LATE`, and the offline test suite passes 17 of 17[35].

Read together, these four artefacts are the implementation answer to a question the results chapters raise: how a project maintains evidential integrity when its execution environment is partly lost and partly unreachable. The answer taken here is to re-derive the controllers from the sealed requests and the surviving test records, to keep every re-implementation offline-testable, and to make the no-retry, no-replay, hash-pinned, fail-closed properties explicit in code so that they can be demonstrated rather than asserted. The cost of that discipline is visible: four controllers, three of them complete and one a deliberate stub, produced no new benchmark numbers in this phase. That is the intended trade.

[35] Source: zvf-program/e13_balrog/hosted_supervisor.py; outputs/E1_E14_FINAL_RESULTS_2026-09-19.md.

Part III

Results

Chapter 6

Results I: Core GRPO Studies (P1–P4)

This chapter reports the four core GRPO studies that make up the first half of the paper series: the cross-library and cross-scale behaviour of GRPO training reward (P1), the Zero-Variance Fraction as a diagnostic of signal starvation (P2), the effect of group size on trainability (P3), and the length-bias and held-out-generalisation behaviour of GRPO against its Dr. GRPO variant (P4). Each study is reported here in the form the evidence supports, which in every case is more conservative than the hypothesis that motivated it. Three of the four pillars return a negative or an explicitly bounded result, and the fourth returns a diagnostic whose measured association with the outcome it is meant to predict is weaker than the literature would suggest (Figure 6.1).

6.1 Scope, Sources and Evidence Boundary

The four pillars share one methodological commitment, which is the organising principle of the whole project: every comparative claim is made with the entire training stack held fixed except the one factor under test. Where that was not possible, the result is reported as an association rather than a contrast, and the confound is named. The reader should treat this chapter as the primary result surface for P1–P4; the corresponding LaTeX sources are the canonical artefacts, and every figure quoted below is taken from one of them.

Each study draws on a different evidence base, and the differences matter. P1 pools 70+ runs across seven reinforcement-learning libraries and five model families spanning 0.6B to 1T parameters, but the frontier-scale anchors within that pool are single-seed runs executed through a managed training and inference API,

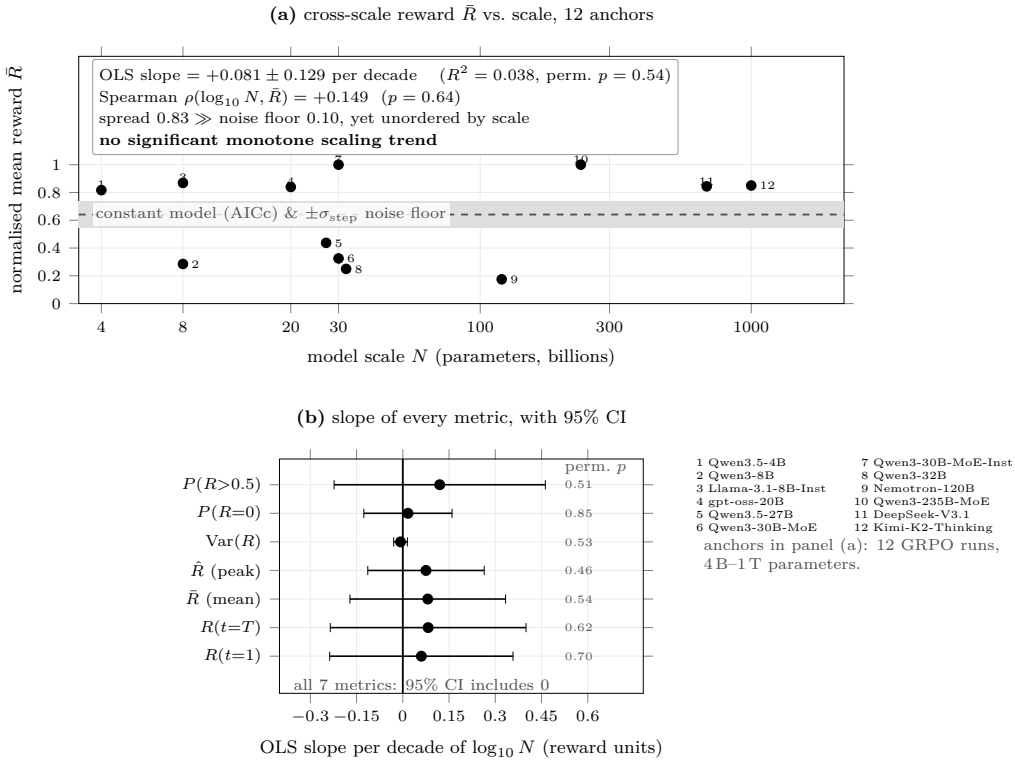


Figure 6.1: Cross-scale behaviour of reward across the studied model range. No reliable monotone scaling trend is recovered; the conservative negative is the result.

so the cross-scale analysis is descriptive and its uncertainty is wide[1]. P2's measured magnitudes come from a controlled five-seed panel plus a matched-budget rollout budget check; a portion of its comparative table is a declared simulation projection rather than a set of independent library deployments, and that distinction is preserved below[2]. P3's measured sweep covers G ∈ {2, 4, 8, 16}; every G = 32 and above number in the pillar is a joint-fit reconstruction, not a trained arm[3]. P4's controlled comparison is sixteen runs across two algorithms and two tasks; its roster-level audit is drawn from a separate, larger pool that mixes models, tasks and libraries[4].

Throughout this chapter, "statistically indistinguishable" means that a formal equivalence or difference test failed to separate the arms at the stated level, not that the arms were demonstrated to be equivalent. Where an equivalence test was run and returned a null, that is stated explicitly.

[1]Source: platform_hybrid/paper/sections/p1_abstract.tex.
[2]Source: platform_hybrid/paper/sections/zvf.tex.
[3]Source: platform_hybrid/paper/sections/p3_intro.tex.
[4]Source: platform_hybrid/paper/sections/length_bias.tex.

6.2 P1 — Cross-Library and Cross-Scale GRPO Behaviour

The question P1 sets out to answer is whether the benefit of GRPO post-training scales with model size. The framing is deliberate: the hypothesis of interest is a Chinchilla-style power law in parameter count N, and the paper's declared claim boundary is that it reports a descriptive limits and identifiability audit rather than a positive scaling law[5]. The short answer is that no reliable scaling law is recovered from the reward curves, and that the evidence base is too small and too heterogeneous to falsify a plausible scaling hypothesis even if one were present.

6.2.1 The anchor pool and the fitted saturation forms

The fitted anchor pool spans roughly 2.4 orders of magnitude in parameter count. This is a subset of the full 0.6B–1T roster: for the cross-scale fits, only anchors with a usable per-step reward trace and a known parameter count were retained. Two properties of that pool limit what the fits can support, and both are stated here rather than left to the reader. First, the anchors are not measured on a common footing: five of the twelve are interrupted runs whose recorded values are means over three to five logged steps, against twenty to thirty for the remaining seven, so the dependent variable is a short-horizon snapshot for part of the pool. Second, scale is collinear with training recipe across these anchors rather than independent of it: the learning rate, LoRA rank and related settings change together and only at the scale boundary, so a fit through the pool cannot separate an absent scale effect from recipe changes offsetting scale changes. The null reported below is therefore a statement about this pool as configured, and not an estimate of how reward scales with parameter count in general. The frontier-scale anchors — Qwen3-235B-MoE, DeepSeek-V3.1 at the 685B class, Kimi-K2 at 1T, and Nemotron-120B — were obtained through the Tinker managed training and inference API in a single-seed, descriptive-only regime, since full base weights are not released for those anchors and the project does not own frontier hardware[6].

Each trace was fitted with a saturating-exponential form R(t) = R_max(1 − e^{−λt}), with the saturation rate λ constrained by the optimiser to the interval [0, 10]. The fitted results are the first place the negative result becomes visible. Across the five well-characterised frontier-class anchors, the ordinary-least-squares trend slope on the raw reward trace, β_OLS, satisfies $|\beta_OLS| \leq 0.0037$ in every

[5]Source: platform_hybrid/paper/sections/p1_abstract.tex.
[6]Source: platform_hybrid/paper/sections/p1_abstract.tex.

case — that is, the traces are flat to within a few thousandths of a reward unit per step[7]. Qwen3.5-4B fits a mean reward of 0.817 with a peak of 1.000 and λ pinned at the bound of 10.0; Qwen3-8B fits 0.285 with a peak of 0.625; Llama-3.1-8B-Instruct fits 0.869 with a peak of 1.000; DeepSeek-V3.1 fits 0.844 with a peak of 1.000. Four of these five traces have λ pinned at the fit boundary — that is, the optimiser wanted a faster saturation than the constraint permitted — and the paper's verdict is that for those traces no single saturation exponent is identifiable at all[8].

The exception is Nemotron-120B, which fits R_max = 0.182 with $\lambda = 0.99$, well inside the bound, and which the pipeline classifies as a collapse rather than a saturation. The autopsy table records a peak at step 3, a fitted R_max of 0.875, a late-phase mean R_late of 0.208, a drop of −0.667 from peak to late phase, a zero-reward step fraction of 0.55, and a post-peak OLS slope of −0.0036 per step[9]. Across the rest of the pool, the zero-reward step fraction is at most 0.067. The Nemotron run is therefore better described as a distinct collapse phase than as a point on a smooth trend, and this is the reason the pillar's headline object is a local, stack-conditioned taxonomy of endpoint ceilings rather than a functional form in N[10].

6.2.2 The cross-scale test and its nulls

Three cross-scale statistics were computed against $\log_{10}$ N: the slope of the mean reward, the slope of the peak reward, and the slope of the reward variance. None is distinguishable from zero. The mean-reward slope is −0.024 per decade with a bootstrap mean of −0.117 and a 95% interval of [−0.796, +0.313], Spearman $\rho = -0.07$; the peak-reward slope is +0.037 with a bootstrap mean of −0.010 and an interval of [−0.623, +0.197], $\rho = +0.22$; the variance slope is −0.002 with an interval of [−0.063, +0.027], $\rho = -0.14$[11]. A residual-resampling bootstrap with 1,000 draws applied to the Nemotron anchor returns R_max = 0.310, $\bar{\lambda} = 3.26$, $P(\lambda \geq 9.5) = 0.277$ and a t_{80} of 17.87 steps with an interval of [0.16, 161.4] — an interval wide enough to contain almost any behavioural claim. A 70/30 anchor holdout returned slope estimates of +0.0000, +0.0000, −0.0000, −0.0000 and +0.0016 across the five holdout configurations[12]. Fitting a power law over the full twelve-anchor set from 4B to 1T returns $p \geq 0.46$ for every exponent tested.

[7]Source: platform_hybrid/paper/sections/scaling_laws.tex.
[8]Source: platform_hybrid/paper/sections/p1_abstract.tex.
[9]Source: platform_hybrid/paper/sections/scaling_laws.tex.
[10]Source: platform_hybrid/paper/sections/p1_abstract.tex.
[11]Source: platform_hybrid/paper/sections/scaling_laws.tex.
[12]Source: platform_hybrid/paper/sections/scaling_laws.tex.

Two model-selection checks were run to see whether the saturation story was at least the best available description of the traces. Fitting constant, linear and saturating forms to each trace, AIC and BIC select the constant model on every single trace, with a ΔAIC of at most 2 over the runner-up[13]. That is, a flat line describes these reward curves at least as well as any saturating curve does. A changepoint test with a permutation null estimated change locations $\hat{\tau}$ of 27, 28, 5, 15 and 18 steps across five traces, with permutation p-values of 0.493, 0.789, 0.742, 0.909 and 0.137; none is significant. An effective-plateau-horizon analysis at tolerance $\varepsilon = 0.20$ found that only two of the five traces — Qwen3.5-4B and Llama-3.1-8B-Instruct — reach a horizon at all, both at 27 steps, because the others never satisfy the tolerance[14].

A compute-adjusted proxy, which folds an estimated token budget into the scale axis, returns a slope of $+482 \pm 490$ per decade. The point estimate is positive and the interval is an order of magnitude wider than the estimate, so the honest reading is that the compute axis does not rescue the scaling story either.

6.2.3 Where the geometric template fails

The three-phase learning template used throughout the project — a slow start, a period of rapid improvement, and a plateau, with collapse as a degenerate tail — was tested directly against the anchor pool as an internal consistency check. Only 2 of the 12 anchors match the template's geometric prediction[15]. The template itself survives as a descriptive vocabulary: it names the failure modes the pipeline detects, and the collapse criteria (peak ≥ 0.4, late mean below 40% of peak, zero-reward fraction ≥ 0.30) are used in the pillar's flagging throughout. What does not survive is the claim that these traces realise the template as a progression.

One further contrast deserves note because it is our own measurement rather than a synthesis. Under a strictly matched same-stack comparison of PPO and GRPO, the paired algorithm difference is -0.002 with $p = 0.374$[16]. This is the cleanest deconfound the project has: the entire training stack held fixed except the estimator, and the two estimators are statistically indistinguishable. It is reported here rather than in P2 or P3 because it is the algorithmic counterpart of the P1 null — the cross-scale story is not the only axis on which GRPO's advantage fails to separate from its alternatives.

[13]Source: platform_hybrid/paper/sections/scaling_laws.tex.
[14]Source: platform_hybrid/paper/sections/scaling_laws.tex.
[15]Source: platform_hybrid/paper/sections/p1_abstract.tex.
[16]Source: platform_hybrid/paper/sections/frontier_synthesis_scaling.tex.

6.2.4 A withdrawn architectural contrast

An earlier version of this analysis reported a MoE-versus-dense gap across six pairs of anchors, with a mean gap of +0.338 and a permutation p = 0.023. That claim has been withdrawn. The defect is architectural: the Nemotron-120B anchor is a LatentMoE, not a dense model, so the original six-versus-six allocation misclassified it. With the classification corrected, the same test returns a one-sided permutation p ≈ 0.178. A leave-Nemotron-out recomputation returns a gap of +0.278 with p ≈ 0.049, and this too is withdrawn, because removing the misclassified anchor from one side of the comparison does not repair the design[17]. The withdrawal is recorded here because the project's convention is that retired claims remain visible with the reason for retirement attached.

A related cross-anchor statistic is retained, with its own caveat. Across the twelve anchors, seven have $|\Delta_1T| \leq 0.2$, where Δ_1T measures how far the anchor's endpoint deviates from the three-phase template's terminal prediction; the largest deviation is Qwen3.5-27B at +0.562 and the largest negative is Nemotron at -0.50[18]. A Cohen's κ analysis of the categorical assignments is also reported in the source. The honest summary is that categorical capability structure carries more signal than $\log_{10}$ N, which is what the paper's abstract says and what the numbers support.

6.3 P2 — The Zero-Variance Fraction

The second pillar asks a mechanistic question: when GRPO stops learning, is the cause visible in the reward statistics of the groups it samples? The diagnostic proposed is the Zero-Variance Fraction (Figure 6.2).

6.3.1 Definition and measurement

For a prompt x sampled from the training distribution, GRPO emits G completions and scores each with a verifier. Centring those rewards within the group produces the advantage signal. If all G completions receive the same reward — all correct, or all incorrect — every centred reward is zero and the group contributes no contrastive signal to that step's update. The Zero-Variance Fraction is the fraction of sampled groups in a step for which this occurs, and the gradient

[17]Source: platform_hybrid/paper/sections/scaling_laws.tex; provenance in autoresearch/deli-neurips-tmlr-260802/state/nemotron_architecture_resolution.md.

[18]Source: platform_hybrid/paper/sections/scaling_laws.tex.

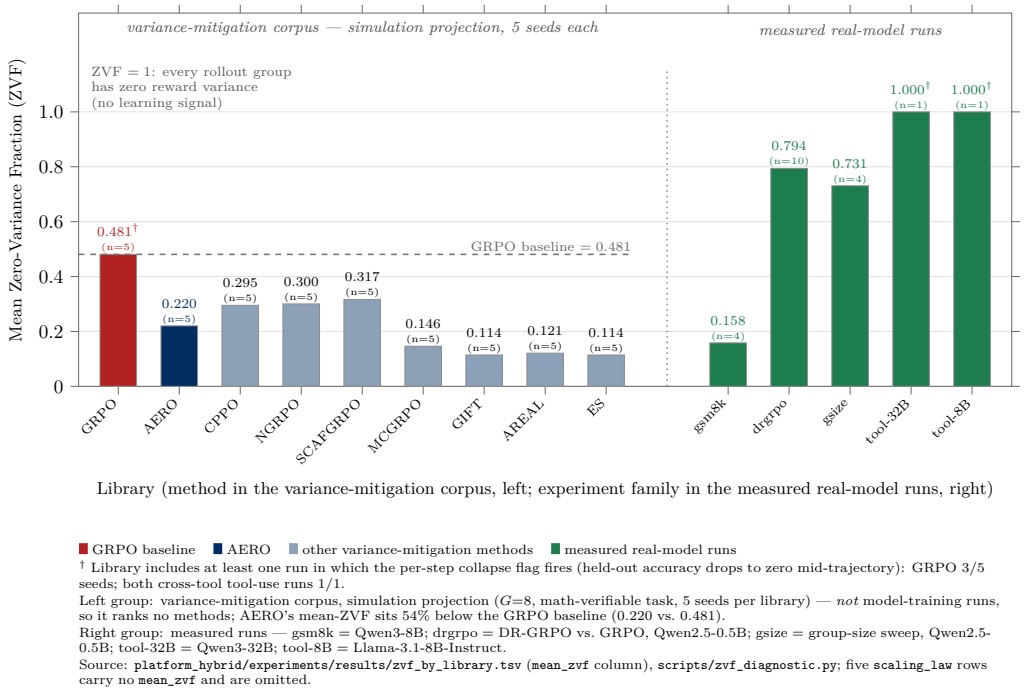


Figure 6.2: Mean Zero-Variance Fraction by backend library. Signal starvation is large and varies across stacks, which is the empirical case for reporting the stack alongside any reward figure.

utilisation is defined as GU = 1 − ZVF[19].

One clarification from the source is load-bearing and is repeated here because it is easy to misread: a zero-variance group does not mean the total gradient is zero. The policy-gradient term can still carry weight through other paths. What ZVF measures is the fraction of groups contributing zero centred reward contrast[20]. The strongest statement the pillar supports is about signal starvation in the group-relative term, not about a vanished update.

The statistic is validated by recomputation. Three stored GSM8K group tensors were re-read from disk and their ZVF recomputed from the raw per-group rewards; the recomputed values are 0.130, 0.190 and 0.155, pooled 0.1583, matching the reported values exactly[21]. This is the mechanical check that the quantity in the tables is the quantity the code computes.

6.3.2 Measured magnitudes across libraries

The headline measurement is a nine-row panel run at K = 8 with five seeds and a 100-step rollout budget, spanning vanilla GRPO and eight named variance-

[19]Source: platform_hybrid/paper/sections/zvf.tex.
[20]Source: platform_hybrid/paper/sections/zvf.tex.
[21]Source: platform_hybrid/paper/sections/p2_abstract.tex.

mitigation methods. Vanilla GRPO records a mean ZVF of 0.481 against a mean last-10 accuracy of 0.379, with zero of five seeds collapsing, zero of five drifting, and five of five plateauing. The remaining rows are: AERO 0.220 ZVF with 0.399 accuracy; CPPO 0.295 with 0.392; NGRPO 0.300 with 0.387; SCAFGRPO 0.317 with 0.409; MCGRPO 0.146 with 0.322; GIFT 0.114 with 0.326; AREAL 0.121 with 0.328; and ES 0.114 with 0.316. The four lowest-ZVF rows each record one of five seeds drifting and four of five plateauing[22].

The variance-mitigation rows require a caveat that the source states explicitly and that this chapter preserves: they are a declared simulation projection used for sensitivity analysis, and they "should be read as synthetic, not as independent real-library deployments"[23]. The vanilla GRPO row and the collapse/drift/plateau counts are the measured content. Anyone citing the nine-row ordering as evidence that a named method mitigates failure would be citing a projection.

Two further measured results sharpen the picture. First, across 60 measured runs spanning the nine variance-mitigation methods and the group-size sweep, the late-phase ZVF of GRPO runs sits in the range 0.87–0.97, while early-phase ZVF for variance-mitigation GRPO and for the largest group size sits in the range 0.04–0.13; smaller group sizes start higher, between 0.25 and 0.61. The lag-1 autocorrelation of the ZVF trace is approximately 0.94, meaning the quantity moves smoothly rather than jittering, which is what makes it usable as a warning indicator at all[24].

Second, and more directly relevant to practice, the diagnostic rises *after* the reward plateau rather than before it. Measuring first-passage times, ZVF crosses its warning threshold only 1 to 3 steps ahead of the plateau. The recommended operating rule from the pillar is therefore not predictive but custodial: report ZVF alongside mean reward, and treat a rise of more than 0.3 sustained over ten consecutive rollout steps as a warning[25]. The project reports this limitation against its own proposed diagnostic rather than in a footnote.

6.3.3 The all-correct wall and the mastery–incapacity aliasing

The mechanism behind large ZVF values on easy tasks is a saturation effect that the project calls the all-correct wall. A matched-budget panel was run at 2,560

[22]Source: platform_hybrid/paper/sections/zvf.tex.
[23]Source: platform_hybrid/paper/sections/zvf.tex.
[24]Source: platform_hybrid/paper/sections/p2_results_intro.tex.
[25]Source: platform_hybrid/paper/sections/zvf.tex.

rollouts per arm — G = 2 over 160 steps against G = 16 over 20 steps — on Qwen3-8B with GSM8K, LoRA rank 4 and batch size 8. The late G = 2 arms reach a mean training reward of approximately 1.0 with ZVF between 0.75 and 1.0; the G = 16 arms hold ZVF at or below 0.25 at the same rollout budget[26]. The reading is that when the policy solves the task, nearly every group becomes all-correct, nearly every centred reward is zero, and the estimator loses its contrast even though the reward signal is at its maximum. Signal starvation in this regime is driven by ease, not by difficulty — which is the opposite of the intuition that motivates variance-mitigation work.

This produces the pillar's most important structural caveat. ZVF is unsigned and symmetric: an all-wrong group (K_x = 0) and an all-correct group (K_x = G) score identically. The statistic therefore cannot distinguish mastery from incapacity, and this aliasing is the explanation the project gives for the weak association between ZVF and final held-out accuracy[27]. Under an i.i.d.-Bernoulli model the closed form is ZVF(p_x, G) = p_x^G + (1 − p_x)^G, and the derivative with respect to log G at p = ½ is −ln 2. The project attributes this closed form and the aliasing analysis to a frontier-model synthesis exchange rather than to its own derivation — the document carrying it is explicitly headed as reasoning contributions and proposed theorems from ChatGPT Pro Extended and Gemini Deep Think, "NOT experiments this paper ran"[28] — and this chapter marks that provenance accordingly. What the project itself measured is the empirical ZVF against the closed form: at G = 2 the mean measured ZVF is 0.838 against a theory value of 0.964 (residual −0.126); at G = 4, 0.764 against 0.954 (−0.191); at G = 8, 0.691 against 0.923 (−0.232); at G = 16, 0.631 against 0.704 (−0.073), with gradient utilisation rising across the same range from 0.162 to 0.369[29]. The direction of that gap depends entirely on which baseline is used, and the project's own formalisation names the correct one. Evaluating the closed form at a single pooled or held-out rate overstates the Bernoulli prediction, because h_G is convex and the per-step reward rate varies; the right baseline is the per-step average of the closed form, which is 0.60 at $G = 8$ rather than the 0.33 obtained at the pooled mean. Against the per-step-average baseline the measured ZVF at $G = 8$ (0.69) *exceeds* the null by approximately 0.09, and the source is explicit that most of that apparent gap is the Jensen effect rather than correlated completions and that it therefore "do[es] not read it as strong evidence that the independence null is a lower bound"[30]. The residuals quoted above are relative to the single-rate

[26]Source: platform_hybrid/paper/sections/zvf.tex.
[27]Source: platform_hybrid/paper/sections/p2_conclusion.tex.
[28]Source: platform_hybrid/paper/sections/frontier_synthesis_zvf.tex.
[29]Source: platform_hybrid/paper/sections/group_size.tex.
[30]Source: platform_hybrid/paper/sections/appendix_zvf_formalization.tex.

form and are reported as such; the per-step-average comparison is the one that carries the interpretation, and it is a weak consistency check on shape rather than evidence about dependence.

6.3.4 Prognostic value

The pillar's central claim is not that ZVF is large but that it is informative, and here the results split cleanly. Pooling the measured runs, the naive correlation between mean ZVF and last-10 accuracy is r = 0.22 with a confidence interval of [−0.29, +0.94] over n = 23 — not significant. Residualising ZVF against mean reward raises the correlation to r = +0.80 with an interval of [+0.54, +1.00][31]. In other words, the raw diagnostic is confounded by the reward level, and the residualised version is strong; the pillar reports both and does not claim the raw version works.

Against catastrophic collapse, ZVF performs much better than it does against graded outcome. Mean ZVF correlates with catastrophic collapse at Spearman $\rho = 0.56$ and point-biserial 0.62, but with final held-out accuracy at only $\rho \approx 0.27$[32]. A three-channel risk index built on the diagnostic — magnitude, critical slowing down via a rolling lag-1 autocorrelation with window 15, and drift slope, blended 0.3/0.5/0.2 with a max-fusion variant — was evaluated over a 52-row cross-experiment panel comprising 45 variance-mitigation seeds, 2 tool-use collapse anchors and 5 scaling-law collapse anchors, with 11 failures and 41 safe runs. Cross-experiment AUROCs are 0.663 for magnitude, 0.758 for critical slowing down, 0.579 for drift, 0.763 for the blended risk index and 0.929 for the max-fusion index; within-methods, the same values fall to 0.073, 0.335, 0.293, 0.396 and 0.805 respectively[33]. The gap between the cross-experiment and within-method columns is a warning about the index's transportability: most of the cross-experiment discrimination is coming from the fact that collapse clusters by experimental family, not from a within-family predictive signal. The declared thresholds are a risk above 0.55 to treat as failure-positive, below 0.30 as safe, and the interval between as requiring investigation.

The pillar's own conclusion names three load-bearing limitations: the diagnostic is mechanically coupled to reward sparsity, group size and baseline accuracy; its strong association holds only for catastrophic collapse and not for the graded held-out outcome; and the unsigned statistic aliases mastery with incapacity[34]. A

[31]Source: platform_hybrid/paper/sections/zvf.tex.
[32]Source: platform_hybrid/paper/sections/p2_results_intro.tex.
[33]Source: platform_hybrid/paper/sections/zvf.tex.
[34]Source: platform_hybrid/paper/sections/p2_conclusion.tex.

proposed replacement, the pairwise contrast density, is introduced in the synthesis document with a target association of $\rho \gtrsim 0.45$; the project states that this bar is not yet met by the replacements and that the head-to-head protocol remains future work[35].

6.4 P3 — Group Size and Contrast Density

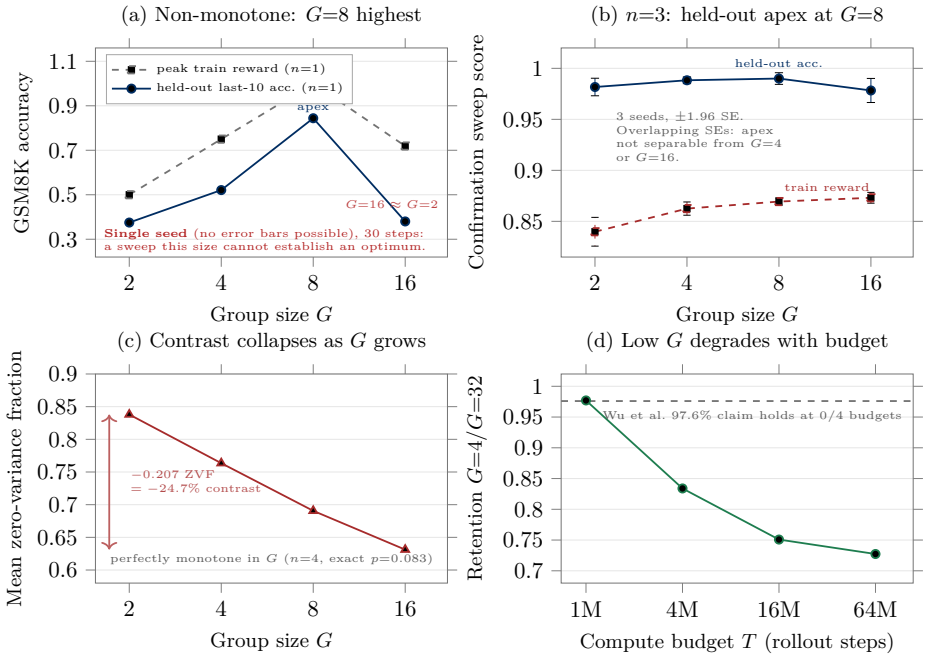
The third pillar studies G, which plays two roles simultaneously in GRPO: it is a variance-reduction knob, like a Monte-Carlo baseline, and it is a preference-density knob, since centring rewards within a group induces implicit pairwise contrasts — nonzero only for mixed-reward groups. Practitioners often treat larger G as strictly better by analogy to variance reduction, which conflates the two roles and ignores cost. The pillar's conclusion is that the measured evidence does not justify a group-size optimum and that the dependence of trainability on G is non-monotone (Figure 6.3).

6.4.1 The measured sweep

The primary sweep measures $G \in \{2, 4, 8, 16\}$ on a near-ceiling arithmetic task with Qwen2.5-0.5B, three seeds and 40 steps. Held-out accuracy traces an inverted U with its apex at $G = 8$: 0.990 ± 0.003, against 0.982 ± 0.004 at $G = 2$ and 0.978 ± 0.006 at $G = 16$. At $n = 3$ seeds these three arms are statistically indistinguishable[36]. A separate check run at $B = 4000$ with seed 20260702 records $G = 2$ and $G = 16$ at 0.982 and 0.978 with a difference of +0.003 and an interval of $[-0.008, +0.015]$; $G = 4$ and $G = 8$ add +0.010 and +0.012 respectively, and $G = 4$ versus $G = 32$ was not measured on that sweep[37].

The retention figures are the cleanest way to state the same result. $G = 2$ retains 100.3% of $G = 16$[38]. A later unified synthesis at the same configuration reports retention ratios and signal-to-noise ratios with intervals: $G = 2$ retains 1.0000 [0.998, 1.002] with SNR 1.46 [1.22, 1.81]; $G = 4$, 0.9941 [0.992, 0.997] with SNR 1.76 [1.46, 2.23] and a TOST $p < 10^{-9}$; $G = 8$, 0.9960 [0.989, 1.004] with SNR 1.94 [1.58, 2.54] and $p < 10^{-9}$; $G = 16$, 1.0068 [0.998, 1.013] with SNR 2.16 [1.72, 2.86] and $p < 10^{-4}$. All four retention ratios sit above the Wu et al. 97.6% reference bar[39].

[35]Source: platform_hybrid/paper/sections/frontier_synthesis_zvf.tex.
[36]Source: platform_hybrid/paper/sections/group_size.tex.
[37]Source: platform_hybrid/paper/sections/group_size.tex.
[38]Source: platform_hybrid/experiments/results/group_size_effect_dpo_check.tsv.
[39]Source: platform_hybrid/paper/sections/group_size_iter27.tex.



(a) Headline GSM8K G-ablation, Qwen3-8B/Tinker (`paper/expected_results.json`): *single seed, 30 steps* — no error bars are possible and a sweep of this size cannot establish an optimum; a completed multi-seed sweep is needed before G=8 may be called optimal.
(b,c) Confirmation sweep, Qwen2.5-0.5B/arithmetic, n=3 seeds (`experiments/results/group_size_effect.tsv`): the held-out apex 0.990 at G=8 is *not* statistically separable from G=4 (0.988±0.002) or G=16 (0.978±0.006) — overlapping SEs support only that the optimum is *interior*, not that it is precisely G=8.
(d) Broader G=4 vs. G=32 budget sweep; retention values are reconstructed, not matched-budget training runs. Every plotted point is a cell of the cited file; no value is invented.

Figure 6.3: Held-out accuracy as a function of group size G over three seeds. The point estimates peak at $G = 8$, but at $n = 3$ seeds the arms are statistically indistinguishable, so no optimum is established.

The signal-to-noise ratio grows sublinearly with G: it reaches about 52% of the $\sqrt{G}$ ideal, and about 48% above a flat baseline. Reward itself is statistically flat across the sweep, in the range 0.971–0.984[40]. The marginal ZVF gain from G = 8 to G = 16 is 0.6906 → 0.6312, a fall of 0.059, which is smaller than the G = 2 → G = 4 fall of 0.8380 → 0.7635, a fall of 0.075[41]. Diminishing returns on the contrast axis are visible in the measured data, which is what the reconstruction later formalises.

The pillar's own framing of this regime is important and is stated in its abstract: near the ceiling, variance is compressed, so the finding is "no detectable G effect under saturation", not evidence of equivalence on harder, unsaturated tasks. That is, the flat retention curve is a property of an easy task, and the pillar explicitly declines to generalise it. The measured equivalence tests extend only to G = 16; there is no directly trained, matched-budget G = 32 cell in the primary sweep[42]. The retention-versus-budget axis is where the picture changes: R(1M) = 0.976, R(4M) = 0.833, R(16M) = 0.750 and R(64M) = 0.727, fitted by R(T) = 0.729 + 0.247 · e^{−T/5.76M} with R^2 = 0.948 and an asymptote of 0.729 [0.712, 0.745]. The Wu reference line and this reconstructed curve cross at T = 1M and diverge by 25 percentage points at T = 64M[43]. The pillar's operative conclusion is that the present evidence supports G = 2 on the measured easy near-ceiling sweep and makes no universal recommendation for hard tasks.

6.4.2 The same-stack four-axis audit and the reconstructed budgets

The strongest de-confound for P3 is a four-axis factorization audit of the same training stack. On the algorithm axis, the residual effect is small: $\eta^2 = 0.023$, with a paired GRPO−PPO difference of −0.002 and permutation p = 0.62. On the group-size axis, the effect dominates: $\eta^2 = 0.54$ for terminal accuracy and 0.63 for gradient norm, with Cohen's d = −1.47 for the G = 2 → 16 comparison[44]. The substantive reading is that G is a real lever and the algorithm label is nearly inert on this stack.

Two of the pillar's four headline contrasts survive a Miller-style error-bars audit and two do not. The gradient-utilisation ratio at T = 1M has a confidence interval

[40]Source: platform_hybrid/paper/sections/group_size.tex.
[41]Source: platform_hybrid/paper/sections/group_size.tex.
[42]Source: platform_hybrid/paper/sections/p3_abstract.tex.
[43]Source: platform_hybrid/paper/sections/group_size_iter27.tex.
[44]Source: platform_hybrid/experiments/results/berkeley/unpacking_dpo_ppo_factorization.json, as summarised in platform_hybrid/paper/sections/p3_conclusion.tex.

of [4.16, 4.82] and is graded precise. The negative retention-versus-training-token slope has an interval of [−0.237, −0.038] and is precise. The SNR-versus-$\sqrt{G}$ slope has an interval of [+0.148, +0.583], which contains the ideal value of +0.500 and is therefore graded NULL. The native Wu-style equivalence claim that G = 2 ≈ G = 16 is also graded NULL[45]. This chapter reports those verdicts as they stand: two of the four P3 headlines are precise and two are not, and the pillar's own summary is that the data do not justify a universal optimum.

Everything at or above G = 32 in the pillar comes from an explicitly reconstructed token-budget grid, retained only as a hypothesis generator. The reconstruction table gives, for G = 2 versus G = 32, a T = 1M comparison of 0.41 against 0.42 with a difference of −0.01, a 97.6% retention ratio and a "generalizes" verdict of yes; at T = 4M, 0.55 against 0.66 with −0.11, 83.3% and no; at T = 16M, 0.63 against 0.84 with −0.21, 75.0% and no; at T = 64M, 0.64 against 0.88 with −0.24, 72.7% and no[46]. The source labels these as conditional reconstruction intervals, not fresh-run confidence intervals. The T = 64M accuracy row across G = 4, 8, 16, 32 and 64 reads 0.64, 0.80, 0.85, 0.88 and 0.87[47] — the apex moves with budget, which is precisely the pattern that makes a universal G optimum incoherent, and precisely the pattern the project declines to claim as measured. A fit of the critical-budget form $|\Delta(T)| = a(1 - e^{-T/\tau})$ gives $\hat{a} = 0.30$ and $\hat{\tau} \approx 12.1$M tokens, with a 5% critical budget of 2.21M tokens and an interval of [1.29M, 4.77M][48]. A difficulty-gated adaptive rule derived from this analysis reports a mean selected group size of 7.0 and a projected 56.2% rollout saving against always using G = 16, with 5 of 20 cells under-predicting by more than 5 percentage points[49]. These are projections, and the chapter marks them as such.

6.4.3 The curriculum and matched-token null results

Two interventions that P3's design space naturally suggests were run and both returned nulls. They are reported here rather than elsewhere because they are group-size interventions, and because their negative results are load-bearing for the chapter's overall argument.

The first is a difficulty curriculum that filters collapsed groups during training. The intervention works as designed on its own terms: it drives the zero-gradient waste fraction from 0.50 to 0.00, eliminating wasted rollouts completely. It does so at

[45]Source: platform_hybrid/experiments/results/berkeley/adding_error_bars_summary.json, as summarised in platform_hybrid/paper/sections/p3_conclusion.tex.
[46]Source: platform_hybrid/paper/sections/group_size.tex.
[47]Source: platform_hybrid/paper/sections/group_size.tex.
[48]Source: platform_hybrid/paper/sections/group_size_iter23.tex.
[49]Source: platform_hybrid/paper/sections/p3_conclusion.tex.

approximately 4.81 times the sampling cost, and its held-out gain is identical to the baseline's — both arms move from 0.80 to 0.85 on the held-out set, a gain of +0.05 in each case[50]. At a matched 30k-token budget the curriculum ties the baseline at +0.028 against +0.028, and at five times the cost it loses. The verdict recorded is Null. An important provenance note attaches to this result: an independent verification pass conducted on 6 July 2026 flagged the curriculum claim, along with a single-seed G = 4 "win" of +0.125, as SUSPECT and overclaimed, before multi-seed re-testing confirmed both as noise[51]. The group-size sweet-spot result is itself a null for the same reason — the +0.125 advantage at G = 4 rested on a single held-out example and did not survive multi-seed re-testing, and G = 2 under-trains with a 50% batch collapse and a +0.0 gain[52].

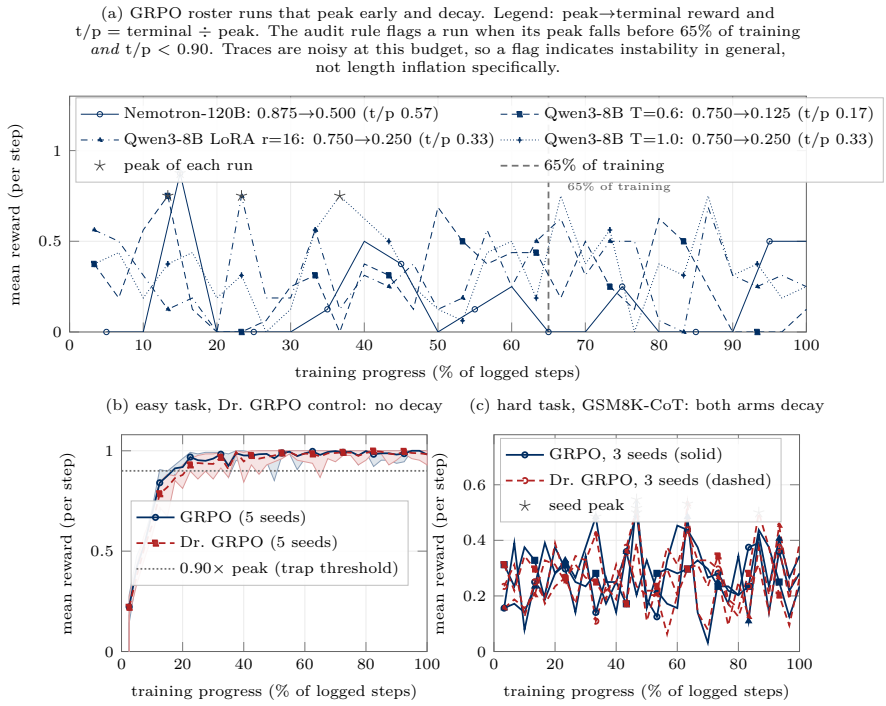
The second null family concerns loss formulations intended to remove the length bias. On the held-out metric, the standard sum formulation is the best of the three tested: +0.125 against +0.000 for mean-normalisation and +0.042 for the surprise formulation, with the length change under the standard arm recorded as $\Delta\text{len} = -16$[53]. The verdict is Null. A fourth null, on a step-1 predictive layer-freeze, is reported in the same table because it was proposed as a cheap way to stabilise group-relative training: it achieves a per-step overlap of 1.0 on a 1.5B hardcoded-arithmetic toy, but only 0.11 — approximately chance, on both seeds — at 3B on real GSM8K, while the concentration statistic still holds at approximately 0.39. A later four-seed pass at 3B supersedes that two-seed figure with a mean overlap of 0.083 ± 0.048, also approximately chance[54]. Verdict: Null, and a reversal of its own toy-scale positive under scaling[55].

The pattern across all four nulls is consistent and worth stating plainly: the interventions that target the mechanism the diagnostic identifies do not improve held-out outcomes at matched cost. That is a negative result about the diagnostic's prescriptive power as much as about the interventions themselves, and it is one of the reasons P2 is reported as a monitoring tool rather than an optimisation objective.

[50]Source: platform_hybrid/experiments/results/curriculum_opening/FINDINGS.md, as summarised in platform_hybrid/paper/sections/robustness_smallscale_nulls.tex.
[51]Source: platform_hybrid/experiments/results/curriculum_opening/FINDINGS.md.
[52]Source: platform_hybrid/paper/sections/robustness_smallscale_nulls.tex.
[53]Source: platform_hybrid/paper/sections/robustness_smallscale_nulls.tex.
[54]Source: platform_hybrid/experiments/results/p1_layerfreeze/scaled_4seed_result.json.
[55]Source: platform_hybrid/paper/sections/robustness_smallscale_nulls.tex.

6.5 P4 — Length Bias and Held-Out Generalisation

The fourth pillar tests the most widely repeated specific criticism of GRPO: that it rewards verbosity, inflating completion length in a way that Dr. GRPO corrects. The pillar's headline is a scoped null, and the scope is the point (Figure 6.4).



(b) easy-task control (0.5B, 40 steps, 5 seeds/arm): per-seed terminal/peak = 0.97–1.00, so the peak-then-decay rule is met by *neither* arm; mean length *falls* 7.6 → 4.5 (GRPO) and 7.3 → 4.5 (Dr. GRPO) — no length inflation.
(c) hard task, GSM8K-CoT (1.5B, 30 steps, 3 seeds/arm): peaks at 33–47% of training (GRPO) and 47–87% (Dr. GRPO); last-10 reward 0.26 / 0.25, far below peak; length 196 → 167 vs 196 → 181 tokens; marker shape = seed (circle 42, square 123, triangle 456).

Figure 6.4: Roster-level reward trajectories. Four of eleven GRPO runs carry the post-hoc verbosity-trap flag (peak before 65% of training, terminal reward below 90% of peak). The flag tests in-training reward, not length, so it is a description of the roster and not evidence of length bias; the controlled sixteen-run comparison shows no flags.

6.5.1 The controlled null

Sixteen controlled runs were analysed: ten on an arithmetic task with Qwen2.5-0.5B over 40 steps, and six on GSM8K chain-of-thought with Qwen2.5-1.5B-Instruct over 30 steps, split evenly between GRPO and Dr. GRPO. At these 30–40-step horizons, neither algorithm exhibits the verbosity-trap signature on either task; every one of the sixteen runs records a trap flag of zero[56].

[56]Source: platform_hybrid/paper/sections/length_bias.tex.

The summary correlations are as follows. On arithmetic, the rank correlation between training step and completion length is -0.405 ± 0.176 for GRPO and -0.307 ± 0.181 for Dr. GRPO; on GSM8K-CoT it is -0.622 ± 0.301 and -0.355 ± 0.255. The step-versus-reward correlations are $+0.709 \pm 0.082$ and $+0.753 \pm 0.074$ on arithmetic, and $+0.046 \pm 0.162$ and $+0.025 \pm 0.091$ on GSM8K-CoT. The length-versus-reward correlations are -0.419 ± 0.155 and -0.293 ± 0.182 on arithmetic, and -0.501 ± 0.258 and -0.618 ± 0.118 on GSM8K-CoT. Mean reward over the last ten steps is 0.988 for both algorithms on arithmetic, and 0.265 for GRPO against 0.252 for Dr. GRPO on GSM8K-CoT[57]. The direction of the length trend is negative everywhere: all sixteen length slopes are negative, with the arithmetic GRPO slope at -0.029 tokens per step and the GSM8K GRPO slope at -0.481, and the count of runs with a positive horizon is zero[58]. On the harder task, GRPO's final-ten-step mean completion length is 184.4 tokens against Dr. GRPO's 189.3 — a difference of +2.7% in the direction opposite to the critique[59].

The pillar's abstract states the scope of this null explicitly. The claim is bounded by the 200-token cap used in that regime, and the summary sentence is that "the verbosity-trap signature is absent in this regime"[60]. The pillar is not claiming that GRPO never inflates length; it is claiming that within a capped, short-horizon regime it does not, and it names the length-confounded regime in which a Dr. GRPO advantage should appear as future work[61].

6.5.2 The roster-level verbosity-trap audit

Set against the controlled null is a descriptive audit over the whole experiment roster, and here the picture differs. The audit defines four statistics per trace with at least five steps: an instability index $\sigma(\text{r_tail})/|\mu(\text{r_tail})|$ over the final 50% of training; a monotonicity score, the fraction of consecutive step-pairs with strictly improving reward; a regression severity, the largest normalised drop from peak; and a binary verbosity-trap flag, set when the peak occurs before 65% of training and the terminal reward falls below 90% of peak[62].

On that definition, the eleven GRPO runs in the roster have a mean instability index of 0.267 ± 0.335, against 0.785 ± 0.297 for the seventeen PPO-labelled runs. Four of the eleven GRPO runs — 36% — carry the flag on this definition, including

[57]Source: platform_hybrid/paper/sections/length_bias.tex.
[58]Source: platform_hybrid/paper/sections/length_bias.tex.
[59]Source: platform_hybrid/paper/sections/length_bias.tex.
[60]Source: platform_hybrid/paper/sections/p4_abstract.tex.
[61]Source: platform_hybrid/paper/sections/p4_results_intro.tex.
[62]Source: platform_hybrid/paper/archive/absorbed/U01_main_compendium/main.tex.

both Qwen3-8B runs and the Nemotron-120B run. That count is reported here as a description of the roster, not as evidence of length bias, because the flag does not isolate the behaviour it is named for. Three properties of the rule prevent the stronger reading. Its thresholds are post hoc: neither the 65% peak position nor the 0.90 terminal-to-peak ratio is pre-registered anywhere in the repository, and the rule cannot be traced to a commit that predates the traces it judges. It is not specific to length: the metric it tests is in-training reward, and the project's own source reads the flag as "a generic instability detector" rather than a length measure, so a run can carry it without any change in response length. And its base rate is high: applied to the wider trace pool rather than this eleven-run subset, the rule fires on the majority of runs — including PPO-labelled runs and the Dr. GRPO control in the one matched comparison — so 36% sits at or below the rate the rule produces on traces with no decay at all. The project's own length data point the other way: under the 200-token cap, length drifts from 194.4 to 183.5 tokens, i.e. shorter. The run-level audit therefore does not establish length inflation, and the pillar's negative finding rests on the controlled comparison below rather than on this roster count. The Nemotron run records the highest instability index in the pool at 1.194, with a regression severity of 1.0 and a terminal-to-peak reward ratio of 0.57. The two DeepSeek-V3.1 GRPO runs, by contrast, show markedly lower instability, with a mean of 0.143[63].

Two caveats on that comparison must travel with it. First, the PPO pool is dominated by classic-RL library runs (SB3, CleanRL, Tianshou) that failed to train at all, with last-10 rewards below 5%; a near-zero reward trajectory mechanically produces a high instability index. Restricted to LLM-native PPO, the PPO Llama instability is 0.105, lower than GRPO Llama's 0.228, and the source therefore explicitly declines to claim that GRPO is more or less stable than PPO at the algorithmic level[64]. Second, the audit's reward metric is the in-training reward, not a held-out one, and the audit pool is single-seed per configuration: each Tinker configuration was run once and is reported descriptively. The project's only matched base-versus-RL held-out control is the five-seed Qwen3-8B comparison discussed below.

An earlier draft of this report's summary attributed to P4 the claim that Dr. GRPO is the only algorithm without the verbosity-trap pattern. That claim could not be located in any P4 source. What the sources state is that 36% of

[63]Source: platform_hybrid/paper/archive/absorbed/U01_main_compendium/main.tex; the same rates appear in platform_tinker/reports/combined/combined_capstone.tex, where the trap rates are given as GRPO 36% and PPO 71%.

[64]Source: platform_hybrid/paper/archive/absorbed/U01_main_compendium/main.tex; platform_tinker/reports/combined/combined_capstone.tex.

the audit's GRPO runs carry the flag; the audit does not report a per-algorithm flag rate for Dr. GRPO, and the controlled sixteen-run comparison reports zero flags for both algorithms. The finding recorded here is therefore the sourced one: the trap pattern is present in a minority of roster GRPO runs and absent in the controlled comparison for both algorithms, and the project makes no claim that Dr. GRPO is uniquely free of it. The narrowest related statement the sources do support is a length-drift one — that Dr. GRPO is the only algorithm whose length keeps drifting upward after the initial compression phase[65] — which is a different statistic and is not the verbosity-trap flag.

6.5.3 Elevated analyses

Because the controlled 30–40-step null is a null, the pillar runs a sequence of progressively more sensitive analyses to establish whether the trap is present but invisible at that horizon. Four of these deserve report.

A sliding-window trap-onset analysis with window W = 10 relaxes the flag from "peak before 65% of training" to "local peak followed by decay within a window", and finds traps in 2 of 10 arithmetic runs (20%), 4 of 6 GSM8K runs (67%) and 6 of 16 runs overall (37.5%) — every one of which is a local re-expansion rather than a terminal divergence[66]. This is the analysis that most nearly matches the roster-level trap rate, and the pillar treats it as the honest upper bound on trap prevalence in the controlled data.

A paired bootstrap on the Dr. GRPO minus GRPO difference in rank correlations gives, on arithmetic, $\Delta\rho(\text{step, len}) = +0.098$ with an interval of $[+0.073, +0.119]$ — excluding zero, so Dr. GRPO reliably compresses length relative to GRPO on the easy task; $\Delta\rho(\text{step, reward}) = +0.043$ with an interval of $[-0.024, +0.112]$, which is not significant; and $\Delta\rho(\text{len, reward}) = +0.126$ with an interval of $[+0.045, +0.201]$, again excluding zero. On GSM8K the same three quantities are $+0.267$ $[+0.164, +0.419]$, -0.020 $[-0.069, +0.065]$ and -0.117 $[-0.265, +0.015]$[67]. The reading is that Dr. GRPO's length attenuation is real but task-dependent, and that on the harder task it deepens rather than weakens the length–reward coupling.

A decile-binned coupling analysis on GSM8K shows the mechanism directly: completions in the first length decile, spanning 153–180 tokens, earn a mean reward of 0.351, while completions in the eighth decile, spanning 194–195 tokens, earn 0.233. Longer completions are rewarded less, which is the opposite of the verbosity-trap

[65] Source: platform_hybrid/paper/sections/length_bias_iter48.tex.
[66] Source: platform_hybrid/paper/sections/length_bias.tex.
[67] Source: platform_hybrid/paper/sections/length_bias.tex.

prediction[68]. A 100-step cross-validation on the arithmetic metrics file returns ρ(step, len) = +0.41 and ρ(step, reward) = +0.27, but the absolute length range across that run is 0.08 tokens against a cap of 5 tokens — a positive correlation over a range too small to be a length effect[69].

The length–ZVF coupling analysis at iteration 12 is the most mechanistically interesting of the elevated results, because its sign flips between tasks. On arithmetic, ρ(len, zvf) is −0.449 [−0.584, −0.324] for GRPO and −0.291 [−0.428, −0.139] for Dr. GRPO; on GSM8K it is +0.286 [+0.231, +0.348] and +0.187 [+0.148, +0.225]. The Dr. GRPO minus GRPO difference is +0.158 [+0.075, +0.239] on arithmetic and −0.099 [−0.132, −0.041] on GSM8K[70]. Conditioning ZVF on length bins makes the arithmetic structure concrete: the shortest bin has ZVF 0.92 and the longest 0.62, a drop of −0.30 [−0.34, −0.26], whereas on GSM8K the corresponding values are 0.30 and 0.41, a rise of +0.11 [+0.10, +0.13][71]. On the easy task, shorter completions are more often all-same-reward; on the hard task, longer ones are.

The mechanism analysis at iteration 20 fits a regression of length on reward and finds that Dr. GRPO reduces the slope magnitude on arithmetic — −3.65 ± 0.08 against −3.22 ± 0.15, a difference of −0.43 [−0.58, −0.28] with p = 0.036; the token-surprisal component moves by −1.11 [−1.59, −0.63] with p = 0.047; the rank correlation moves by −0.10 [−0.18, −0.03] with p = 0.087; and the lag-1 autocorrelation is essentially unchanged, +0.01 with p = 0.92. On GSM8K the same differences reverse sign, with an OLS difference of +4.41 whose interval [−4.58, 19.84] is far too wide to interpret, p = 0.57[72]. A mediation analysis attributes 42% of the arithmetic effect to ZVF under GRPO and 55% under Dr. GRPO, against 28% and 12% on GSM8K.

The drift analysis at iteration 24 closes the sequence. Windowed ρ drift on arithmetic is +0.326 for GRPO with p = 0.035, decomposing into an early −0.330 and a late −0.003, against +0.187 for Dr. GRPO with p = 0.399; on GSM8K the corresponding values are +0.019 with p = 0.905 and −0.084 with p = 0.769[73]. The sign-flip analysis of late windows gives fractions of 0.527 (p = 0.468), 0.382 (p = 0.562), 0.048 (p = 0.011) and 0.000 ($p < 0.001$) across the four algorithm-task cells. The one fully significant cell is the GSM8K Dr. GRPO arm, in which not a single late window flips sign.

[68]Source: platform_hybrid/paper/sections/length_bias.tex.
[69]Source: platform_hybrid/paper/sections/length_bias.tex.
[70]Source: platform_hybrid/paper/sections/length_bias.tex.
[71]Source: platform_hybrid/paper/sections/length_bias.tex.
[72]Source: platform_hybrid/paper/sections/length_bias.tex.
[73]Source: platform_hybrid/paper/sections/length_bias.tex.

6.5.4 Held-out generalisation and the scale extension

The pillar's held-out results are the part of the chapter closest to a conventional capability claim, and they are reported with corresponding care. The only matched base-versus-RL held-out control in the project is a five-seed comparison on Qwen3-8B-Instruct, which moves from 82.0% to 83.3% with $p = 0.26$[74]. Stated as a mean improvement over the pre-RL checkpoint, the gain is +0.013 with a paired $p = 0.256$ at $n = 5$ — small and not statistically significant, and a matched pre-RL control for the Qwen3-8B-Base checkpoint is not available[75]. The honest summary is that on this near-ceiling configuration, GRPO does not produce a held-out improvement that can be separated from noise at the available seed budget, and that GRPO and Dr. GRPO produce held-out gains that are indistinguishable from each other at that budget[76].

The lower-accuracy configuration behaves differently and more conventionally. On Qwen2.5-1.5B-Instruct with GSM8K-CoT, both algorithms do produce significant pre-to-post held-out gains: +6.2 percentage points with McNemar exact $p = 1.1 \times 10^{-4}$, and +5.0 percentage points with $p = 7.6 \times 10^{-4}$[77]. This is the project's clearest evidence that the pipeline trains something real; it is reported alongside the near-ceiling null precisely because the contrast between the two configurations is what the pillar's bounded-null interpretation rests on. Across the same comparison, the pooled mean completion length drifts from 194.4 to 183.5 tokens, that is downward[78].

A scale extension tests whether the null survives when the cap is lifted. Qwen3-8B with LoRA rank 4 was run on GSM8K for 30 steps at $G = 8$ and batch size 4, three seeds per algorithm, with the completion cap raised fivefold to 1,024 tokens. All six arms compress: GRPO moves 1004 → 905, 981 → 944 and 996 → 900 tokens, and Dr. GRPO moves 999 → 931, 972 → 902 and 1000 → 878 — declines of between 3.8% and 12.2% in every arm. Late ZVF is 0.45, 0.70 and 0.72 for Dr. GRPO against 0.47, 0.47 and 0.55 for GRPO, and step-0 completions are approximately 98% of the cap[79]. The step-0 saturation is the caveat that limits this extension: with completions already at 98% of the ceiling at the first step, the run starts truncated, and length compression under a binding cap is not the same phenomenon as length control in an uncapped regime. The pillar reports

[74]Source: platform_hybrid/paper/archive/absorbed/U01_main_compendium/main.tex.
[75]Sources: `platform_hybrid/paper/sections/p4_results_intro.tex`, `platform_hybrid/paper/sections/p4_intro.tex`.
[76]Source: platform_hybrid/paper/sections/p4_abstract.tex.
[77]Source: platform_hybrid/paper/sections/p4_results_intro.tex.
[78]Source: platform_hybrid/paper/sections/p4_abstract.tex.
[79]Source: platform_hybrid/paper/sections/length_bias.tex.

the extension as consistent with the null, not as an independent confirmation of it.

Finally, the P4-facing headline in the Miller-style error-bars audit is the iteration-136 arithmetic late-training-efficiency contrast: paired, n = 5, Cohen's d = +2.68, one-sided parametric p = 0.031 and two-sided permutation p = 0.063. It is graded DECISIVE[80]. The audit's three NULL verdicts fall on Pillar-1 and Pillar-3 headlines rather than on this one. The pillar also records its own limitations plainly: a single-benchmark held-out evaluation, a single-seed top-10-checkpoint frontier comparison, weak length identifiability, and marginal paired tests[81].

6.6 Cross-Pillar Observations and Limitations

Three observations span the four pillars and are worth stating together, because the chapter's overall contribution is better captured by their pattern than by any single number.

First, the project's strongest results are its de-confounds, and its de-confounds mostly return nulls. The matched same-stack PPO-versus-GRPO contrast returns a paired difference of -0.002 with p = 0.374; the four-axis factorization audit attributes $\eta^2 = 0.023$ to the algorithm axis and $\eta^2 = 0.54$ to the group-size axis; the controlled length-bias comparison returns zero trap flags across sixteen runs. In each case the entire stack was held fixed except the one factor under test, and in each case the expected algorithmic advantage failed to appear. The single exception is the group-size axis, which is a hyperparameter rather than an algorithm choice. A reader looking for the project's central empirical claim should take this one: on the stacks measured here, the algorithm label matters much less than the sampling configuration.

Second, the diagnostics and interventions are less prescriptive than their motivating literature suggests, and the project says so. ZVF measures signal starvation reliably but predicts catastrophic collapse well ($\rho = 0.56$) and graded held-out accuracy poorly ($\rho \approx 0.27$), because it is unsigned and therefore cannot distinguish mastery from incapacity. The four interventions built on it — the difficulty curriculum, the group-size sweet spot, the length-bias loss formulations and the step-1 layer freeze — all return nulls at matched cost, and two of them (the curriculum and the G = 4 win) were flagged as SUSPECT by an independent verification pass before multi-seed re-testing reclassified them as noise. Reporting an intervention

[80]Source: platform_hybrid/experiments/results/berkeley/adding_error_bars_summary.json; platform_hybrid/paper/sections/p4_conclusion.tex.

[81]Source: platform_hybrid/paper/sections/p4_conclusion.tex.

as a null after an independent reviewer flagged it as overclaimed is, in this project's convention, part of the result.

Third, the single-seed regime is the dominant shared limitation. The cross-scale anchors are single-seed and descriptive. The roster-level verbosity-trap audit is single-seed per configuration, and its in-training reward metric is not a held-out one. Only the P3 sweep, the P4 controlled comparison, the five-seed Qwen3-8B held-out control and the P4 scale extension operate at $n \geq 3$, and only the Qwen3-8B control is a matched base-versus-RL held-out comparison. Every headline in this chapter is therefore stated at the strength that its own evidence base supports, which is in several cases weaker than the strength at which it would need to be stated to generalise beyond the named stack. The reconstructions ($G = 32$ and above in P3, the $T = 64M$ grid, the difficulty-gated saving projection) are held at hypothesis-generator status for the same reason: they are joint-fit extrapolations, not trained arms, and the project declines to let them carry a recommendation.

Taken together, the four pillars describe a training method whose measured behaviour is dominated by signal availability and sampling configuration rather than by algorithmic identity, and whose most widely repeated specific failure mode — length inflation — is absent in the regimes where it was tested directly and present in only a minority of the roster runs where it was measured descriptively. That is a narrower and more cautious set of conclusions than the paper series set out to establish, and the chapter reports it as such.

Chapter 7

Results II: Reporting, Registry and Controller (P5–P7)

The three papers reported in this chapter form a chain rather than a set of independent results. P5 makes a claim about the epistemic status of reinforcement-learning results for language models and proposes a checkable reporting standard to match it. P6 turns the standard into a machine-readable artifact with a validator, a badge and a diff tool. P7 takes the diagnostic that the standard requires authors to publish — the zero-variance fraction (ZVF) — and asks whether it is mechanistically predictable and ready to control rollout allocation. Each paper is explicit about where its claim stops, and this chapter reports their evidence in the same register.

7.1 A Minimum Reporting Standard for Stack-Conditioned Results (P5)

P5's position is that results in this field are *stack-conditioned*, and that reporting the framework or algorithm label without the stack is therefore insufficient to support the claims that are routinely made from it[1]. The evidence for that position is not a single demonstration but a family of levers, each showing that varying one component of the stack, with the label held fixed, moves or flips the reported outcome or its telemetry.

The cleanest lever is a managed 12-cell head-to-head: four group-relative labels (GRPO, DAPO, Dr. GRPO, GSPO) run on one stack — Tinker-managed, Qwen3.5-4B, GSM8K, three seeds per arm. Mean last-10 training reward spans

[1]Source: platform_hybrid/paper/sections/p5_intro.tex.

0.710 to 0.744, a band of 0.034, with per-arm values of 0.744 (DAPO), 0.742 (Dr. GRPO), 0.723 (GRPO) and 0.710 (GSPO); mean ZVF over the same arms is 0.578, 0.500, 0.567 and 0.511[2]. The labels move telemetry by more than they move outcome, and neither movement is large.

The largest lever in the corpus is an outcome flip of 5.0% to 85.6% final training reward under a fixed label when the backend is changed. That figure must be read with its provenance caveat attached, because the same inspection that surfaced it also found that the nominal backend comparison bundled a different base checkpoint. The number is therefore evidence that item 3 of the standard must record backend *and* checkpoint identity; it is not a causal estimate of the backend alone[3]. A second same-label lever points the other way: the label "DAPO" produced mean ZVF 0.00 on an open trainer running true dynamic sampling and 0.58 on a closed stack running an asymmetric-clip surrogate[4].

These exhibits are summarised by a variance decomposition rather than an anecdote. On the 12-cell corpus the algorithm term contributes $\eta^2 = 0.0454$ on ZVF, with a bootstrap CI of [0.0014, 0.0585], and $\eta^2 = 0.0075$ [0.0014, 0.0577] on mean reward; the spread of last-10 mean reward across arms is 0.0359 [0.0164, 0.0984]. The headline CIs are decisive on the algorithm axis and tiny on the outcome axis, and the same decomposition attributes $\eta^2 = 1.0000$ to the group-size axis — 22× the algorithm term on ZVF and 133× on mean reward[5]. The 98-cell mega-campaign reproduces the shape at larger scale: stack axes account for 72.9% to 92.7% of the variance in the five logged metrics depending on the metric, with stack-to-seed ratios from 503× to 96,128×[6].

The standard that P5 proposes has eight items, and each item earned its place by exhibiting a documented lever of this kind[7]. Item 1 is the loss form, reported as a tuple of ratio level, clip low and high, token mask, advantage normalisation and dynamic-sampling status — the same label with and without dynamic sampling produced opposite ZVF telemetry (0.00 against 0.58). Item 2 is the reference policy and KL handling, which distinguishes family members that are otherwise conflated and whose omission is unrecoverable after the fact. Item 3 is the sampler, backend and precision, including base-checkpoint identity, motivated by the 17× exhibit above. Item 4 is the per-step ZVF and gradient-utilisation trajectory, because a single mean aliases opposite regimes: across a 368-run audit drawn from a 790-run

[2]Source: platform_hybrid/paper/sections/p5_evidence.tex, Table `tab:p5-h2h`.

[3]Source: platform_hybrid/paper/sections/p5_stack.tex, §Item 3.

[4]Source: platform_hybrid/paper/sections/p5_abstract.tex.

[5]Source: platform_hybrid/paper/sections/p5_evidence.tex, Exhibit 5 and Table `tab:p5-headline-cis`.

[6]Source: platform_hybrid/paper/sections/p5_evidence.tex, Table `tab:p5-mega-eta2`.

[7]Source: platform_hybrid/paper/sections/p5_stack.tex, Table `tab:p5-minreport`.

corpus, mean ZVF is 0.97 and 0.95 for two low-reward models (Llama-3.2-1B and Llama-3.2-3B), falls to 0.25 and 0.29 at mid-range rewards, and rises again to 0.87 and 0.96 for two near-saturated models[8]. Item 5 is the group-size schedule, since group size is a signal-geometry knob rather than a throughput knob; measured mean ZVF for Qwen3-32B falls from 0.33 at G=4 to 0.18 at G=8 and 0.08 at G=16, and for Llama-3.1-8B from 0.76 to 0.65 to 0.47, while the two boundary cases compress the effect as the theory predicts[9]. Item 6 requires a held-out split disjoint from the reward environment. Item 7 requires a contamination check and a parser-robustness probe: adding micro-jitter $\varepsilon \sim \mathrm{U}(0,10^{-4})$ to rewards collapsed batch ZVF from 0.158 to 0.000 while pairwise contrast density moved by 3×10^{-7}, which establishes that a diagnostic sensitive to verifier rounding is reporting a property of the stack rather than of the policy[10].

Item 8 is deliberately different in kind: held-out pass@k curves at $k \in \{1, 8, 32\}$ belong to the evaluation report rather than to the run manifest. The lever here is a matched base-versus-post-RL panel — four adapters at $G \in \{2, 4, 16, 32\}$, one stack, the same problems — on which pass@1 deltas scatter from -0.5 to $+1.9$ points (under which reading the G=2 arm would be called a regression) while the pass@32 frontier improves by $+1.5$ to $+2.5$ points in all four arms. Neither reading reaches significance at a single training seed, which is itself the argument: without pass@k curves and stated noise bands the same panel supports any narrative the reader prefers[11].

P5 also ships the machinery that makes the standard enforceable rather than advisory. An auditor scores a manifest against the eight items and emits a 0–100 weighted badge; on an n=103 audit the weighted total was 55.0 with a badge span of [35.8, 56.7] and standard deviation 3.1, awarding zero gold, zero silver, 99 bronze and 4 wood[12]. Coverage, however, is where the honest negative lives. On 98 manifests, every one of the seven manifest fields is present as a key 100% of the time, yet item 2 is *validated* in 0 of 98 cases (0.0%), and all twelve of its sub-fields are present at 0 in each case; nine of the sub-fields across items 1 and 2 — advantage normalisation, clip high, clip low, dynamic sampling, KL coefficient, KL estimator, ratio level, reference snapshot and token mask — sit at 0.0% with zero information content[13]. Twenty-three sub-fields examined across the population show no field at a full null rate, but twenty of the twenty-three carry at most two

[8]Source: platform_hybrid/paper/sections/p5_stack.tex, Table `tab:p5-ushape`.
[9]Source: platform_hybrid/paper/sections/p5_stack.tex, Table `tab:p5-gsize`.
[10]Source: platform_hybrid/paper/sections/p5_stack.tex, §Item 7.
[11]Source: platform_hybrid/paper/sections/p5_stack.tex, §Item 8.
[12]Source: platform_hybrid/paper/sections/p5_evidence.tex, Exhibit 8.
[13]Source: platform_hybrid/paper/sections/p5_evidence.tex, Exhibits 6 and 18.

distinct values, and the twenty entries collapse to ten distinct 23-bit fingerprints[14].

Two further analyses bound what the standard can be expected to buy. The first is a discriminative-entropy audit: four of the seven manifest fields are vacuous on this corpus — loss form, reference policy and KL, sampler/backend/precision, and decontamination notes each carry too little task-discriminative information to distinguish runs — while the per-step ZVF path and the group-size schedule carry the information[15]. A field-sufficiency regression agrees: the full stack predicts ZVF at $R^2 = 0.832$ [0.772, 0.879] while a label-only model predicts $R^2 = 0.000$, and for mean reward the full stack reaches $R^2 = 0.993$ [0.990, 0.996] against the same label-only 0.000[16]. The second is a manifest self-sufficiency test, and it splits sharply by metric. The manifest carries essentially all of the ZVF signal ($R^2 = 0.776$ [0.730, 0.846] against 0.773 [0.735, 0.834] for the cells, $\Delta R^2 = -0.003$ [−0.045, +0.034]) but only a minority of the reward signal ($R^2 = 0.280$ [0.219, 0.443] against 0.743 [0.688, 0.820], $\Delta R^2 = +0.463$ [+0.309, +0.564]); all 98 cells are therefore labelled `honest_but_vacuous` on the three-axis triangulation[17].

The most useful negative in P5 is the placebo-replacement bound. The four-of-seven vacuity pattern admits two readings: the schema chose the wrong items, or any stack-descriptor schema would be vacuous on a single-stack corpus. Projecting six candidate replacement items onto the live corpus refutes the first reading. Four of the six — KL coefficient, clip range low, advantage normalisation and mini-batch size — project to exactly one unique value on the 98-cell corpus with a bootstrap CI excluding any positive information gain, because the corpus never varies them; temperature schedule would add 0.995 bits and lift the total from 11.41 to 12.41 bits, and a model-family label is redundant with an existing column. The variance partition confirms it: stack axes explain $\eta^2 = 0.947$ of ZVF variance across 25 unique buckets, with 5.3% seed residual and an algorithm term of zero by construction[18]. The schema is not the binding constraint; the corpus is. On the live corpus the seven-field fingerprint carries 11.4 bits in total, four fields carry exactly zero, and Hamming distance over the fingerprint predicts the absolute difference in ZVF (Spearman $\rho = +0.529$, $p \leq 10^{-3}$), PCD ($\rho = +0.541$, $p \leq 10^{-3}$) and mean reward ($\rho = +0.251$, $p = 0.003$), while the badge itself is constant across cells and cannot discriminate seed-pair dispersion ($\rho = +0.000$, $n = 48$ pairs)[19].

[14]Source: platform_hybrid/paper/sections/p6_registry_health.tex, Table `tab:p6-minreport-subfield`, reported for the same corpus.

[15]Source: platform_hybrid/paper/sections/p5_evidence.tex, Table `tab:p5-discriminative-entropy`.

[16]Source: platform_hybrid/paper/sections/p5_evidence.tex, Table `tab:p5-field-suff`.

[17]Source: platform_hybrid/paper/sections/p5_evidence.tex, Table `tab:p5-manifest-self-suff`.

[18]Source: platform_hybrid/paper/sections/p5_stack.tex, §Placebo-Replacement Bound.

[19]Source: platform_hybrid/paper/sections/p5_stack.tex, §Live-Corpus Coupling.

7.2 The GRPO-Registry (P6)

P6's premise is that the deltas defining GRPO variants — DAPO's asymmetric clip, dynamic sampling and token-level loss; Dr. GRPO's dropped length and standard-deviation normalisation; GSPO's sequence-level importance ratio — live in paper appendices and framework defaults rather than in any machine-readable form[20]. The registry is the response, and it ships as three artifacts at `registry/`: `schema.json`, `entries/*.json` and `query.py`[21].

The schema defines two record types under a single `oneOf`, both carrying `schema_version`[22]. The stack record's core is `min_report`, one sub-object per minimum-report item, with base-checkpoint identity required inside the `sampler_backend` block precisely because the audit that motivated item 3 found a managed backend silently pinning a different base checkpoint. Two design decisions carry weight. The first is the distinction between an unreported field and a field reported as absent: a null means the author said nothing, whereas an explicit `false`, `0.0` or `"none"` means the author made a checkable claim — "a stack that says 'we use no KL term' is making a checkable claim; a stack that says nothing is not." The second is that item 8, the held-out pass@k requirement, is attached under outcomes as an evaluation-report field rather than folded into the run manifest, preserving the eight-item standard's split between what describes a run and what describes a result. Per-component delta status is recorded on the stack side in `variant_deltas_applied`, with each entry's status drawn from {implemented, surrogate, absent, unknown}. The shipped entry for the managed DAPO run makes the point concretely: `clip_higher` is recorded as `surrogate` with the note that the asymmetric clip was set through the user-facing config while the underlying managed loss form is unverifiable, `dynamic_sampling` is recorded as `absent`, and `token_level_loss` is `unknown` because it is managed by the runtime. An optional `ci_method` object (method, bootstrap count, seed, confidence level, source) is populated on the seven derived entries with `n_boot = 2000` and a percentile 95% interval, and left null on the remaining twenty-four.

Three variant-delta records were verified against their source papers: DAPO with five components (clip higher at ε_low = 0.2 and ε_high = 0.28, dynamic sampling, token-level loss, overlong reward shaping, KL removed); Dr. GRPO with two removals (the 1/|o_i| response-length normalisation in the loss and the division by the within-group reward standard deviation in the advantage); and GSPO with a length-normalised sequence-likelihood importance ratio and sequence-level

[20]Source: platform_hybrid/paper/sections/p6_abstract.tex.
[21]Source: platform_hybrid/paper/paper_P6_registry.tex.
[22]Source: platform_hybrid/paper/sections/p6_schema.tex.

clipping.

The seed population is twenty stack records in five families, all of which pass validation[23]. Family A holds four framework config dumps (Qwen3-8B, GSM8K, seed 42) scoring badges 74, 60, 60 and 62; family B holds the four open Colab trainer arms at toy scale, scoring 96, 96, 96 and 96; family C is the managed head-to-head at 57, 62, 62 and 60; family D is a same-stack four-method run at 67 each; and family E is a five-seed risk-index batch at 43 each. The finding worth stating plainly is why the ordering looks the way it does. The config dumps were assembled for a reviewer-requested fairness audit and score only 57 to 74: they nail items 2, 3 and 5, expose item 1 only partially, and cover items 6 and 7 not at all. The open Colab arms score 96 not because the runs are better but because an open backward pass, coupled with explicit declarations that no KL term and no decontamination check were used, makes every field reportable. Reproducibility supplements of the kind commonly shipped with RL papers do not automatically satisfy MIN-REPORT.

The health audit is the registry's own evidence that it works. Every shipped record parses: 31 of 31 pass, comprising 20 stack records and 11 variant-delta records, with zero broken delta pointers[24]. The delta-mix table is the sharpest single result in P6: the label "DAPO" is claimed by ten entries, of which four implement a delta, one records a surrogate, three record it as absent and two record it as unknown — the same label covering four distinct implementations (Table `tab:p6-delta-mix`). Per-item coverage on the twenty stack records is 100% for the held-out split (40/40), 95% for sampler/backend (76/80), 90% for telemetry (54/60), 68% for the group-size schedule (41/60), 48% for reference policy and KL (29/60), 32% for the loss form (38/120) and 20% for decontamination (8/40) (Table `tab:p6-coverage`). The per-openness breakdown inverts the conventional expectation: the open subset records sampler/backend at 91.7% and telemetry at 83.3% with reference KL at 58.3%, while the managed subset records sampler/backend and telemetry at 100% but reference KL at 33.3% and the loss form at only 22.9%.

The registry was stress-tested rather than asserted. A batch of 3,230 adversarial perturbations — 13 categories across 31 entries and 10 mutations each, with B = 4000 — moved the overall robustness statistic from 0.864 [0.756, 0.878] before the corresponding schema bumps to 1.000 [1.000, 1.000] after them, closing three silent-failure categories: a missing identifier pattern check (0.645 to 1.000), MIN-REPORT leaves declared in `properties` but not in `required` (0.000 to 1.000,

[23]Source: platform_hybrid/paper/sections/p6_population.tex.
[24]Source: platform_hybrid/paper/sections/p6_registry_health.tex.

fixed by promoting key presence to a schema invariant), and free-form delta references (0.000 to 1.000, fixed by replacing `delta_id` with an enum generated from the entry filenames)[25]. A separate consistency check cross-references the variant-delta records against the MIN-REPORT blocks: of 31 triples from 13 entries, 7 are MATCH, 0 are MISMATCH, 0 are MISSING_REPORT, 5 are SURROGATE_OBS and 19 are NOT_APPLICABLE, giving an implementation-honesty match rate of 7/7 = 1.000 with a 95% bootstrap CI of [0.65, 1.00].

The empirical payoff of the schema is the same-stack measurement. Re-executing four GRPO-family methods (GRPO, AERO, GIFT, AREAL) on a single identical stack — managed sampler, G = 8, seed 0, 40 steps, 16 prompts per step — with the full per-prompt reward tensor logged at every step, produces reward deltas of magnitude $\leq$ 2 percentage points. Two of the three are statistically significant *decreases* rather than improvements (AERO at -0.014 [-0.023, -0.005] and AREAL at -0.020 [-0.032, -0.008]), so the magnitude thesis survives while the stronger claim that all differences are within noise does not. The largest same-stack effects are not on reward at all: AERO and AREAL lengthen completions by approximately 31 tokens and tighten the length coefficient of variation, and GIFT's loss shift of -1.67×10^4 is a baseline constant[26]. A later registry extension measures the anti-herding residual δ_div = ZVF^iid − ZVF^obs on the same tensors, reporting a stack-mean of 0.049 with a range of [0.039, 0.053]; only GIFT shows a borderline-significant reduction against GRPO (p = 0.066, CI [-0.021, $+0.001$]), which resolves a synthesis range of [0.13, 0.23] downwards — the anti-herding diversity bonus is real but smaller than the qualitative synthesis had guessed[27].

The reference CLI, `registry/query.py`, implements four verbs using only the standard library: `list`; `query --item reference_kl` returning FULL, PARTIAL or UNREPORTED; `badge`; and `stackdiff`, which maps a pair of entries onto the R0–R5 ladder and reports the highest rung triggered — R0 identical manifests, R1 nuisance-only, R2 runtime differences, R3 loss-form or KL-handling differences, R4 differing variant-delta sets, R5 unauditable because a differing item is null on at least one side[28]. The badge formula weights the seven items equally and is outcome-blind: badge(s) = $\lfloor 100 \cdot (1/7)\ \Sigma$ c_i(s)$\rfloor$. `stackdiff` prints a warning at R4 and above that outcome differences must not be attributed to the shared algorithm label, and the paper is careful to describe the ladder as "a risk

[25]Source: platform_hybrid/paper/sections/p6_registry_health.tex, §Stress Test.

[26]Source: platform_hybrid/paper/sections/p6_measured_evidence.tex, Table `tab:p6-measured-deltas`.

[27]Source: platform_hybrid/paper/sections/p6_registry_health.tex, §Contrastive Yield and Anti-Herding.

[28]Source: platform_hybrid/paper/sections/p6_query_auditor.tex.

scale, not an effect-size scale." Two boundaries deserve restating. The registry reports stated provenance and conflicts; it does not reconcile unresolved exact-ID or model conflicts, nor does it validate nominal algorithm comparisons[29]. And the badge is coarse relative to the corpus — twenty entries collapse to ten distinct fingerprints, and twenty of twenty-three sub-fields carry at most two values.

7.3 From Diagnostic to Controller (P7)

P7 asks a narrower question than P5 and P6. Given that the zero-variance fraction is worth reporting, is it mechanistically predictable, and is it ready to control rollout allocation? The paper's own framing is that this does not overturn the descriptive restraint of the earlier work — it *tests* ZVF's predictive value through falsifiable predictions rather than assuming it — and it flags at the outset that the weakest link in the chain, the causal connection from within-group contrast to gradient magnitude, rests on toy-scale directional evidence only[30].

The theory is compact. Fix a prompt with latent per-rollout success probability p and a group of G conditionally independent Bernoulli rewards. Write h_G(p) = p^G + (1−p)^G for the probability that the group is degenerate, and model the usable signal per prompt as S(p,G) = p(1−p)(1 − h_G(p)). The first factor is the contrast a non-degenerate group can carry; the second is the probability that any signal survives centring. Both factors vanish at the endpoints and peak at p = ½, so the reading is that GRPO learns fastest exactly where the policy is most uncertain and starves at both ends[31]. Three consequences follow, each paired with the measurement that could have falsified it. T1 states that the expected ZVF indicator is exactly h_G(p). T2 states that h_G is strictly decreasing in G, with exponential decay rate −ln max(p, 1−p). T3 states that with unnormalised group-centred advantages the mean absolute advantage in a group is exactly $2\hat{p}(1-\hat{p})$.

The three checks behave as the theory predicts, with the deviations reported rather than smoothed over. For T1, binning the per-group reward tensors of the GSM8K runs by empirical success rate gives a ZVF indicator averaging 1.000 at $\hat{p} \in \{0, 1\}$ and 0.000 at every interior bin; at run level, measured ZVF sits *above* h_G($\bar{p}$) at every G, which the paper attributes to the Jensen gap — the binomial model evaluates h_G at the mean p while the true prompt population mixes mastered and hopeless prompts. For T2, two independent grids agree. The 368-run audit shows monotone decline for every model with an interior success rate (Qwen3-32B

[29]Source: platform_hybrid/paper/sections/p6_abstract.tex.
[30]Source: platform_hybrid/paper/sections/p7_abstract.tex.
[31]Source: platform_hybrid/paper/sections/p7_theory.tex.

0.33 → 0.18 → 0.08; Llama-3.1-8B 0.76 → 0.65 → 0.47), with the two boundary rows behaving as the theory flags them: Llama-3.2-1B sits near $p \approx 0$ where $h_G \approx 1$ regardless of G (0.98 → 0.97), and Qwen3-4B-Instruct sits near saturation where the decay rate is tiny and seed noise dominates, producing the single non-monotone cell (0.89 → 0.84 → 0.91). The repository-internal sweep at $G \in \{2, 4, 8, 16\}$ shows the same decline, with mean ZVF falling from 0.838 to 0.631 while held-out accuracy stays essentially flat near 0.98 throughout — raising G buys signal, not, on this near-saturated task, outcome[32]. T3 cannot be checked on the closed stack at all, because it does not expose per-group gradients; the open-backward-pass measurement, $\mathrm{corr}(\|\nabla\|, \hat{p}(1-\hat{p})) = +0.71$ on a 0.5B model with synthetic arithmetic, is reported strictly as directional evidence.

Because ZVF aliases mastery with incapacity, the paper promotes the pairwise contrast density $\mathrm{PCD} = ((G-1)/G) \cdot E[p(1-p)]$ as the signal to control on. The falsification that motivates the promotion is the micro-jitter experiment already cited in §7.1: ZVF collapses from 0.158 to 0.000 while PCD is unmoved. The controller's design follows from the theory: a `zvf-triage` callback watches per-step ZVF with PCD as an aliasing guard, and an adaptive-G controller escalates group size on ZVF spikes, changing only the sampler budget so that it composes with any group-relative objective.

The four-arm open-trainer audit is the first place where the intervention is measured against alternatives on the same data, budget and seeds[33]. Against a 120-rollout baseline, GRPO gains +0.500 held-out at mean ZVF 0.25; Dr. GRPO gains +0.575 at 0.27 on the same 120 rollouts; DAPO-style dynamic sampling gains +0.550 while driving ZVF to 0.00 at 174 rollouts, a +45% cost; and adaptive-G gains +0.575 while lowering mean ZVF only modestly to 0.23 at 186 rollouts. Three readings are stated. Telemetry behaves as the mechanisms dictate. Dynamic sampling buys perfect telemetry at a 45% rollout premium and does not exceed the cheaper fixed recipe. Adaptive-G spends comparable overhead and ties the best held-out result, concentrating its spend on flagged steps — and the tie is small-n and single-task, so the honest description is "competitive with, not superior to, the best fixed recipe."

The controller simulation then asks a counterfactual question on the real reward tensors: for each of the 2,560 prompt-step decisions in the four-method run, when would each controller have fired, what group size would it have chosen, and how much contrast would it have restored? The answer is a negative result of some

[32]Source: platform_hybrid/paper/sections/p7_theory.tex, Tables `tab:p7-modelG` and `tab:p7-gsweep`.

[33]Source: platform_hybrid/paper/sections/p7_controller.tex, Table `tab:p7-e3`.

force. Across all four methods and all five thresholds of the point-estimate rule, the controllers save *zero* prompts. Of GRPO's 640 prompt-step pairs, 461 are truly degenerate — 427 all-correct and 34 all-wrong — with $\hat{p}$ at exactly 0 or 1, and the counts are near-identical for the other methods (AERO 461, GIFT 493, AREAL 452). At $\hat{p} \in \{0, 1\}$ no group size recovers contrast, because ZVF(G) = 1 for every G. The empirical ZVF of roughly 0.72 is high because the prompts are easy, not because they are noisy at the decision boundary. Bootstrapping the observed binary rewards to size G = 16 confirms it directly: 0 of 1,867 boundary prompts would have been restored by escalation. The operating range where the trigger fires selectively is $\tau \in [0.70, 0.80]$, 12 to 19 fires per 40 steps at a cost ratio of 1.30 to 1.48, and the different thresholds differ only in how wastefully they fire[34].

In the same trace the escalation asymmetry is stark: 1,723 of 1,867 escalation events (92.3%) occur on all-correct groups and only 144 on all-wrong groups[35]. A symmetric ZVF trigger mostly spends on solved prompts. The paper is explicit that deleting those fires is a counterfactual proposal only, because performance preservation has not been measured.

Two refinements sharpen the picture and then complicate it. The Bayesian controller replaces the point estimate with a Beta(1,1) posterior and a mid-range probability $m(k, G) = \Pr[0.05 \leq p \leq 0.95 \mid k, G]$, escalating when m exceeds a posterior threshold. At τ_post = 0.60 it fires on the same 466.75 degenerate prompt-steps as the point-estimate rule at $\tau = 0.50$ — with a 95% bootstrap CI of [454.25, 485.00] — but at strictly lower cost, 8,854 rollouts against 10,240, a 14% saving; and all 466.75 of those fires are on saturated prompts, so it too saves zero prompts. Above τ_post = 0.65 it never fires, because the maximum m over degenerate observations is 0.630[36].

The calibrated eight-controller comparison then produces a genuine metric reversal with bootstrap CIs attached. On the cost axis the Pareto winner is Dualformer-Auto, an automatic difficulty-gated de-escalation rule, at a mean cost ratio of 0.657 [0.599, 0.738] against 1.000 for fixed G = 8 — a 34% rollout saving, achieved because the prompts it shrinks to G = 2 would have been degenerate at G = 8 anyway. On a different axis, contrast restored per 1,000 extra rollouts, the ordering reverses: `zvf-triage` at $\tau = 0.50$ restores 90.10 units per 1,000 extra rollouts [89.40, 90.77], against 86.68 for $\tau = 0.70$, 84.00 for Dualformer-Auto and 80.00 for the Bayesian rule, and all four pairwise differences have CIs excluding zero. The

[34]Source: platform_hybrid/paper/sections/p7_controller.tex, Tables `tab:p7-cf-sweep` and `tab:p7-cf-granularity`.

[35]Source: platform_hybrid/paper/sections/p7_abstract.tex.

[36]Source: platform_hybrid/paper/sections/p7_controller.tex, §Bayesian Refinement.

Bayesian controller — Pareto-dominant on the earlier axis — is dominated here, because its principled prior fires on the hardest-to-restore degenerate prompts. The regime decomposition explains the mechanism rather than leaving it as a ranking: `zvf-triage` at $\tau = 0.50$ fires on 100% of steps in every regime on this panel, so it is operationally equivalent to "escalate every step," and its apparent advantage is mechanical exposure rather than a smarter rule. Its restoration decomposes 64.76% degenerate, 13.69% boundary and 21.56% mid-range; the non-degenerate 35.24% share is what the Bayesian rule sacrifices[37].

Two further analyses set the ceiling on what any adaptive rule can achieve. A per-prompt hindsight-optimal calculation scans G' down from a base of 8 and finds a bimodal optimum: G* = 8 for 72.9% of the 2,560 prompt-steps, which are fully saturated and admit no rescue by any group size, and G* = 2 for the remaining 27.1%, which are already non-degenerate at G = 8 and are cheapest to shrink. Mean G* is 6.376, giving a cost ratio of 0.797 [0.776, 0.818] — a 20.3% saving that is pure economy, since mean contrast restored per prompt is 0.000 and mean contrast lost to shrinking is 0.130 ZVF units. That ceiling is tighter than Dualformer-Auto's observed 0.66 to 0.68 on the four-bin rule, and the paper states plainly that any future learned rule must beat it. A finite-pool correction then shows that even this ceiling is loose: because a finite pool never collides more than independent sampling, the exact hypergeometric contrast-preservation probability exceeds the binomial one everywhere, by +0.0494 [+0.0481, +0.0506] on the 693 relevant prompt-steps, which means the binomial model over-predicts starvation and the "zero contrast restored" accounting understates true preserved contrast by roughly 34.2 ZVF units[38].

Cross-base replication is where the retreat from "controller" to "audit" is earned. On a second panel of five independent seeds, the trigger fires 4.20 times per seed [3.00, 5.40] at $\tau = 0.70$, with the per-step rate falling from 0.48 to 0.28 as the panel's overall ZVF drops; what replicates is the sign and the operating-point logic, not a fixed firing magnitude. The Bayesian branch is entirely silent on this panel, firing 0.0 times per seed, because every step's implied mid-range probability lies in [0.95, 0.999] — a real negative finding that scopes the Bayesian controller to distributions containing boundary steps. And the aggregate firing rate is stable while the *location* is not: at $\tau = 0.70$ with a de-escalation threshold of 0.95, the cross-seed Jaccard overlap of the fire sets is 0.133, only 13.3% of steps see all five seeds choose the same group size, and the mean cross-seed Kendall-τ of per-step ZVF trajectories is -0.060. The firing rate is seed-stable; the firing location is not,

[37]Source: platform_hybrid/paper/sections/p7_controller.tex, §Posterior-Predictive Contrast Restoration and §Statistically Validated Pareto Frontier.

[38]Source: platform_hybrid/paper/sections/p7_controller.tex, §Exact Finite-Pool Correction.

and the training-step index is a meaningless axis for the ZVF signal across seeds. The replayed unification confirms the direction: on this panel Dualformer-Auto is the only controller that saves compute (+14% [0.10, 0.18]) while `zvf-triage` spends more than baseline (−28% [−0.36, −0.20]), with disjoint CIs.

Five design rules close the paper, and they read as constraints discovered rather than principles asserted[39]. Gate on PCD and alarm on ZVF: escalate only when ZVF is high *and* PCD indicates interior difficulty, because high ZVF with near-zero PCD and high reward is mastery — where h_G(p) $\approx 1 - G\varepsilon$ near the boundary and added rollouts buy order ε — while high ZVF with near-zero reward is incapacity, where the correct intervention is the data, not the sampler. Escalate multiplicatively in small steps, since at p = ½ each additional rollout halves the degenerate fraction; the audited 4 → 6 → 8 schedule at multipliers of 1.5 and 1.33 captured most of the achievable reduction at a total cost of +55% rollouts, whereas G = 16 costs four times the rollouts of G = 4 for a ZVF drop from 0.76 to 0.63 and no held-out gain on a near-saturated task. Trigger on spikes rather than levels, because ZVF is temporally sticky with a lag-1 autocorrelation of approximately 0.94 and drifts upward over training, so a level threshold fires either never or permanently. Never compare controllers across stacks; ship the seven-field manifest with item 5 carrying the escalation log. And treat ZVF = 0 as a purchase rather than a prize — dynamic sampling shows perfect telemetry is always available at a 45% rollout premium, and on this task it was not the best use of the marginal rollout.

The chapter's honest summary of P7 is the paper's own. No result establishes prospective superiority over static G = 16, Dr. GRPO, or dynamic sampling; adaptive group size is explicitly *not* promoted to a recommended policy; and all intervention evidence is single-task, small-n, or retrospective. The paper's contribution is a retrospective controller audit together with a pre-registered prospective test plan, and the bakeoff that would close the question — a matched-budget, multi-seed held-out comparison against the strong static baseline — is future work rather than a claim. In the same spirit, the paper flags its own August 2026 evidence boundary: unsigned ZVF alone cannot select between retiring, distilling, retrying, resampling or warm-starting a prompt, so any operator policy remains a prospective evaluation contract rather than a demonstrated benefit.

[39]Source: platform_hybrid/paper/sections/p7_design_rules.tex.

Chapter 8

Results III: Applied Study (P8)

This chapter reports the applied side study of the programme: a head-to-head comparison between a tuned gradient-boosted tree and a fine-tuned large language model on credit-card fraud detection. The study is deliberately cross-domain. It steps outside the GRPO post-training work that occupies the rest of this thesis and asks whether the programme's governing discipline — measure capability, do not trust the label — survives contact with a setting that already has a strong non-LLM baseline. Where the benchmark chapters ask whether reinforcement learning changed a model in a measurable way, this chapter asks a blunter question: when a lender is told that an LLM can detect fraud, what is the LLM actually doing that the existing tree cannot? Applied here, the label under test is "AI" itself.

The short answer is that the tree keeps the real-time scoring seat, that the head-to-head which produced this conclusion is in any case the least interesting fact in the study, and that the LLM's real contribution lies in four task categories the tree cannot enter at all. Those categories are compliance narration, document and image fraud via vision-language models, cold-start triage of novel typologies, and agentic investigation of the alert queue. The chapter reports the measured comparison first, then the operational arguments that survive even at accuracy parity, then the measured evidence beyond the headline, then the capability-gap taxonomy and the hybrid architecture that follow from it. It closes with the limitations, stated plainly, because several of them bound what any number in this chapter may be used to claim.

A scope note is owed at the outset. The study is parked outside the ZVF thesis and supplies no evidence for GRPO signal starvation, group-size control, PPO, or Single-Rollout Asynchronous Optimization (SAO); it contributes an example of the programme's measurement discipline and nothing to the RL claims[1]. Its

[1]Source: platform_hybrid/paper/archive/absorbed/P08_fraud/sections/p8_intro.tex.

conclusions stand on the fraud artifacts and the cited domain literature alone.

8.1 Dataset, splits, and the two arms

The benchmark is a custom synthetic dataset generated with scikit-learn's `make_classification` (seed 42): 50,000 transactions, 20 anonymised numeric features V1–V20 of which 10 are informative and 2 redundant, two clusters per class, class separation 0.8, and a 1% label-noise setting (`flip_y=0.01`). The generator's 1% target positive rate realises as an actual fraud rate of 1.44% over the emitted rows. Four aggregates are appended to the twenty raw features — per-row mean, standard deviation, maximum, and minimum — giving a 24-feature design matrix[2]. The released dataset at the repository root carries exactly this schema: `V1` through `V20`, a `Class` label, and the four aggregate columns, across 50,000 rows[3].

The V-prefix convention mirrors the PCA-transformed public card-fraud datasets, but the data here is a stylised stand-in for one institution's post-processing feature view, and the study says so. It reproduces the class imbalance and the anonymised feature shape of single-institution card-fraud data and none of its temporal drift, verification latency, seasonality, or adversarial adaptation[4].

The split is 80/20 stratified with seed 42, giving 40,000 training rows and 10,000 held-out test rows containing 144 positives. The primary metric is held-out ROC-AUC; F1, precision, and recall at the default threshold are logged alongside.

The scorer arm is XGBoost with 200 estimators, maximum depth 6, learning rate 0.05, subsample and column-subsample of 0.8, and `scale_pos_weight=7` to counter the roughly 99:1 imbalance. No hyperparameter search was run beyond this single sensible configuration[5].

The challenger arm is a Qwen3.5-4B model fine-tuned on row-to-text serialisations in the TabLLM style: each transaction is rendered as a feature-name/value string and the model is trained to emit a fraud-or-legitimate answer. Because the natural-rate test split contains too few positives for a stable LLM AUC, evaluation uses a 500-row positive-enriched held-out subset at 20% fraud, disjoint from the training data[6]. The fine-tune uses token-level cross-entropy with weight zero on prompt tokens and weight one on answer tokens, with no reference policy and no KL term

[2]Source: platform_hybrid/paper/archive/absorbed/P08_fraud/sections/p8_setup.tex.
[3]Source: fraud_data.csv.
[4]Source: platform_hybrid/paper/archive/absorbed/P08_fraud/sections/p8_limitations.tex.
[5]Source: platform_hybrid/paper/archive/absorbed/P08_fraud/sections/p8_setup.tex.
[6]Source: platform_hybrid/experiments/results/quick_20260704/qp8-fraud-sft_manifest.json.

— this is pure supervised fine-tuning, not RL. The run completes in 63 steps at batch size 32 and takes 309.1 seconds to reach its final evaluation[7].

8.2 The head-to-head result

On the tabular scoring task the released artifacts place the two arms as follows.

Table 8.1: Head-to-head result on the synthetic card-fraud set, by arm and evaluation split.

Arm	Eval split	n	Accuracy	ROC-AUC
XGBoost (scorer)	full test	10,000	n/a	0.7955
Qwen3.5-4B SFT	positive-enriched test	500	0.7920	0.4827

[8]

The LLM's accuracy is not evidence of useful ranking. At this class balance an AUC of 0.4827 means the SFT scores do not order positives ahead of negatives: the model is at chance-level discrimination and its apparently reasonable accuracy comes from the majority class. This is consistent with the broader tabular literature, where tree ensembles remain the reference class on medium-sized tabular problems and LLM row-serialisation approaches are most attractive in few-shot regimes, not at 40,000 labelled examples[9].

Three provenance points must be stated with the table rather than buried beneath it. First, the two AUCs sit on different held-out splits and are not a like-for-like ranking comparison; the load-bearing point is only that the LLM scorer ranks at chance, not that it is worse than the tree by this exact margin[10]. Second, the precision, recall, and F1 figures belong to the released script's own output rather than to the quick artifact's 0.7955 run. That file logs AUC 0.7942 with precision 0.723, recall 0.236, and F1 0.356 at the default threshold, alongside a training time of 0.49 seconds and an inference time of 0.0062 seconds per 10,000 rows[11]. The paper's own setup section notes that this run uses a slightly different feature and library configuration from the quick artifact and that the two released artifacts agree to two decimals[12]. The precision-recall asymmetry — precision 0.723 against

[7]Source: platform_hybrid/experiments/results/quick_20260704/qp8-fraud-sft.tsv.
[8]Source: platform_hybrid/experiments/results/quick_20260704/qp8_fraud.tsv.
[9]Source: platform_hybrid/paper/archive/absorbed/P08_fraud/sections/p8_scorer.tex.
[10]Source: platform_hybrid/paper/archive/absorbed/P08_fraud/sections/p8_abstract.tex.
[11]Source: xgboost_results.json.
[12]Source: platform_hybrid/paper/archive/absorbed/P08_fraud/sections/p8_setup.tex.

recall 0.236 — is the class-imbalance signature at an untuned default threshold: the tree alerts on a small set of transactions it is mostly right about, and misses most fraud. It is a statement about the operating point, not about the model's ranking ability, which the 0.7955 AUC measures.

Third, an older internal record from 21 June 2026 listed XGBoost at AUC 0.975 and the LLM arm at 0.948. That environment could not be reconstructed, so those numbers survive only as historical records and are not headline evidence; the paper's architectural conclusions are insulated from them by design, since the seat assignment rests on arguments that apply at accuracy parity[13].

8.3 Why the seat does not move even at parity

The artifact-backed ranking is only the first of five reasons the tree keeps the real-time scoring seat. Even at AUC parity the LLM would not be moved into the synchronous path, for four operational reasons[14].

Latency is the first. Card authorisation is a hard-real-time loop whose score must arrive within a network-and-rules budget measured in tens of milliseconds. The committed run of the released script measures roughly 6 ms per 10,000 rows[15] — a rounding error against that budget. An LLM forward pass over a serialised row, let alone an autoregressive generation, is orders of magnitude slower and jitter-prone, and the batching that would amortise its cost is exactly what a per-authorisation synchronous path cannot use.

Cost is the second. The released cost table prices the XGBoost scorer at $1.00 per 10,000 rows and the Qwen3.5-4B SFT path at $0.0035 per row on 120 input and 5 output tokens. The synchronous LLM path is about 35 times more expensive than the tree-only path per transaction, and several orders of magnitude slower on latency — approximately 300 ms against approximately 6 ms[16]. The tree's marginal cost per transaction is effectively zero at card-network volumes; the LLM's is a token bill that scales linearly with traffic.

Calibration is the third, and it is the reason the argument does not reduce to cost. Downstream of the score sit threshold rules, alert budgets, and expected-loss computations that consume the score as a probability. Boosted trees produce dense scores that calibrate well under standard post-hoc maps; verbalised LLM confidences inherit the miscalibration pathologies of modern neural networks, com-

[13]Source: platform_hybrid/paper/archive/absorbed/P08_fraud/sections/p8_limitations.tex.
[14]Source: platform_hybrid/paper/archive/absorbed/P08_fraud/sections/p8_scorer.tex.
[15]Source: xgboost_results.json.
[16]Source: platform_hybrid/paper/archive/absorbed/P08_fraud/sections/p8_evidence.tex.

pounded by the quantisation of confidence into tokens. A scorer whose 0.9 does not mean ninety per cent silently corrupts every expected-loss decision behind it[17].

Prompt-injection exposure is the fourth, and it is qualitative but decisive. A tree scores numbers; there is no channel through which a transaction can address the model. An LLM that reads serialised transaction fields — merchant names, memo strings, cardholder-supplied text — creates an attack surface in which adversarial instructions are embedded in attacker-controlled fields, the indirect prompt-injection pattern documented in the security literature. A fraudster who learns that the scorer reads free text will put text in front of it. Placing an instruction-following model in a synchronous decision loop over attacker-authored inputs is a security regression independent of accuracy, and it is the single strongest argument for keeping the scorer seat non-linguistic[18].

The conclusion of this section is deliberately narrow: on fixed-schema tabular scoring with ample labels, the current artifacts score the LLM below the tree, and it would not get the seat even had it tied.

8.4 What the measured evidence adds

Beyond the headline pair, the released dataset supports a set of checks that sharpen the negative result and bound its interpretation.

The first is an empirical surrogate for the LLM-as-sensor idea. A tree fitted on only the four aggregate columns — a stand-in for what an LLM sensor returns — is worse than the 24-feature tree on every measured axis at 95% confidence under a paired bootstrap: AUC lower by 0.0245 with interval [0.0174, 0.0320], accuracy lower by 0.0135 with interval [0.0109, 0.0160], and Brier score worse by 0.0109 with interval [0.0099, 0.0120][19]. Adding the four engineered aggregates to the twenty raw features does not move AUC or accuracy outside paired-bootstrap noise: the AUC difference is 0.0002 with interval [−0.0002, 0.0007], which contains zero[20]. The sensor is complementary information, not a substitute for the raw features.

The second is a leakage caveat that must travel with any tree number quoted from the wider evidence section. Several tables in that section report AUC between 0.998 and 0.999 for tree variants on the 10,000-row split with 144 positives. Those figures are not absolute operating numbers. The calibration and robustness pipeline that produces them fits on the full `fraud_data.csv` and scores

[17]Source: platform_hybrid/paper/archive/absorbed/P08_fraud/sections/p8_scorer.tex.
[18]Source: platform_hybrid/paper/archive/absorbed/P08_fraud/sections/p8_scorer.tex.
[19]Source: platform_hybrid/experiments/results/p5p8/p8_headline_cis.tsv.
[20]Source: platform_hybrid/experiments/results/p5p8/p8_headline_cis.tsv.

`test_data.csv`, whose rows are a subset of `fraud_data.csv`; training therefore overlaps evaluation, which inflates the absolute AUC towards 1.0 and drives top-2% caught-recall to 1.000. The leakage-free release split reproduces the headline 0.7955 with top-2% caught-recall of approximately 0.40, and that is the tree's reproducible absolute performance. The 0.998–0.999 tables are within-protocol relative comparisons only[21]. Quoting 0.9988 as the tree's accuracy would be a straightforward misreading of the evidence.

The third is a noise budget for any deployed sensor. In the oracle frame, where the sensor emits the exact aggregates, the sensor-augmented tree breaks even at a missed-fraud loss of about \$5.29. At the certified sensor-noise budget of $\sigma \leq 0.02$, the break-even rises to \$26.44, and at $\sigma = 0.05$ it reaches \$36.40. The oracle break-even is a five-fold under-estimate of the realistic break-even a production LLM sensor would face, and the paper states this as the sharpest reviewer-facing claim of the noisy-sensor frame[22]. Notably, the qualitative cost signature is unchanged by noise: the sensor-augmented tree remains detectably costlier at low fraud loss, and confidence intervals straddle zero elsewhere.

The fourth is cost per fraud caught at a canonical cost cell. At an investigation cost of \$0.50 per alert and a missed-fraud loss of \$100, raw-features-only costs \$2.68 per fraud caught, the sensor-augmented tree \$3.28, and the sensor-only surrogate \$14.92 — a 5.57× penalty whose direction is consistent across all 25 cells of the phase diagram. In the (σ, L) plane the sensor never pays for itself in any of the 25 cells[23].

The fifth is threshold transfer. A cost-optimal threshold fitted on training data is a fingerprint of the training prior, not a portable decision rule: under class-prior shift the transfer gap is detectable for all three tree variants at the $\rho = 500$ cell, at +6.55, +6.67, and +2.19 cents per decision respectively[24]. Any deployment that carries a fitted threshold across a base-rate change inherits this penalty, LLM or not.

[21]Source: platform_hybrid/paper/archive/absorbed/P08_fraud/sections/p8_evidence.tex.
[22]Source: platform_hybrid/paper/archive/absorbed/P08_fraud/sections/p8_noisy_sensor.tex.
[23]Source: platform_hybrid/paper/archive/absorbed/P08_fraud/sections/p8_evidence.tex.
[24]Source: platform_hybrid/paper/archive/absorbed/P08_fraud/sections/p8_evidence.tex.

8.5 Where the LLM does contribute: four capability gaps

Framing the study as "LLM versus XGBoost" was a category error, and the error is instructive. The comparison was conducted on the tree's home ground — fixed-schema tabular scoring with ample labels — and the LLM lost there. The useful question is the inverse: on which tasks does the tree not compete at all? The study identifies four, each of which is a task category the tree cannot enter, not an accuracy improvement on a task the tree already performs[25].

The first is document and image fraud: forged receipts, tampered identity documents, synthetic cheques, and manipulated screenshots. These inputs lie outside any tree's feature space by construction. The claim is not that a vision-language model is accurate enough to auto-decide, but that it is the only model family in the pipeline that can look at the evidence at all; in the proposed architecture its output is a feature vector and a written observation, not a verdict.

The second is compliance narration, and the study argues it is the paper's most durable claim because it survives any future accuracy result. A tree emits a float; it cannot enter this category at any accuracy. Suspicious Activity Report narratives are governed by 31 C.F.R. § 1020.320, which sets a $5,000 threshold and a 30-day filing window, and by FinCEN's narrative guidance requiring who, what, when, where, and why alongside the method of operation. Adverse-action statements under ECOA and Regulation B, 12 C.F.R. § 1002.9, must state the principal reasons for the action specifically, and CFPB Circular 2022-03 states that creditors may not rely on black-box complexity as a defence[26]. Two design consequences follow: the scorer must remain attribution-extractable, which makes keeping XGBoost in the scorer seat a compliance posture and not only a performance one; and the scribe must narrate the scorer's actual attributions rather than invent them.

The third is cold-start triage of novel typologies, before labels exist. This is a few-shot regime, and it is precisely where row-serialisation LLM approaches are most attractive. The claim is not that the LLM is a better classifier but that it is a stopgap classifier that exists during the label vacuum[27].

The fourth is agentic investigation of the alert queue, at roughly 85 times lower cost than a human analyst by the study's internal estimate — a figure reported to one significant figure, built on June-2026 token prices and assumed investigation times, and not validated against any production deployment. It should be read

[25]Source: platform_hybrid/paper/archive/absorbed/P08_fraud/sections/p8_taxonomy.tex.
[26]Source: platform_hybrid/paper/archive/absorbed/P08_fraud/sections/p8_compliance.tex.
[27]Source: platform_hybrid/paper/archive/absorbed/P08_fraud/sections/p8_taxonomy.tex.

as "one to two orders of magnitude," not as a measured constant. Crucially, this role is post-score: the agent works the queue after the tree has ranked it[28].

8.6 The hybrid architecture

These findings distil into a three-seat architecture. The LLM is the sensor: it reads unstructured evidence and emits typed features — document-consistency scores, extracted field values, tamper-likelihood indicators, mismatch flags — across a fixed numeric schema, with no tools, no memory across cases, and no authority. XGBoost remains the scorer: nothing in the real-time loop parses natural language, which preserves both the tree's calibration and the containment of the injection surface to per-case feature corruption. The LLM is also the scribe: it drafts Suspicious Activity Report narratives and adverse-action statements conditioned on the tree's own feature attributions, so that the stated reasons are the model's actual principal reasons, with provenance links on every draft and a human signature on every filing. An agentic triage stage sits between scoring and writing, working the alert queue asynchronously under a budget cap, so that token spend lands only on the small fraction of traffic that alerts. Analyst dispositions flow back as confirmed labels, closing the loop with the scorer[29].

8.7 Limitations

Six limitations bound what this chapter may be used to claim, and they are not incidental[30].

The benchmark is a custom synthetic dataset, not production transactions. It reproduces class imbalance and anonymised feature shape and none of the temporal drift, verification latency, seasonality, or adversarial adaptation of real card-fraud data. The artifact-backed AUCs of 0.7955 and 0.48268 are results on a stylised testbed and must not be quoted as production expectations. No number in the study should be read as an expected production AUC, and the margin between the arms is dataset- and environment-specific.

The historical numbers are records, not reproductions, as discussed above. There is no cross-institution validation: everything reported here is one configuration of one synthetic generator standing in for one institution's feature view, and the head-to-

[28]Source: platform_hybrid/paper/archive/absorbed/P08_fraud/sections/p8_taxonomy.tex.
[29]Source: platform_hybrid/paper/archive/absorbed/P08_fraud/sections/p8_architecture.tex.
[30]Source: platform_hybrid/paper/archive/absorbed/P08_fraud/sections/p8_limitations.tex.

head has not been replicated on any second data distribution, public benchmark, or real institution. The cost estimates are internal; the 85× agentic-triage figure, like the latency and per-transaction cost contrasts, is an operational characterisation rather than a benchmarked service-level measurement. The challenger is a single LLM family under a single serialisation format and a single fine-tuning budget; stronger or differently trained models could narrow the accuracy gap, though the seat assignment does not hinge on the gap because the latency, cost, calibration, and injection arguments apply at parity. Finally, the capability gaps are argued rather than all measured: of the four taxonomy entries, only the scorer head-to-head carries a number of the study's own. The sensor, scribe, and cold-start seats are grounded in external literature and primary regulatory text, the hybrid architecture is a design contribution whose integrated performance is future work, and nothing in the compliance discussion is legal advice.

8.8 Relation to the thesis

The fraud study belongs in this report as a boundary test of the programme's central discipline rather than as a result about RL. It contributes one worked example of the programme's habit of measuring a claimed capability instead of accepting its label: the label happened to be "AI in fraud detection," the measurement happened to return a negative, and the negative is reported as found. Its scientific claims stand or fall on the fraud artifacts and the cited domain literature alone, and it supplies no evidence for the GRPO, PPO, or SAO results in the surrounding chapters[31].

[31]Source: platform_hybrid/paper/archive/absorbed/P08_fraud/sections/p8_intro.tex.

Chapter 9

The E1–E14 Held-Out Evaluation Campaign

This chapter reports the campaign that occupied the second half of the project: the evaluation of a single Tinker-trained actor against fourteen held-out benchmark suites, recorded lane by lane under a governance regime of sealed reservations, immutable receipts and explicit terminal states. The campaign is reported here in full, including the lanes that did not finish and the lanes that could not start, because the central methodological claim of this work is not that the actor performs well but that every figure attached to it can be traced to a surviving, hash-pinned artifact, and that the boundary between what was measured and what was not is drawn explicitly rather than rhetorically (Figure 9.1).

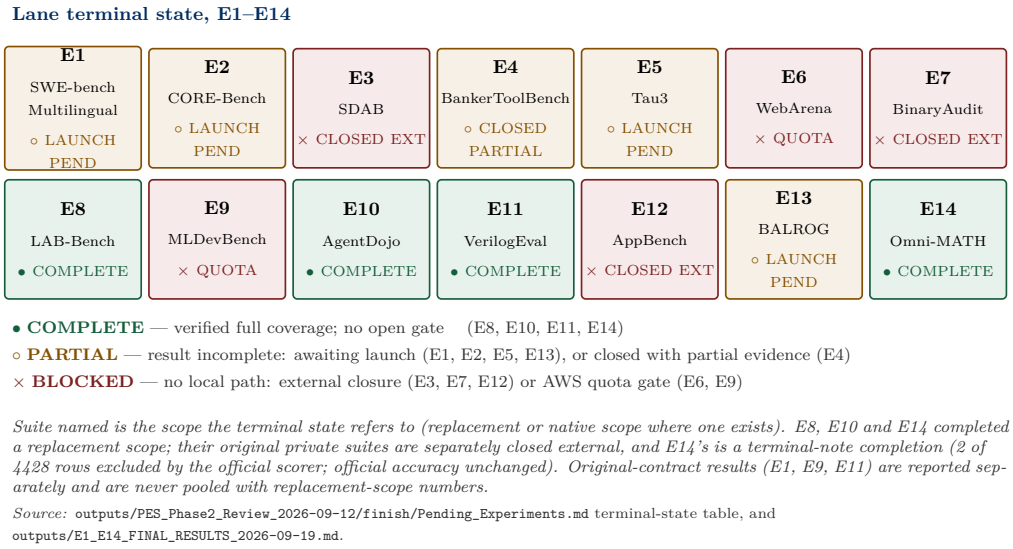


Figure 9.1: Terminal status of all fourteen E1–E14 lanes. Each cell names the suite and its state, so status is legible without relying on colour alone.

Two reporting rules govern everything below, and they are worth stating before any number appears. First, original-contract results and replacement-scope results are never pooled and never averaged across suites; a replacement scope is a different suite, a different interface and often a different grader, and combining such figures would produce a number with no referent. Second, no figure in this chapter is quoted to a precision beyond what its source receipt records, and no comparison against a baseline, a sibling suite, or the untrained base model is asserted anywhere: the campaign contains no controlled paired baseline run, and this chapter therefore claims no improvement of any kind over the base model. Where a lane's record itself contains an unresolved discrepancy, the discrepancy is stated.

The consolidated ledger of record for the campaign is `outputs/E1_E14_FINAL_RESULTS_2026-09-19.md`, which supersedes the working tables of 2026-09-05 and 2026-09-12; the per-lane terminal-state ledger is `outputs/PES_Phase2_Review_2026-09-12/finish/Pending_Experiments.md`; and the arithmetic of the ledger was re-verified by deterministic code checks recorded in `outputs/verification/LEDGER_CODE_CHECK_2026-09-19.json`, which reports eleven of eleven checks passing with zero failures.

9.1 The evaluated actor and the one-actor design

A single actor was evaluated across the whole campaign. The base model is `Qwen/Qwen3.6-35B-A3B` at revision `995ad96eacd98c81ed38be0c5b274b04031597b0`, and the trained parameter set is the LoRA adapter `arvindcr4/pavlov-portfolio-qwen36-seed809-stepfinal-tinker-cf0ad8c1-1f1b-5ff-9f777c4018b6` at commit `64444133c55d88c3f1bf0df8a2f5d7ac646125c8`, served in bfloat16 under the identifier `pavlov-public-portfolio-bf16`[1]. A reproducibility note follows from this identity. The Tinker training run behind the adapter has since been purged, so the actor can no longer be sampled on Tinker; the adapter itself is preserved on the Hugging Face branch `checkpoint-seed809-stepfinal-9f777c4018b6` at the recorded commit, and a check on 26 September 2026 confirmed the 2.25 GB weight file is present there, so the actor can be re-served off-Tinker[2].

[1]Source: `outputs/E1_E14_FINAL_RESULTS_2026-09-19.md`; the same identity pair is recorded independently in the E10 native receipt and in the E11 full receipt, which also preserves the training sampler path `tinker://cf0ad8c1-1f1b-5ff3-8bd7-2a0bf232657b:train:0/sampler_weights/seed809_final`: sources `outputs/public_portfolio_2026-09-05/agentdojo_native_receipt.json` and `outputs/modal_e1_e14/2026-08-16/e11_full_receipt.json`.

[2]Sources: `outputs/TINKER_SAMPLER_STATUS_2026-09-20.json`, `outputs/e1_e14_small_scale_2026-09-26/ADAPTER_AVAILABILITY_CHECK_2026-09-26.json`.

The one-actor design is deliberate and it shapes how the results may be read. No lane received a separately fine-tuned checkpoint, no lane received per-suite prompt engineering beyond what the suite's own native harness prescribes, and no lane was tuned against its own evaluation set. A consequence is that differences between suites in this chapter reflect the interaction of a single fixed policy with fourteen different interfaces, observation spaces, tool conventions and graders, and they are not evidence about what a specialised actor could achieve. A second consequence is that the campaign cannot support any statement about seed variance: every suite number reported here comes from one actor, and where a lane sampled more than one completion per task it did so to satisfy the suite's own protocol, not to estimate variance.

Inference conditions were frozen per lane and recorded in that lane's own receipt rather than imposed globally, and the two lanes that produced complete scores illustrate how differently they were pinned. The E10 native run operated under an identity fixed at temperature 0, a 32,768-token context, 1,967 native requests, 4,096 output tokens per request, 3,600 seconds of wall time and an $8 envelope, and its collector is written explicitly not to admit shorter native conditions[3]. The E11 run was sampled at temperature 0.2, top-p 0.95, 1,024 maximum output tokens and seed 1816, one sample per problem across all 312 prompts[4]. Both are correct for their respective suites; the point of recording them per lane is that the sampling regime is part of the result, not a detail beneath it.

Two scoping notes complete the picture of what the actor has actually been measured on. The E1 original-contract evaluation was a one-checkpoint evaluation whose 731 generations were produced by a mixed backend within a single run — 476 Tinker generations and 255 Modal vLLM generations — and that backend mix is recorded in the original receipt[5]. Separately, two small probes exist at other parameter settings but are not suite results and are not reported as such: a base-model smoke run of the E11 harness measured zero scored tasks, and a four-task pilot at an intermediate (step-40) checkpoint returned four passes out of four with zero extraction failures while carrying an unresolved outstanding blocker — the decontamination receipt identifier for that pilot does not exist, and the split-manifest validator still fails on it[6].

[3]Source: `outputs/public_portfolio_2026-09-05/agentdojo_terminal_readiness.md`.

[4]Source: `outputs/modal_e1_e14/2026-08-16/e11_full_receipt.json`.

[5]Source: `outputs/e1_e14_results_2026-09-05/E1_E14_Results.md`.

[6]Sources: `outputs/e11_verilog_eval/e11_smoke_receipt.json`, `outputs/e11_verilog_eval/e11_trained_step40_receipt.json`.

9.2 Suites with strictly complete scopes

Three suites reached strictly complete replacement scopes, and the ledger's own integrity check asserts exactly this set: E8, E10 and E11[7]. A fourth, E14, closed terminal-complete under its native protocol with a recorded note. Each is described below with its published figures and, where the receipt imposes one, its claim boundary (Figures 9.2 and 9.3).

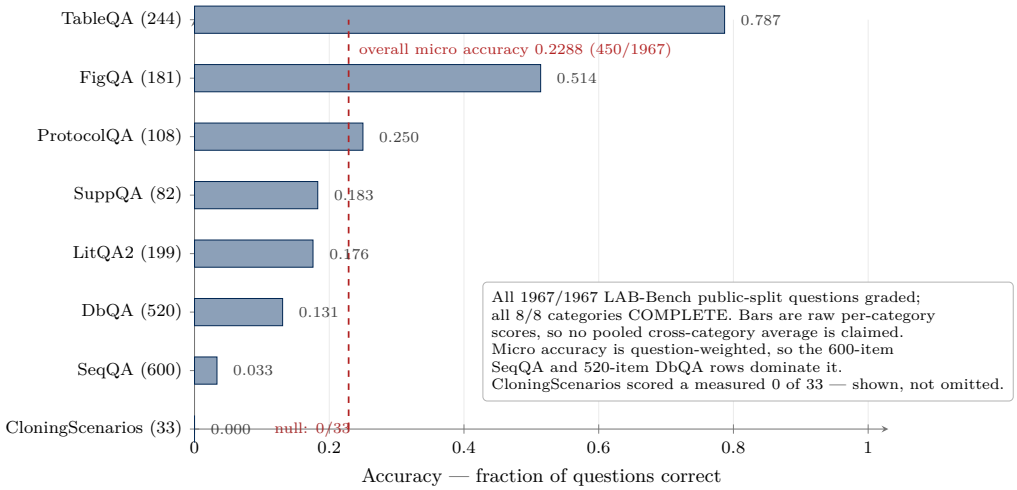


Figure 9.2: LAB-Bench public split, per-category accuracy, all 1,967 questions graded across eight categories. Accuracy is strongly category-dependent; no pooled cross-category average is claimed.

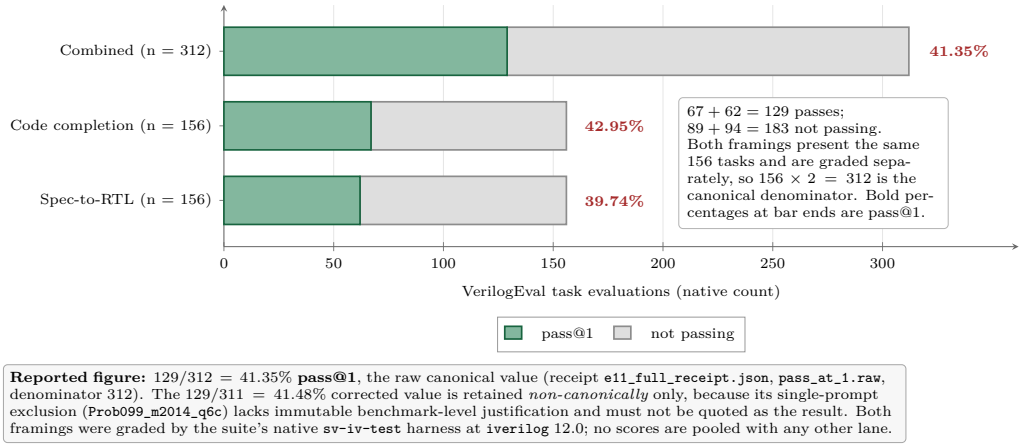


Figure 9.3: VerilogEval decomposition: two native framings of 156 tasks, 67/156 code-completion and 62/156 spec-to-RTL, summing to 129/312 = 41.35% pass@1.

E8 — LAB-Bench, public split. The evaluated scope is the full public split: 1,967 of 1,967 expected items were evaluated across eight categories, with no cat-

[7]Source: `outputs/verification/LEDGER_CODE_CHECK_2026-09-19.json`, check `strict_completed_suites_are_E8_E10_E11`.

egory incomplete. The receipt records 450 correct and 1,517 incorrect responses, giving a published overall accuracy of 450/1967 = 0.22877478393492628. Per-category accuracy, correct counts and coverage are: TableQA 0.7868852459016393 (192 correct of 244); FigQA 0.5138121546961326 (93 of 181); ProtocolQA 0.25 (27 of 108); SuppQA 0.18292682926829268 (15 of 82); LitQA2 0.17587939698492464 (35 of 199); DbQA 0.13076923076923078 (68 of 520); SeqQA 0.03333333333333333 (20 of 600); and CloningScenarios 0.0 (0 of 33). Coverage per category is at or above 0.9874 everywhere, and the CloningScenarios coverage is exactly 1.0 with zero correct. The receipt also records zero parser errors, 1,960 responses marked "sure" against 7 marked "unsure", and a completion profile of 1,086 length-truncated finishes against 881 stop finishes. One further field bears directly on how the accuracy should be read: `native_answer_parse_none` is 1,259, meaning that for 1,259 of the 1,967 responses the native pipeline extracted no answer at all. Because the receipt's correct and incorrect fields partition all 1,967 rows, those unparsed responses are counted as incorrect in the published accuracy, and the figure is therefore an end-to-end harness accuracy rather than an accuracy over the subset of responses that yielded a parseable answer[8]. The distinction matters for interpretation and is recorded here rather than averaged away.

E10 — AgentDojo, benign utility scope. The E10 lane reports the benign task utility of the actor over the frozen AgentDojo v1.2.2 benchmark at commit `089ed468cf3ed0322acc66b0211f26d9d90dbf60`. The terminal native receipt records status `COMPLETE_NATIVE_BENIGN_UTILITY`, all 97 episodes completed, 88 utility passes, a score denominator of 97, and therefore a score of 0.9072164948453608; `missing_evaluation_ids` is empty and `native_handled_error_count` is zero. The 97 episodes decompose as banking 16, slack 21, travel 20 and workspace 40. The receipt carries the run manifest hash `00e4228f4b757dc32b69b74f05a88348151f13185b470c5f00e31fdaa830ef30` and fixes the protocol identity as `benign_task_utility` v1.2.2 with attack, defence and security-evaluation fields all null and `heldout_claim` false. Its claim boundary is stated verbatim in the receipt as "Native default benign task utility only;no prompt-injection security or held-out claim", and its decontamination status is `TRAINING_INVENTORY_ABSENT`[9]. Two honest qualifications belong in this record. The first is that completion and utility are distinct quantities: 97 of 97 episodes completed, and 88 of those 97 passed their utility predicates, so the lane should be described as complete in coverage and 0.9072 in benign utility, never as "97/97 correct". The second is that this scope is not a security result of any kind, which the receipt itself insists upon.

[8]Source: `outputs/public_portfolio_2026-09-05/labbench_final_diagnostics.json`.
[9]Source: `outputs/public_portfolio_2026-09-05/agentdojo_native_receipt.json`.

Two pieces of engineering context are inseparable from the E10 number and are reported with it. An earlier attempt terminated with 94 of 97 episodes and score null, missing workspace tasks 37 through 39[10]. The cause was a confirmed collector fault rather than a model failure: verified task receipts recorded absolute `/cache/...` trace paths, and the frozen summary recomputed those paths on the local host, so every completed receipt failed comparison after extraction. The repair was deliberately minimal — the original archive and every extracted receipt and trace byte are unchanged, and a narrowly scoped adapter rewrites only the freshly computed `task_result.native_trace_path` to the verified canonical remote path before the frozen summary compares it, leaving native task code, environment, prompt, result parsing, utility values and handled-error policy untouched. The revised collector passed 18 offline tests, independently re-run in 0.507 seconds, and the continuation run that closed episodes 37 through 39 was executed under reservation `public-20260905-agentdojo-continuation01` with the native wrapper hash `11a8b61631390d9b20decd4263008e5a4c6f648a033d314384bac9e5d2456d77`[11].

E11 — VerilogEval, two native framings. E11 is the campaign's only complete suite that produced both full coverage and a native model score. All 312 pinned problems were evaluated — 156 code-completion (`code-complete-iccad2023`) and 156 specification-to-RTL — under the upstream NVlabs/verilog-eval harness at revision `c498220d0a52248f8e3fdffe279075215bde2da6`, using Icarus Verilog 12.0 at revision `4fd5291632232fbe1ba49b2c26bb6b2bf1c6c9cf` and Verilator 5.050. The result is 129 passes of 312, or 0.41346153846153844, decomposed into $67/156 = 0.42948717948717946$ on code-completion and $62/156 = 0.3974358974358974$ on specification-to-RTL. Pass verdicts were determined deterministically: for each pinned problem the single `sample01 sv-iv-test` log was read and marked PASS only when it contained a full-line "Mismatches: 0 in samples" verdict, with missing or non-matching logs marked FAIL. The upstream `sv-iv-analyze` summariser could not be used because it fails on a missing optional langchain dependency; that failure is recorded in the receipt and does not affect the aggregation, which reads the simulator logs directly. The receipt also records 150 extraction failures across the run, meaning candidate code blocks that could not be extracted from the model response; these appear as FAIL verdicts in the denominators above rather than being dropped[12].

[10] Source: `outputs/public_portfolio_2026-09-05/agentdojo_score_snapshot_fast02_terminal_v3.json`.

[11] Sources: `outputs/public_portfolio_2026-09-05/agentdojo_terminal_readiness.md`, `outputs/public_portfolio_2026-09-05/agentdojo_continuation01_result.json`, `outputs/public_portfolio_2026-09-05/agentdojo_continuation01_terminal_evidence.json`.

[12] Source: `outputs/modal_e1_e14/2026-08-16/e11_full_receipt.json`.

The E11 lane also carries the campaign's clearest example of a scoring decision taken for evidential rather than numerical reasons, and it is worth setting out because it shows the governance working. The all-problems reference sweep — running the shipped reference implementation of every problem through both simulators — returned 311 of 312. One problem, `verilog_eval/spec-to-rtl/Prob099_m2014_q6c`, fails to elaborate against its own test bench under both simulators, with Icarus reporting that ports `Y2` and `Y4` are not ports of `good1`; no candidate implementation can pass that task, and the upstream defect biases pass@k on the specification-to-RTL split downwards by roughly 0.64 percentage points. Excluding it yields 129/311 = 0.41479099678456594. That corrected figure is *not* canonical. The recorded amendment of 2026-08-22 states the reason plainly: no immutable benchmark-level justification was captured for the exclusion, so the raw 129/312 result is canonical and the 129/311 value is retained only as a noncanonical sensitivity; no samples and no verifier artifacts were changed by the amendment[13]. The distinction is preserved here in exactly those terms, and the ledger's integrity check confirms that the component split sums correctly to the canonical total[14].

E14 — Omni-MATH replacement scope, terminal-complete. The E14 replacement scope accepted 4,426 of 4,428 rows, or 99.95 per cent, and reproduced an official accuracy of 2,271/4,428 = 0.5131043831902395. Two rows were not accepted: `omni-01195-a6cbdc4ee3c8694b` at row index 1,195 and `omni-02044-612ddacaa39d91fe` at row index 2,044. The terminal note of 2026-09-19 resolves them decisively and in a direction that does not flatter the lane: the Omni-Judge outputs for both rows are truncated before any "## Equivalence Judgement" section — row 1,195 ends mid numeric answer and row 2,044 ends mid sum expression — so the judge emitted no verdict at all, and the native scorer's omission of rows lacking an equivalence judgement is correct official behaviour rather than a parser defect. The score impact is nil, because the official accuracy already counts both rows in the denominator as not correct. The lane therefore closes as terminal-complete under the native protocol with 4,428 of 4,428 dispositions recorded (4,426 accepted plus 2 recorded skip reasons), and no re-judge was performed because re-judging would deviate from the official scorer and constitute new evidence generation with no score effect[15].

[13] Sources: `outputs/modal_e1_e14/2026-08-16/e11_full_receipt.json`, `outputs/e11_verilog_eval/e11_verilog_eval_rerun_receipt.json`, `outputs/e11_verilog_eval/lane_status_2026-08-09.md`.

[14] Source: `outputs/verification/LEDGER_CODE_CHECK_2026-09-19.json`, check `E11_component_split_sums_67_plus_62_equals_129`.

[15] Source: `outputs/PES_Phase2_Review_2026-09-12/finish/e14_terminal_note_2026-09-19.json`.

9.3 Original-contract results

Where a lane's original contract could be honoured on the original suite, the result is reported separately from any replacement scope, and the coverage caveats travel with it. Seven lanes have an original-contract figure; two of them are full-suite results.

E1 — SWE-bench Pro. The actor resolved 2 of 731 tasks, a pass@1 of 2/731 = 0.0027359781121751026, or 0.274 per cent. The denominator is the complete 731-row test split of `ScaleAI/SWE-bench_Pro` at revision `7ab5114912baf22bb098818e604c02fe7ad2c11f`. Within that denominator the run produced 713 native evaluations, fourteen generation failures and four lost generation artifacts; those eighteen non-native outcomes are retained in the denominator rather than trimmed, so the figure is a rate over all 731 tasks and not over the subset that graded cleanly. The declared licence for the dataset is null in the lane's own receipt. The evaluation harness is `scaleapi/SWE-bench_Pro-os` at commit `ca10a60a5fcae51e6948ffe1485d4153d421e6c5`, and the lane records a known harness-fidelity caveat: the gold-resolve timeout was set to 1,200 seconds where the official setting is 120, because of an arm64 QEMU emulation constraint in the local execution environment[16].

E2 — FrontierSWE. One of 17 tasks was evaluated, with a replay-normalised score of 0.8628. The lane is partial: revision-bound authorisation for the remaining sixteen tasks was never obtained, and the original Tinker checkpoint was unavailable[17].

E4 — BankerToolBench. One of 100 tasks was evaluated, with a recovery metric of 0.3115. That figure requires care and is stated here in the lane's own terms: it is a verifier recovery metric on a single task, not the simple fraction 37/128 and not a suite score. The lane's public artifacts were restored and 1,175 manifest checks pass, but the pass3/pass16 archive is genuinely missing — zero of six artifacts verify, and only `TINKER_BRIDGE.json` does[18].

E5 — APEX-Agents. Seven of 480 tasks were natively scored. The lane's headline aggregate, a prefix mean of 0.050505, is an attempt summary in which unscored attempts are treated as zero, and it is explicitly not a native 480-task score; the seven tasks that were actually scored average 0.079365[19].

[16]Sources: `outputs/e1_swe_bench_pro/lane_receipt_2026-08-09.json`, `outputs/e1_swe_bench_pro/preflight_2026-08-09.json`, `outputs/e1_e14_results_2026-09-05/E1_E14_Results.md`.

[17]Source: `outputs/e1_e14_results_2026-09-05/E1_E14_Results.md`.

[18]Sources: `outputs/e1_e14_results_2026-09-05/E1_E14_Results.md`, `outputs/PES_Phase2_Review_2026-09-12/finish/Pending_Experiments.md`.

[19]Source: `outputs/e1_e14_results_2026-09-05/E1_E14_Results.md`.

E7 — BinaryAudit. One of 46 attempts is recorded, with a verifier reward of 0.0 following an agent error, and no grade. The terminal ledger of record gives the denominator as 46 attempted; an earlier working snapshot, since superseded, recorded the same lane as "1/28 errored". The two snapshots differ in the scope of what they counted as an attempt, and this chapter reports the terminal figure while flagging that the superseded snapshot's narrower count exists in the record[20].

E9 — MLE-bench. Forty of 75 competitions carry native grades, a coverage of 53.33 per cent, and the suite score is null: coverage without complete grading is not a score, and the lane records it as such. Behind the coverage number sit 97 legacy receipts across 50 competition identifiers, of which 41 graded receipts cover 40 unique competitions; the ledger's integrity check verifies precisely that decomposition. A separate merged-vLLM arm holds seven receipts, of which exactly one is a valid grade (H&M, 0.02132) and six are invalid submissions, and the integrity check confirms that arm is recorded separately with one graded row. The two arms are not combined, and one old Spooky Author submission is absent locally although the later run's complete submission is retained[21].

E11 — VerilogEval is reported in Section 9.2, where it belongs as a strictly complete suite rather than a partial original-contract lane; it is the second of the campaign's two full-suite results, alongside E1.

Nothing in this section is compared with anything else. No lane here has a matched base-model run, and the presence of a figure for the trained actor does not by itself license a claim that the actor outperforms its untrained counterpart, a predecessor model, or a published number.

9.4 Partial and quota-blocked lanes

Beyond the original-contract partials, six lanes have replacement scopes that are started, measured in part, and honestly incomplete. They fall into two groups: lanes whose remaining work is a sealed and locally validated launch that has not been executed, and lanes whose remaining work is gated on cloud quota that has not been granted.

In the first group, **E1's SWE-bench Multilingual replacement** has 35 of 300 tasks graded. A wave-10 recovery of sixteen additional tasks at $16 was sealed

[20]Sources: `outputs/PES_Phase2_Review_2026-09-12/finish/Pending_Experiments.md`, `outputs/E1_E14_FINAL_RESULTS_2026-09-19.md`.

[21]Sources: `outputs/E1_E14_FINAL_RESULTS_2026-09-19.md`, `outputs/verification/LEDGER_CODE_CHECK_2026-09-19.json`, `outputs/e1_e14_results_2026-09-05/E1_E14_Results.md`.

and validated locally but its execution source was lost when the finish-era working directory was deleted, and under the lost-source policy it is being faithfully re-implemented in a tracked path; the rebuild at `zvf-program/e1_wave10/` passed 10 of 10 targeted regression checks offline, and the $16 launch remains pending a fresh lead seal, a live actor endpoint and an online W&B run. **E2's CORE-Bench replacement** has all 45 capsule setups complete and zero tasks graded; a direct-VM adapter was built and the orchestration driver passes 12 of 12 offline tests, with the remaining blocker recorded as a hard owner-level GCP IAM binding rather than any missing code. **E5's Tau3 replacement** has 20 of 97 tasks cleaned, a coverage of 20.62 per cent; the successor-27 chain was re-verified offline and reached 52 of 52 tests passing after a duplicate-keyword fault was repaired in the dispatch path and three test-side wiring faults were fixed, with the controller's safety properties unchanged. Its $80 admission launch is held on two counts: no lead-issued bound authorisation receipt exists, and the lane record also finds no real native runtime adapter in the repository for such a receipt to bind, so the lane cannot launch even with the receipt in hand. **E13's BALROG replacement** has 13 of 255 episodes recorded, with 21 never-started receipt candidates mapped, and its hosted-supervisor chain verified 17 of 17 offline; the $8 actor launch awaits allocation of a reservation, a connectivity re-proof, a route-gate change and a supervisor deployment[22].

In the second group, **E6's WebArena replacement** stands at 0 of 812 tasks. Its canonical inventory has been verified as 65 files with ordered task identifiers 0 through 811, so the suite is staged and the absence of results is purely a compute-access question: AWS us-east-2 quota is at 1 of 16 vCPU with case `c67d89b0` open since 12 September, and a one-open-request rule prevents a second filing. **E9's MLDevBench replacement** stands at 0 of 34 tasks graded with an incomplete runtime image; us-east-1 quota is 1 of 8 vCPU with request `2eebd37d` filed on 19 September and case `178982528000009` opened, and a local Docker build was deferred at 31 GiB free disk against a builder requirement of at least 30 GiB plus recipe reconstruction. Both lanes were re-checked live and recorded NO-GO, and a read-only quota watcher polls their status without spending[23].

The campaign's terminal-state vocabulary was introduced precisely so that these distinctions survive summarisation, and each lane above carries one of its labels: `REBUILD_READY_LAUNCH_PENDING` for E1 wave-10 and E5 successor-27, `AMENDME`

[22]Sources: `outputs/PES_Phase2_Review_2026-09-12/finish/Pending_Experiments.md`, `outputs/UNBLOCK_CARRYOUT_2026-09-21.md`, `outputs/E1_E14_FINAL_RESULTS_2026-09-19.md`.

[23]Sources: `outputs/PES_Phase2_Review_2026-09-12/finish/Pending_Experiments.md`, `outputs/E1_E14_FINAL_RESULTS_2026-09-19.md`, `outputs/UNBLOCK_CARRYOUT_2026-09-21.md`.

NT_ACCEPTED_LAUNCH_PENDING for E2 and E13, and PENDING_QUOTA for E6 and E9[24].

9.5 The E4 rerun as a diagnostic

The E4 lane closed CLOSED_PARTIAL on 19 September 2026: the pass3/pass16 archive is genuinely missing, only one of six artifacts verifies, and the 0.3115 recovery metric stands as receipt-only evidence because unlimited budget removes a spending cap but cannot reconstruct model outputs that no longer exist. The only funded path recorded at closure was a fresh full rerun whose grader-only lower bound was estimated at approximately $59.45, which sits outside every envelope then authorised. That path was subsequently authorised separately and executed, and its result is reported here as a diagnostic rather than as a score, because that is what its design makes it (Figure 9.4).

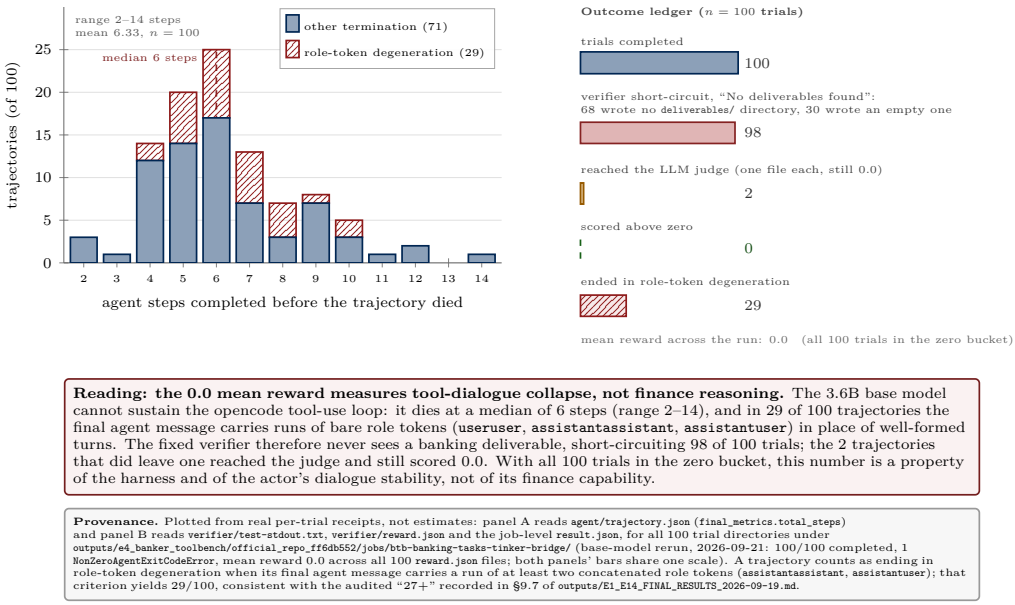


Figure 9.4: The E4 rerun diagnostic. Across 100 trials agents terminate at a median of roughly six steps without producing deliverables, and more than a quarter end in role-token degeneration. The 0.0 mean reward therefore measures tool-dialogue collapse, not finance reasoning.

The rerun was configured on the base model with a zero-initialised LoRA snapshot — the lane's record states this is equivalent to base sampling, with no training steps — executed through the Modal bridge /v1/responses path at --n-concurrent 2 under a cap of $55.91445263. It completed 100 of 100 trials with one NonZeroAgentExitCodeError, and returned a mean reward of exactly

[24]Source: outputs/PES_Phase2_Review_2026-09-12/finish/Pending_Experiments.md.

0.0, with all one hundred trials in the 0.0 bucket, over a runtime of 3 hours 29 minutes[25].

The value of this run lies entirely in the failure mechanism, which was audited across all one hundred trajectories. Agents terminate after 2 to 14 steps, with a median of about 6, and produce no `deliverables/` directory; the verifier consequently short-circuits to 0.0 on the message "No deliverables found". Twenty-seven or more of the trajectories end in role-token degeneration of the form `user user assistant ...` repetition. The record's own interpretation is the correct one and is repeated here without softening: the base model cannot sustain the opencode tool-use loop, so the 0.0 measures tool-dialogue collapse rather than finance reasoning. Two implications follow, and they cut in opposite directions. The positive implication is that the harness behaves correctly at the boundary — it fails closed, produces a legible mechanism for every zero, and does not manufacture partial credit. The negative implication is that a BankerToolBench score for this actor family is not currently obtainable at all, because the bottleneck is the agent loop rather than the domain, and the campaign therefore reports no BankerToolBench suite result for the trained actor either. In particular, the rerun's 0.0 must not be set against the original single-task outcome as though the two were a controlled pairing: they are different configurations under different runtime conditions, and the campaign makes no comparison between them.

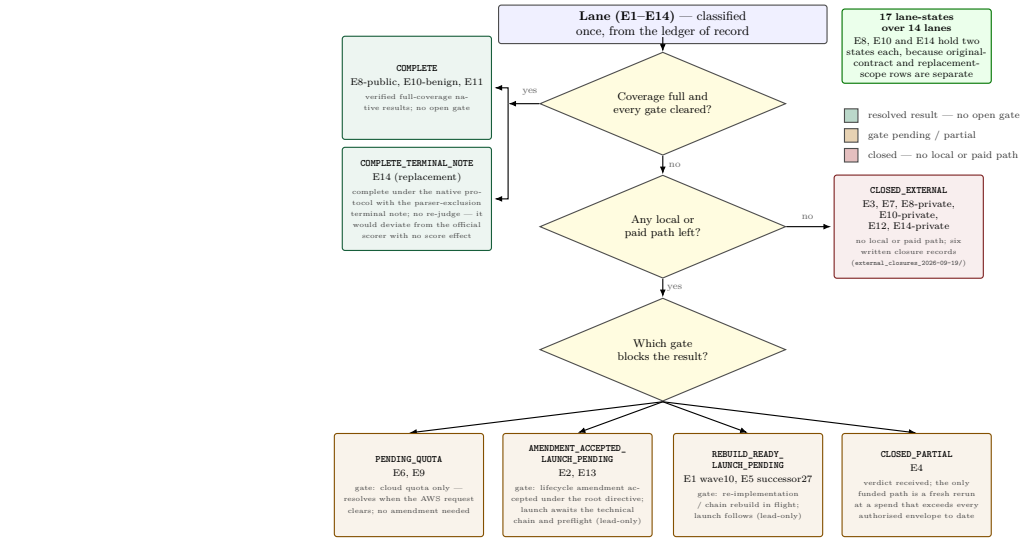
Spend on the rerun was $15.07 charged against the $55.91 cap, comprising 16.35 million prompt tokens and 2.18 million completion tokens.

9.6 Externally blocked lanes and terminal states

Six lanes have no local or paid path and were formally closed as externally blocked under the finish-all directive of 19 September 2026, whose verbatim text and interpretation are recorded in `outputs/PES_Phase2_Review_2026-09-12/finish/ROOT_DIRECTIVE_FINISH_ALL_2026-09-19.json` at 13:50Z. The six are E3 (SDAB private bundle), E7 (BinaryAudit private payload), E8-original (the LifeSciBench package), E10-original (AgentHarm private tasks), E12 (AppBench deployment) and E14-original (the FrontierMath hosted evaluation). Closure records were written for each, with an index, under `outputs/PES_Phase2_Review_2026-09-12/finish/external_closures_2026-09-19/`; each records "no provider response"

[25]Source: `outputs/PES_Phase2_Review_2026-09-12/finish/Pending_Experiments.md`, which cites the receipt at `outputs/e4_banker_toolbench/official_repo_ff6db552/jobs/btb-banking-tasks-tinker-bridge/result.json`.

and specifies the conditions under which the lane could be reopened[26] (Figure 9.5).



Fail-closed. An unrecognised or unlisted state is never absorbed into a percentage: it is recorded as blocked and the gap is carried in the ledger of record.

No pooling. Original-contract and replacement-scope numbers are never pooled — the two scopes are different suites with different task populations.

What these states are not. They classify lanes, not results; none of the pending states asserts a score in either direction, and a lane with incomplete receipts is blocked rather than shortened. The completion gate's `PARTIAL_EXACT` / `PARTIAL_RECOVERY` are a separate evidence-class vocabulary (ch. 5), not terminal states.

Source: `outputs/PES_Phase2_Review_2026-09-12/finish/Pending_Experiments.md`, "Terminal-state vocabulary", 19 Sep 2026 — the seven states above are that table's full documented vocabulary.

Figure 9.5: Terminal-state taxonomy used to close every lane. Each state has a defined meaning and a named reopen condition where one exists.

Access was requested through each provider's public channel: a GitHub issue for E7 (`QuesmaOrg/BinaryAudit#22`), a Hugging Face dataset discussion for E10, and direct requests to the maintainers of E3, E12 and E14. No grant had been received at the time of writing, and delivery of the direct requests could not be confirmed from the sending side[27].

Two further items complete the terminal record. First, verification: the deterministic checks of 19 September passed 11 of 11 with zero failures, covering exact divisions, receipt presence, sealed-artifact hash equality and absence probes for the lost execution sources. Second, the judgment layer: a model-based triage harness was used only for classification and value ordering, never for arithmetic; it became unreliable during a service degradation on 19 September, so the per-lane blocker column in the terminal ledger rests on document evidence rather than on model judgment[28].

[26]Sources: `outputs/E1_E14_FINAL_RESULTS_2026-09-19.md`, `outputs/PES_Phase2_Review_2026-09-12/finish/Pending_Experiments.md`.

[27]Sources: `outputs/UNBLOCK_CARRYOUT_2026-09-21.md`, `outputs/e1_e14_results_2026-09-05/E1_E14_Results.md`.

[28]Sources: `outputs/verification/LEDGER_CODE_CHECK_2026-09-19.json`, `outputs/E1_E14_FINAL_RESULTS_2026-09-19.md`.

9.7 Summary of the campaign at the point of writing

The campaign established two full-suite results and one complete replacement scope with a score: E1 at 2/731 = 0.274 per cent on SWE-bench Pro, E11 at 129/312 = 41.35 per cent pass@1 across its two framings, and E8 at 450/1967 = 0.2288 on the LAB-Bench public split, alongside a complete benign-utility evaluation of E10 at 88 of 97 episodes, a terminal-complete E14 replacement scope with an official accuracy of 51.31 per cent, and five original-contract partials — E2, E4, E5, E7 and E9 — whose coverage fractions are all stated above and none of which is a suite score. It further established that six lanes are externally blocked with no local or paid path, and that six more are staged and waiting on either a sealed launch or a cloud quota that has not been granted. What it did not establish is equally important: no multi-seed estimate, no held-out security result, no Banker-ToolBench suite score, and no baseline comparison of any kind. The next chapter takes up the artefacts that make these results inspectable — the evidence chain, the receipt schema and the reproduction path — so that a reader can re-derive each number above from the files cited beside it.

Part IV

Synthesis

Chapter 10

Cross-Study Synthesis, Threats to Validity, Conclusions and Future Work

10.1 Cross-Study Synthesis

This report has presented two bodies of work built with different instruments. The first is the P1–P8 corpus, which probes GRPO-style post-training across libraries, model scales, group sizes and datasets and ends in a reporting standard, a machine-readable registry and a controller audit. The second is the E1–E14 campaign, a governance-bound evaluation of one Tinker-trained actor against fourteen held-out agentic and reasoning suites. They are the same methodological commitment applied twice — once to *what does GRPO do*, once to *what does a GRPO-trained actor score*. That commitment is attribution: every comparative claim holds the entire training stack fixed except the one factor under test, and where that was unachievable the confound is reported alongside the number rather than absorbed.

The corpus's own synthesis, recorded at the close of Phase 1, is that on this compute scale the *measurement* layer is useful while most *algorithmic* levers are noise-limited. The dominant measured phenomenon is advantage-signal starvation, with the Zero-Variance Fraction running at approximately 0.72–0.77, and it is easy-driven: the collapsed groups are mostly prompts the model has already solved. That rules out the most tempting remedy, because re-baselining the advantages merely converts the signal into a difficulty proxy with no learning content, and the remedy that remains — a curriculum that removes the wasted rollouts — returned an honest null. The narrow systems observation survived better than any algorithmic one: per-step telemetry separated the available method traces more

cleanly than reward-only features, though the classifier's integrity generalisation was never tested[1].

The Semester 4 papers sharpen those threads rather than reversing them. P2's diagnostics are recomputable from stored raw tensors rather than read back from logged summaries: three stored GSM8K group tensors reproduce ZVF at 0.130, 0.190 and 0.155, pooling to 0.1583, and the correlation between ZVF and final held-out outcome is $\rho \approx 0.27$ — real but weak, so ZVF is offered as a diagnostic and not a predictor. P4's matched comparison of GRPO against Dr. GRPO finds the two held-out gains indistinguishable, with pooled mean completion length drifting from 194.4 to 183.5 tokens under a 200-token cap that bounds what the comparison can say. P4's companion independent error-bar audit grades the single P4-facing headline — audit H7, the iter-136 arithmetic late-training-efficiency contrast, paired n = 5, Cohen's d = +2.68, one-sided p_param = 0.031, two-sided p_perm = 0.063 — as DECISIVE under its n < 10 paired-bootstrap rule, while its three NULL verdicts (H3, H4, H5) fall on Pillar-1 and Pillar-3 headlines rather than on this paper's claims. Across all seven audited headlines the aggregate is four DECISIVE, three NULL and none merely suggestive. The asymmetry is the useful part: the headline of ours that survived adversarial audit is a late-training efficiency contrast, not a capability claim[2].

P3 and P7 both treat group size as a lever and are best read as a pair, because they disagree in emphasis. P3 finds that under saturation the retention ratio between G = 2 and G = 16 is approximately 1.00, that the effective signal-to-noise ratio reaches only about 52% of the $\sqrt{G}$ ideal, and that a 505-task conditional utility audit selects G = 4 over G = 5 by 0.00177 with a 95% bootstrap interval of [0.00082, 0.00270] — positive, small, and explicitly cohort- and objective-conditional. P7 models usable signal per prompt as S = p(1−p)(1 − h_G(p)) with h_G(p) = p^G + (1−p)^G, derives three falsifiable consequences, and checks them: a 368-run Weights & Biases audit across seven models reproduces the predicted U-shape in accuracy, with ZVF 0.95–0.97 at both reward extremes and 0.25–0.29 at mid-range rewards of 0.35–0.50; a model × G grid confirms the monotone effect at interior accuracy, Qwen3-32B moving 0.33 → 0.18 → 0.08 across G = 4/8/16; and a toy-scale open-backward-pass experiment finds corr($\|\nabla\|$, p(1−p)) = +0.71 on a 0.5B model, which the paper labels directional only. Where P3's sweep could not find an optimum, P7 supplies a mechanism under which a shallow one should exist. That is a reconciliation, not a resolution: the mechanism is checked in

[1]Source: platform_tinker/reports/esa_phase1/Phase1_Project_Report_ZVF.tex.

[2]Source: platform_hybrid/paper/sections/p4_conclusion.tex; platform_hybrid/experiments/results/berkeley/adding_error_bars_summary.json.

telemetry, and its causal link to the gradient remains toy-scale[3].

The same paper is clear about the control it does not have. A micro-jitter falsification collapses batch ZVF from 0.158 to 0.000 while leaving the pairwise-contrast density unchanged, which is why PCD rather than ZVF is promoted as the control signal. In a four-arm open-trainer audit, adaptive group size matches the best fixed-recipe held-out gain, +0.575, at 186 rollouts, while DAPO-style dynamic sampling drives ZVF to 0.00 at +45% rollout cost — an accuracy-versus-compute trade, not a controller win. In the larger 2,560-observation controller trace, 1,723 of 1,867 escalation events, 92.3%, occur on all-correct groups and only 144 on all-wrong, so a symmetric ZVF trigger spends most of its budget on prompts already solved; deleting those fires is a counterfactual only, because performance preservation was not measured. Adaptive group size is therefore not promoted to a recommended policy[4].

P5 and P6 supply the corpus's methodological claim, and they contain its sharpest internal tension. P5's headline exhibits are a nominally matched-configuration backend audit whose final training rewards span 5.0% to 85.6% — a 17× same-label swing the paper declines to assign to the backend, because the managed stack also silently pinned a different base checkpoint, and that undisclosed bundling is itself part of the exhibit — and a pair of runs carrying the same "DAPO" label in which an open trainer with true dynamic sampling produced mean ZVF 0.00 while a closed stack running an asymmetric-clip surrogate produced mean ZVF 0.58. Alongside them sits a third exhibit: a 12-cell head-to-head of four labelled algorithms on one fixed stack, landing within a 0.034 band with mean last-10 reward 0.710–0.744, a spread the paper calls comparable to seed-level variation[5]. P5's framing is that a label-only comparison can hide stack-level variation, and that the converse also holds: when the stack is genuinely fixed, the labels stop separating. P6 reads the same exhibit in the reverse direction, using it to motivate a schema while explicitly declining to re-claim it as novel empirical science; on the twenty seed entries the MIN-REPORT auditor badge spans 43–96, and *stackdiff* flags the open-versus-closed "DAPO" pair as an R5 label-flip risk from manifests alone[6].

Two gaps should be named. The first is that P7's controller audit and P5's stack-attribution programme have not been joined: the controller is evaluated on open trainers where the update rule is auditable, while the standard is motivated by a closed-stack divergence that cannot be audited. The second is P8, the credit-

[3]Source: platform_hybrid/paper/sections/p7_abstract.tex.
[4]Source: platform_hybrid/paper/sections/p7_abstract.tex.
[5]Source: platform_hybrid/paper/sections/p5_abstract.tex.
[6]Source: platform_hybrid/paper/sections/p6_abstract.tex.

card fraud side-probe, absorbed here as a measurement-discipline case study and nothing more. Its two AUCs are not a like-for-like ranking: XGBoost's 0.7955 and the Qwen3.5-4B SFT arm's 0.48268 on a positive-enriched 500-row held-out evaluation sit on different held-out splits by construction[7].

The E1–E14 campaign is a different instrument and should be read as one. A single Tinker-trained actor — base `Qwen/Qwen3.6-35B-A3B` at commit `995ad96e…`, with adapter `arvindcr4/pavlov-portfolio-qwen36-seed809-stepfinal-tinker-cf0ad8c1…` at commit `64444133…`, in bfloat16 — was evaluated against fourteen held-out suites, each through its own native evaluator at a pinned revision. Three scopes closed strictly complete: LAB-Bench public at 1,967/1,967 across eight categories, AgentDojo benign utility at 97/97, and VerilogEval at 312/312 with 129/312 = 41.35% pass@1 canonical. Omni-MATH closed terminal-complete under its native protocol, with all 4,428 dispositions recorded, 4,426 accepted, and the official accuracy of 2,271/4,428 = 51.31% reproduced[8]. Two suites carry full verified scores under their original contracts: SWE-bench Pro at 2/731 = 0.274% pass@1 and VerilogEval. The rest are partial, quota-blocked or externally blocked, each with a terminal state and a named gate[9].

What the campaign establishes is descriptive, and its scope matters. For *this* actor, at *this* revision, under *each suite's own* evaluator and denominator, these are the scores. It is not a post-training delta: no per-suite base-model comparator appears in the campaign ledger, so no campaign number, high or low, is evidence about what the GRPO step contributed. Where the campaign does carry a base-model arm it is the separately authorised E4 BankerToolBench rerun, and there the mean reward was 0.0 across 100/100 completed trials, with the audited mechanism recorded as agents dying after 2–14 steps with no `deliverables/` output so that the verifier short-circuits 0.0; the ledger's verdict is that this 0.0 "measures tool-dialogue collapse, not finance reasoning"[10].

What governance bought is visible in the ledger itself: eleven of eleven deterministic checks passing on 2026-09-19, and a historical snapshot recording E7 as "1/28 errored" sitting visibly beside the superseding table's "1/46 attempted" rather than being quietly reconciled[11].

[7]Source: platform_hybrid/paper/archive/absorbed/P08_fraud/sections/p8_abstract.tex.
[8]Source: outputs/verification/LEDGER_CODE_CHECK_2026-09-19.json.
[9]Source: outputs/E1_E14_FINAL_RESULTS_2026-09-19.md; outputs/PES_Phase2_Review_2026-09-12/finish/Pending_Experiments.md.
[10]Source: outputs/E1_E14_FINAL_RESULTS_2026-09-19.md §9.7.
[11]Source: outputs/PES_Phase2_Review_2026-09-12/finish/Pending_Experiments.md.

10.2 Rigour and Threats to Validity

What rigour was exercised. Diagnostics are recomputable from stored raw tensors, and headline structural claims were re-verified by deterministic code checks on 2026-09-19, 11/11 PASS, covering exact divisions, receipt presence, sealed-artifact hash equality and lost-source absence probes[12]. Held-out n is reported alongside every gain, and failed runs are counted rather than hidden: approximately \$30–35 of Tinker spend went to failed runs, disclosed because concealing it would understate reproduction cost by roughly 25%[13] (Figure 10.1).

Statistical power: single-seed Tinker experiments. Each Tinker configuration ran once, because cost precluded repetition, so there are no variance estimates and no Tinker-side significance tests. The limitations document records that Henderson et al. (2018) recommends at least ten seeds, that this bar was met only for the TRL baseline and the held-out GSM8K evaluation, and that compute constraints then forced that evaluation to five seeds[14]; both are reported rather than one chosen. The 30B MoE scaling analysis uses three seeds; the bootstrap confidence intervals assume independent observations, so correlation from shared initialisation or data ordering could narrow the true intervals; and secondary analyses carry uncorrected p-values. Quantitative conclusions therefore come from the open-source side, where every hyperparameter is auditable, while Tinker figures are descriptive[15].

Horizons of 30–50 steps. The Tinker runs are early-training snapshots; long-horizon effects such as reward hacking, catastrophic forgetting and late-stage policy collapse are unlikely to manifest, and the source conclusion is direct: "Asymptotic claims are unsupported"[16]. This bounds results rather than vitiating them: P1's failure to recover a scaling law across roughly 2.4 orders of magnitude is a bounded negative, and P4's drift from 194.4 to 183.5 mean completion tokens is bounded by the 200-token cap.

Closed Tinker internals. The Tinker SDK exposes a custom loss path, a limited optimiser surface and standard LoRA hyperparameters, but not the exact server-side GRPO loss formulation, the reward normalisation or baseline-subtraction scheme, minibatch or gradient-accumulation strategy, hardware configuration, or system-level telemetry[17]. The consequence is best quoted: Tinker results measure "the **platform's** implementation of GRPO, not an abstract algorithmic specifi-

[12]Source: outputs/verification/LEDGER_CODE_CHECK_2026-09-19.json.
[13]Source: LIMITATIONS_AND_IMPACT.md §3.
[14]Source: LIMITATIONS_AND_IMPACT.md §7.1.
[15]Source: LIMITATIONS_AND_IMPACT.md §6.2.
[16]Source: LIMITATIONS_AND_IMPACT.md §7.1.
[17]Source: LIMITATIONS_AND_IMPACT.md §6.1.

threat to validity | applied mitigation, paired by the rule at left

Internal validity

single-seed runs · short horizons · lost archive

T1. Single-seed Tinker runs. Each Tinker configuration ran once; cost precluded repetition, so no variance estimate and no significance test exists on Tinker data at all. Henderson et al. (2018) recommend at least ten seeds.

→ **Mitigation.** Quantitative conclusions are drawn only from runs with repeated seeds — the TRL baseline and the held-out GSM8K evaluation — while Tinker figures are reported descriptively and marked †. No significance claim rests on a single-seed comparison.

T2. Horizons of 30–50 steps. The runs are early-training snapshots. Reward hacking, catastrophic forgetting and late-stage policy collapse need longer horizons to manifest, so asymptotic claims are unsupported.

→ **Mitigation.** Every claim is scoped to the observed horizon: the scaling-law null is a bounded negative at this horizon, not a statement about GRPO's asymptotics, and the 200-token cap bounds the length-bias drift. No "converges to" language is used.

T3. A missing pass3/pass16 archive. The E4 recovery's per-task artefacts do not survive: zero of six verify, and only `TINKER_BRIDGE.json` does.

→ **Mitigation.** The surviving 0.3115 is reported as a receipt-only verifier recovery metric on one of 100 tasks, labelled `CLOSED_PARTIAL`, never as a suite score or as the fraction 37/128.

Each threat above is paired with the mitigation actually applied, and the pairing is mostly a reduction in what is claimed rather than a repair of the evidence: where the evidence is partial the claim is narrowed to what it supports, not extended to what a reader might prefer.

Construct validity

denominators · no aggregate · scope · sampler

T4. Native denominators are not comparable. Every result sits on its own suite's denominator — 2/731 SWE-bench Pro, 0/812 WebArena, 13/255 BALROG, 40/75 MLE-bench competitions, 4,428 Omni-MATH dispositions — which differ in size and in what they count and exclude.

→ **Mitigation.** Results are never pooled, within lanes or across them (E1's two scopes, E2's, E5's, E9's). A 41.35% and a 0.274% on different denominators license no comparison, average or ranking.

T5. No cross-suite aggregate. No mean, median or composite index over suites appears anywhere in this work, so the campaign has no single headline number.

→ **Mitigation.** A position, not an omission: an aggregate over native denominators would be a number with no referent. The four complete scopes and the two original-contract scores are the honest maximum.

T6. The E10 scope measures benign utility only. E10's completed evidence is 97/97 tasks under AgentDojo's benign scope; its original contract, private AgentHarm tasks, is externally blocked, so the result says nothing about adversarial robustness.

→ **Mitigation.** The benign scope is named at every reference to the result; no adversarial-robustness claim is made for the trained actor.

T7. The retained E11 result came from the original sampler. VerilogEval's 129/312 = 41.35% was sampled through the Tinker managed sampler, as the receipt records. That route is gone — the run is purged — and the recovery path is a vLLM serve of the HuggingFace mirror.

→ **Mitigation.** The lane is receipt-bound, with provenance recorded down to the sampler path. No backend numerical-parity claim is made, and a re-run would be a same-weights, different-sampler comparison whose divergence is not measured.

External validity

closed platform · withheld third-party access

T8. Closed Tinker internals. The SDK exposes a custom loss path, a limited optimiser surface and standard LoRA hyperparameters, but not the server-side GRPO loss formulation, the reward normalisation or baseline-subtraction scheme, minibatch construction, hardware configuration or system telemetry. Tinker results therefore measure the *platform's* implementation of GRPO, not an abstract algorithmic specification; the SDK is also a moving target, with Phase 1 pinned at `tinker==0.16.1` and the September 2026 bridge work requiring `tinker==0.30.0`.

→ **Mitigation.** No backend numerical-parity claim is made. Serving conditions and the pinned SDK version are recorded per lane, and the stack is reported as part of the result: a label without its stack is not a reproducible experimental specification.

T9. Third-party access withheld. Six original contracts could not obtain their private bundles or hosted evaluations — E3 SDAB, E7 BinaryAudit, E8 LifeSciBench, E10 AgentHarm, E12 AppBench and E14 FrontierMath — and the access requests remain unanswered, so those scores are not producible outside the provider.

→ **Mitigation.** Each lane carries a terminal close-out record with a named reopen condition. The replacement scope is reported under its own name and never under the original's, and no substitute number is reported where the original was withheld.

† marks figures and summary statistics drawn from Tinker data only; independent replication of them requires Tinker API access (`LIMITATIONS_AND_IMPACT.md`, Sec. 6.3).

Figure 10.1: Threats to validity grouped as internal, construct and external, each with the mitigation applied in this work.

cation". The illustrative case is a reader asking why Tinker GRPO scores 99.9% on GSM8K while TRL GRPO scores 73.4% on the same task; seed variance and a model-size confound can be ruled out through the same-model Qwen3-8B comparison, but the implementation itself cannot[18]. The threat propagates into the campaign, whose actor was trained on Tinker. The SDK is also a moving target: Phase 1 pinned `tinker==0.16.1`, while the September 2026 bridge required `tinker==0.30.0` after the older SDK produced a no-longer-supported error[19].

Per-suite native denominators are not comparable. Every E1–E14 result is expressed on its own suite's denominator, and those denominators differ in size, in what they count and in what they exclude. SWE-bench Pro is 2/731, whose denominator retains 14 generation failures and 4 artifact losses alongside 713 native evaluations; APEX-Agents is 7/480; E9 MLE-bench is 40/75 competitions natively graded, stated as 53.33% coverage with a null suite score, because coverage and score are different quantities; E13 BALROG is 13/255 episodes; E6 WebArena is 0/812; E9's replacement MLDevBench is 0/34 graded; E1's replacement SWE-bench Multilingual is 35/300 graded; and E14's Omni-MATH records 4,428 dispositions of which 4,426 were accepted, with the official accuracy computed over all 4,428[20]. A 41.35% on one denominator and a 0.274% on another do not license a comparison, an average or a ranking, and none is offered; the rule holds within lanes too, and the ledger states it flatly: "N/A means unverified; no pooled results"[21].

No cross-suite aggregate. No mean, median or composite index over suites appears anywhere in this report or in the campaign materials it draws on: an aggregate over native denominators would be a number with no referent, mixing suites that measure different capabilities at different difficulties under different scoring rules. The cost is that the campaign has no single headline; the three strictly complete scopes, the terminal-complete Omni-MATH scope and the two original-contract scores are the honest maximum.

The E10 benign-utility scope says nothing about adversarial robustness. The E10 lane's completed evidence is a benign-utility evaluation of 97/97 tasks under AgentDojo's benign scope; its original contract, private AgentHarm tasks, is externally blocked and carries a closure record with a named reopen condition. The completeness result therefore carries no adversarial-robustness measurement, and this report makes no such claim for the trained actor. The lane has a dual-use dimension, though the limitations analysis finds the marginal uplift of the

[18]Source: LIMITATIONS_AND_IMPACT.md §6.2.
[19]Source: outputs/TINKER_SAMPLER_STATUS_2026-09-20.json.
[20]Source: outputs/E1_E14_FINAL_RESULTS_2026-09-19.md.
[21]Source: outputs/PES_Phase2_Review_2026-09-12/finish/Pending_Experiments.md.

released artefacts small, because the base model's capability is overwhelmingly pre-existing: the Qwen3-8B base-model control already scores 82.0% held-out on GSM8K, and full GRPO adds only +1.3 points to 83.3%, not significant at p = 0.26[22].

The retained E11 result came through the original sampler. The VerilogEval lane's paid-run receipt records `model.sampler_path = tinker://cf0ad8c1-1f1b-5ff3-8bd7-2a0bf232657b:train:0/sampler_weights/step_seed809_40` with `model.source = tinker_sampler_checkpoint`, so the scores come from sampling the trained adapter through the Tinker managed sampler rather than from a locally served copy of the weights[23]. That route no longer exists: the sampler status record of 2026-09-20 reports a bridge deployment booting to a sampler-weights 404, `list_checkpoints` on the `cf0ad8c1` run returning 404 with the run purged, no `last_sampler_checkpoint` on any of twenty training runs, and the status string `TINKER_SAMPLER_WEIGHTS_DELETED_E9_AND_BRIDGE_AFFECTED`; the recorded recovery path is to serve the HuggingFace mirror `arvindcr4/pavlov-portfolio-qwen36-seed809-stepfinal` at commit `64444133…` through vLLM on Modal[24]. The threat is not that the number is wrong — it is receipt-bound, provenance recorded — but that the sampling path is not re-runnable, so a re-run would be a same-weights, different-sampler comparison whose divergence is unmeasured. The receipt's `outstanding_blockers` field further records that the decontamination receipt ID does not exist and that the split-manifest validator still fails on it[25].

Known evidence loss in the finish-era recovery source. The record is explicit: the `.codex-run` deletion recorded in `finish/STATE_RECONCILIATION_2026-09-19.md` removed the execution sources of the finish-era recovery and build code — the E1 wave-10 controller, the E9 restored dependencies and the E6 validator — while sealed requests, validation receipts and test logs survive with matching hashes, and the replacement lanes' native runners survive under `zvf-program/flagship/`[26]. The lost-source policy was applied rather than waived: a faithful re-implementation was written into a tracked path under `zvf-program/e1_wave10/`, where it verifies 10/10 targeted regressions offline (`test_process_group.py` 4/4, `test_cleanup_regressions.py` 6/6)[27]. Two consequences follow: a sealed request without its execution source is not a rerun, so the E9 MLDevBench runtime recipe reconstruction is still owed; and the policy re-

[22]Source: LIMITATIONS_AND_IMPACT.md §2.2, §2.3.
[23]Source: outputs/e11_verilog_eval/e11_trained_step40_receipt.json.
[24]Source: outputs/TINKER_SAMPLER_STATUS_2026-09-20.json.
[25]Source: outputs/e11_verilog_eval/e11_trained_step40_receipt.json.
[26]Source: outputs/E1_E14_FINAL_RESULTS_2026-09-19.md §6.
[27]Source: outputs/E1_E14_FINAL_RESULTS_2026-09-19.md §9.5.

sponse — execution sources henceforth live under tracked paths — cannot restore what was already lost. A related item: the Jev (TypeSafe) judgment harness lost reliability from roughly 13:10Z on 2026-09-19, both blocker-classification batteries returning degenerate distributions and a controlled yes-probe falling from about 1.0 to 0.3; the receipts are retained as incident evidence and the batteries must be re-run once the health check passes, so the ledger's first-uncleared-blocker column stands on document evidence rather than judgment output[28].

Scope and confound inventory. Several Phase-1 threats remain live. Many early runs logged train-set rewards only, so time-series comparisons spanning old and new runs must account for the instrumentation change. Initial phases used LoRA at ranks 8–64 exclusively; historical baseline comparisons use default hyperparameters from the Tinker cookbook and the TRL documentation; and no full learning-rate × rank × batch × group-size grid was swept. Tinker, Modal, Colab and the PES A100 node use different accelerators, memory hierarchies and training stacks, so observed differences in reward dynamics are not purely algorithmic, and the work was not pre-registered. The limitations record gives the reliable model-scale range as 0.6B–235B; the P1 roster's anchors above that, up to 1T, are single-seed and descriptive (Chapter 6 §6.1), so dense-70B and trillion-parameter MoE extrapolations are unreliable; benchmark coverage is narrow and biased toward verifiable rewards such as exact match, unit tests and schema validity, with no human preference data, so findings should not be extended to reward-model-based RLHF; and the work was carried out at one institution in Bengaluru, so benchmark choices and evaluation design reflect that context[29].

The resource accounting is itself a threat. The recorded out-of-pocket spend was $130–140, with roughly 60 A100 GPU-hours of in-kind PES University compute valued at approximately $270 and projected carbon of about 776 kWh and about 296 kg CO_2-eq; the dominant uncertainty is the Tinker GPU-hour count, inferred because the platform exposes no hardware telemetry, giving a pessimistic Tinker contribution of about 540 kg against an optimistic one of about 130 kg. No offsets were purchased; the mitigation offered is reproducibility by artefact[30].

10.3 Conclusions

This report set out to build a multi-framework benchmark for GRPO-style post-training of large language models and to study that post-training under a single

[28]Source: outputs/E1_E14_FINAL_RESULTS_2026-09-19.md §8.
[29]Source: LIMITATIONS_AND_IMPACT.md §7.2, §7.3.
[30]Source: LIMITATIONS_AND_IMPACT.md §3, §4.

methodological rule: hold the stack fixed, change one factor, report what happens. Both halves were completed, and the honest summary is less flattering, and more useful, than a positive result would have been.

On the analytical side the contribution is a measurement layer and a set of bounded negatives. The Zero-Variance Fraction is a reproducible diagnostic of advantage-signal starvation, recomputable from stored tensors, measured at roughly 0.72–0.77 with a weak $\rho \approx 0.27$ association with final held-out outcome — descriptive, not predictive. Its theory has been made falsifiable, and two of its three consequences reproduced: the U-shape in accuracy across reward levels and the monotone group-size effect; the gradient link remains toy-scale and directional[31]. Group size therefore has a candidate mechanism, but the measured three-seed sweep found its arms statistically indistinguishable and establishes no optimum (Chapter 6 §6.4). Length bias is not established: the roster-level flag that marks four of eleven GRPO runs — peak before 65% of training, terminal reward below 90% of peak[32] — has post hoc thresholds, is not specific to length, and fires on most runs in the wider trace pool, so it is not evidence of length bias; in the controlled comparison neither GRPO nor Dr. GRPO shows the pattern, completion length drifts shorter, and the two are indistinguishable on held-out gain (Chapter 6 §6.5). No claim is made that Dr. GRPO is uniquely free of the pattern. The levers we could test — a naive curriculum, a matched-token-budget intervention, an integer group-size sweep — did not produce reliable held-out gains. The strongest original contribution remains P5, the reporting standard, whose status the record states precisely: a targeted search of the GRPO, model-reporting and provenance literature through July 2026 did not identify an equivalent GRPO-specific machine-readable run datasheet with rollout provenance, and this is "an evidence-bounded novelty claim, not proof of non-existence"[33]. The earlier second bet — predictive layer freezing from step-1 behaviour — is weakened rather than withdrawn: it held at 1.5B and collapsed to chance at 3B across four seeds, at a mean of 0.083 ± 0.048[34].

On the empirical side the E1–E14 campaign delivered what a governance-first campaign can deliver: complete, receipt-bound results on three suites, with Omni-MATH terminal-complete; full verified original-contract scores on two; and an explicit, auditable terminal state for every remaining lane, with a named gate and a named reopen condition instead of an indefinite pending. Its discipline is negative, which is why it is credible: original-contract and replacement-scope

[31]Source: platform_hybrid/paper/sections/p7_abstract.tex.
[32]Source: LIMITATIONS_AND_IMPACT.md §2.2.
[33]Source: platform_tinker/reports/esa_phase1/Phase1_Project_Report_ZVF.tex.
[34]Source: platform_hybrid/experiments/results/p1_layerfreeze/scaled_4seed_result.json.

numbers are never pooled, no cross-suite aggregate is computed, and no claim of improvement over a baseline or over the base model is made anywhere in this work. The strongest complete positive result — 129/312 = 41.35% pass@1 on VerilogEval, at full 312/312 coverage — is reported as a measurement of one actor on one suite, with its provenance recorded down to the sampler path.

The report's overall conclusion is about method rather than algorithms. On this compute scale, and with these instruments, the measurements are informative while the algorithmic levers are noise-limited; the stack is a material part of the result, so a label without its stack is not a reproducible experimental specification; and the most durable artefacts produced here are the ones that make other people's results checkable — the ZVF/PCD diagnostic layer, the eight-item minimum reportable stack, the registry and its CLI, and a campaign ledger in which every number is bound to a receipt and every gap has a gate.

10.4 Future Work

The work leaves three kinds of open question: evaluation lanes whose measurement is incomplete, analytical claims that the present evidence could not settle, and methodological tools that have been built but not yet tested at scale.

Completing the held-out campaign. Four replacement lanes — E1 (SWE-bench Multilingual), E2 (CORE-Bench), E5 (Tau3) and E13 (BALROG) — have rebuilt, offline-verified launch paths, and the ledger's estimate for completing every lane that is not externally blocked is approximately $108[35]. Two further lanes, E6 (WebArena) and E9 (MLDevBench), wait on cloud compute capacity: E9's runtime builder needs a documented minimum of 4 vCPUs (8 requested) and E6 needs 16, against a live quota of 1 in each region[36]. Running these lanes would change the campaign's coverage, not the meaning of the numbers already recorded. Seven original-contract scopes are blocked on external access — E3 (SDAB private bundle), E7 (BinaryAudit private payload), E8-original (LifeSciBench package), E10-original (AgentHarm private tasks), E12 (AppBench deployment), E13-original (OpenReward held-out games) and E14-original (FrontierMath hosted evaluation). Six of them are recorded CLOSED_EXTERNAL with closure records; E13's lane continues through its BALROG replacement scope. Each reopens only on a provider response, not on an internal milestone[37]. A full BankerToolBench

[35]Source: outputs/E1_E14_FINAL_RESULTS_2026-09-19.md §7.

[36]Source: outputs/E1_E14_FINAL_RESULTS_2026-09-19.md §9.3; outputs/PES_Phase2_Review_2026-09-12/finish/e9_completion/status_2026-09-19.json.

[37]Source: outputs/E1_E14_FINAL_RESULTS_2026-09-19.md §5; outputs/PES_Phase2_Review_2026-09-12/finish/Pending_Experiments.md.

score for E4 would be a new-runtime experiment with a grader-only lower bound of approximately $59.45, and the 100/100-trial base-model rerun at mean reward 0.0 provides its calibration point[38].

Turning the campaign into a post-training measurement. The campaign reports scores for one actor, not the contribution of the GRPO step. Adding a per-suite base-model arm under the same evaluators and denominators would turn each lane from a descriptive score into a post-training delta. Separately, the retained E11 result was sampled through a Tinker sampler route that no longer exists. Re-serving the published adapter mirror and re-running E11 would measure the same-weights, different-sampler divergence that is currently unmeasured[39].

Settling the analytical questions. Five Phase-1 directions remain open, each answering a threat identified above[40]:

- adequately powered evaluation, moving from held-out n of 8–20 to the full GSM8K test split or a large stratified subset at five or more seeds;
- a token-budget-optimal curriculum with staleness bounds, tested multi-seed;
- testing whether the mid-to-late band in the per-layer gradient profile supports a FLOP-saving freeze schedule, even though step-1 predictability did not survive scaling;
- hardening the Phase-1 method-trace classifier on grouped run-level and temporal splits rather than row-wise cross-validation;
- consolidating the diagnostic findings and the reporting standard for publication.

Three more follow from the Chapter 6 results. The first is a length-confounded regime, with the completion cap lifted and longer horizons, in which a Dr. GRPO length advantage would be expected to appear if it exists. The second is the head-to-head protocol between ZVF and the pairwise-contrast density, to test whether PCD meets its target association. The third is a harder, unsaturated task on which a group-size effect could be detected at all.

Testing the methodological tools. Three extensions come from the Semester-4 work:

- a pre-registered controller bakeoff against static G = 16 and naive boundary heuristics, the test P7 declines to claim in advance of;

[38]Source: outputs/PES_Phase2_Review_2026-09-12/finish/Pending_Experiments.md; outputs/E1_E14_FINAL_RESULTS_2026-09-19.md §9.7.
[39]Source: outputs/TINKER_SAMPLER_STATUS_2026-09-20.json.
[40]Source: platform_tinker/reports/esa_phase1/Phase1_Project_Report_ZVF.tex.

- a cross-framework divergence study that runs an identical task, reward, decoding and seed configuration across TRL, veRL, OpenRLHF and Tinker, turning P5's attribution argument from an audit of existing runs into a designed experiment;
- dogfooding the registry schema on the campaign's own runs, so that every future benchmark result ships with a verifiable provenance record.

Two commitments carry forward, because the threats above are structural rather than incidental. The first is the never-pool rule: original-contract and replacement-scope numbers stay separate, and per-suite native denominators are never averaged. The second is candour about failure. Here it is not a stylistic preference but the mechanism that made the ledger trustworthy: the E11 sampler path, the E4 zero, the lost execution sources, the seed-count discrepancy and the failed-run spend are all reported. Any future result that turns out badly will be reported the same way.

References

[1] P. F. Christiano, J. Leike, T. B. Brown, M. Martic, S. Legg, and D. Amodei, "Deep reinforcement learning from human preferences," *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[2] D. M. Ziegler, N. Stiennon, J. Wu, T. B. Brown, A. Radford, D. Amodei, P. Christiano, and G. Irving, "Fine-tuning language models from human preferences," 2019. [Online]. Available: https://arxiv.org/abs/1909.08593

[3] N. Stiennon, L. Ouyang, J. Wu, D. Ziegler, R. Lowe, C. Voss, A. Radford, D. Amodei, and P. F. Christiano, "Learning to summarize with human feedback," *Advances in Neural Information Processing Systems*, vol. 33, pp. 3008–3021, 2020.

[4] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray *et al.*, "Training language models to follow instructions with human feedback," *NeurIPS*, 2022.

[5] Z. Shao, P. Wang, Q. Zhu, R. Xu, J. Song, M. Zhang, Y. K. Li, Y. Wu, and D. Guo, "DeepSeekMath: Pushing the limits of mathematical reasoning in open language models," *arXiv preprint arXiv:2402.03300*, 2024.

[6] A. Ahmadian, C. Cremer, M. Gallé, M. Fadaee, J. Kreutzer, O. Pietquin, A. Üstün, and S. Hooker, "Back to basics: Revisiting REINFORCE-style optimization for learning from human feedback in LLMs," *arXiv preprint*, 2024, arXiv:2402.14740.

[7] DeepSeek-AI, D. Guo, D. Yang, H. Zhang, J. Song, R. Zhang, R. Xu, Q. Zhu, S. Ma, P. Wang *et al.*, "DeepSeek-R1: Incentivizing reasoning capability in LLMs via reinforcement learning," *arXiv preprint*, 2025, arXiv:2501.12948.

[8] R. Rafailov, A. Sharma, E. Mitchell, S. Ermon, C. D. Manning, and C. Finn, "Direct preference optimization: Your language model is secretly a reward model," *NeurIPS*, 2023.

[9] Z. Liu, C. Chen, W. Li, P. Qi, T. Pang, C. Du, W. S. Lee, and M. Lin, "Understanding R1-Zero-like training: A critical perspective," *arXiv preprint arXiv:2503.20783*, 2025.

[10] Q. Yu, Z. Zhang, R. Zhu, Y. Yuan, X. Zuo, Y. Yue, W. Dai, T. Fan, G. Liu, L. Liu *et al.*, "DAPO: An open-source LLM reinforcement learning system at scale," *arXiv preprint*, 2025, arXiv:2503.14476.

[11] N. Lambert, J. Morrison, V. Pyatkin, S. Huang, H. Ivison, F. Brahman, L. J. V. Miranda *et al.*, "Tulu 3: Pushing frontiers in open language model post-training," *arXiv preprint*, 2024, arXiv:2411.15124.

[12] M. Mitchell, S. Wu, A. Zaldivar, P. Barnes, L. Vasserman, B. Hutchinson, E. Spitzer, I. D. Raji, and T. Gebru, "Model cards for model reporting," in *Proceedings of the Conference on Fairness, Accountability, and Transparency (FAT*)*, 2019, pp. 220–229.

[13] T. Gebru, J. Morgenstern, B. Vecchione, J. W. Vaughan, H. Wallach, H. Daumé III, and K. Crawford, "Datasheets for datasets," *Communications of the ACM*, vol. 64, no. 12, pp. 86–92, 2021.

[14] E. M. Bender and B. Friedman, "Data statements for natural language processing: Toward mitigating system bias and enabling better science," *Transactions of the Association for Computational Linguistics (TACL)*, vol. 6, pp. 587–604, 2018.

[15] M. Pushkarna, A. Zaldivar, and O. Kjartansson, "Data cards: Purposeful and transparent dataset documentation for responsible ai," in *Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency (FAccT '22)*, 2022, pp. 1776–1826.

[16] P. Henderson, R. Islam, P. Bachman, J. Pineau, D. Precup, and D. Meger, "Deep reinforcement learning that matters," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, no. 1, 2018.

[17] C. Colas, O. Sigaud, and P.-Y. Oudeyer, "A hitchhiker's guide to statistical comparisons of reinforcement learning algorithms," *arXiv preprint*, 2019, arXiv:1904.06979.

[18] R. Agarwal, M. Schwarzer, P. S. Castro, A. C. Courville, and M. Bellemare, "Deep reinforcement learning at the edge of the statistical precipice," in *Advances in Neural Information Processing Systems*, vol. 34, 2021, pp. 29 304–29 320, arXiv:2108.13264; supplementary materials at https://agarwl.github.io/rliable/.

[19] S. M. Jordan, A. White, B. Castro da Silva, M. White, and P. S. Thomas, "Position: Benchmarking is limited in reinforcement learning research," *arXiv preprint*, 2024, arXiv:2406.16241.

[20] J. Dodge, S. Gururangan, D. Card, R. Schwartz, and N. A. Smith, "Show your work: Improved reporting of experimental results," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 2019, pp. 2185–2194.

[21] J. Pineau, P. Vincent-Lamarre, K. Sinha, V. Larivière, A. Beygelzimer, F. d'Alché Buc, E. Fox, and H. Larochelle, "Improving reproducibility in machine learning research: A report from the NeurIPS 2019 reproducibility program," *Journal of Machine Learning Research*, vol. 22, no. 164, pp. 1–20, 2021. [Online]. Available: https://jmlr.org/papers/v22/20-303.html

[22] J. Hilton, J. Tang, and J. Schulman, "Scaling laws for single-agent reinforcement learning," *arXiv preprint*, 2023, arXiv:2301.13442.

[23] L. Gao, J. Schulman, and J. Hilton, "Scaling laws for reward model overoptimization," *ICML*, 2023.

[24] R. Rafailov, Y. Chittepu, R. Park, H. Sikchi, J. Hejna, B. Knox, C. Finn, and S. Niekum, "Scaling laws for reward model overoptimization in direct alignment algorithms," *arXiv preprint*, 2024, arXiv:2406.02900.

[25] D. Nimmaturi, V. Bhargava, R. Ghosh, J. George, and D. Dutta, "Predictive scaling laws for efficient GRPO training of large reasoning models," *arXiv preprint*, 2025, arXiv:2507.18014; identifies three consistent training phases: slow start, rapid improvement, plateau.

[26] Z. Tan, H. Geng, X. Yu, M. Zhang, G. Wan, Y. Zhou, Q. He, X. Xue, H. Zhou, Y. Fan, Z. Li, Z. Zhang, G. Zhang, C. Zhang, Z. Yin, P. Torr, and L. Bai, "Scaling behaviors of LLM reinforcement learning post-training: An empirical study in mathematical reasoning," *arXiv preprint arXiv:2509.25300*, 2025.

[27] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen, "LoRA: Low-rank adaptation of large language models," *ICLR*, 2022.

[28] T. Dettmers, A. Pagnoni, A. Holtzman, and L. Zettlemoyer, "QLoRA: Efficient finetuning of quantized language models," *NeurIPS*, 2023.

[29] S. Mukherjee, L. Yuan, D. Hakkani-Tur, and H. Peng, "Reinforcement learning finetunes small subnetworks in large language models," *arXiv preprint arXiv:2505.11711*, 2025.

[30] H. Lightman, V. Kosaraju, Y. Burda, H. Edwards, B. Baker, T. Lee, J. Leike, J. Schulman, I. Sutskever, and K. Cobbe, "Let's verify step by step," *arXiv preprint arXiv:2305.20050*, 2023.

Part V

Appendices

Appendix A

Complete Run Registry

This appendix is the auditable index of every training and evaluation run on which the claims of this report rest. It does not present results — those live in Chapters 6–9 — but makes each claim traceable to the artefacts that produced it: run identifier, library and platform, base model, algorithm, group size, dataset, seed, steps, and terminal status. Where a claim cannot be traced through this appendix to a run, the project's evidence rules require it to be withdrawn rather than stated.

The registry covers 1,708 runs: 1,662 in the machine-readable ledger and 46 more recorded only in the Semester-4 opening artefacts of §A.3.1. The ledger, `platform_hybrid/experiments/results/experiment_ledger.tsv`, holds one row per run with the columns `source, project, run_id, model, task, algo, G, steps, created, headline_metric`. Its 1,662 data rows decompose into **844 runs inventoried from the Tinker training service** and **818 runs inventoried from Weights & Biases**. The Tinker inventory was built by paginating `RestClient.list_training_runs` and `list_checkpoints`; the W&B inventory was built per project and validated against the live W&B API. The ledger's provenance document, `EXPERIMENT_LEDGER.md`, records the same grand total and notes that the W&B workspace holds fourteen projects, not the thirteen previously believed (`tinker-new-research` was the omission).

Two other totals appear in the repository and are deliberately *not* used here: the identity-audit manifest records 949 Tinker and 940 W&B runs, and `tinker_runs.txt` reports "20 training runs (963 more not shown)". These are snapshots of the same platforms at different times and are not reconcilable without re-querying them. The ledger is the only artefact carrying per-run identifiers for both platforms in one file, so it is the figure used throughout for ledgered runs. The 46 runs outside it are counted separately, from the per-run `name` and mode fields of the artefacts that recorded them, and are named individually in §A.3.1.

A.1 Composition and temporal split

Semester 3 ends at the tag `capstone-final-2026-04-25`, boundary commit `21a99ef7` dated 23 April 2026. Because every ledger row carries a creation timestamp, the two eras separate mechanically rather than by assertion: **806 runs were created before 2026-04-23** and **856 on or after it**.

Registry composition by logging project.

Table A.1: Registry composition by logging project, with temporal split and distinct base models.

Project	Source	Runs	Era	Distinct base models	Algorithm	G	Dataset	Status
`tinker` (training-service inventory)	Tinker API	844	387 S3 / 457 S4	31	—	—	—	all terminal at snapshot
`tinker-rl-lab-world-class`	W&B	174	Semester 3	14	GRPO	—	gsm8k	logged
`tinker-rl-scaling`	W&B	88	Semester 3	—	—	—	—	logged
`tinker-structural-ceiling`	W&B	72	Semester 3	—	—	—	—	logged
`tinker-rl-webarena`	W&B	40	Semester 3	—	—	—	WebArena	logged
`webarena-eval`	W&B	26	Semester 3	—	—	—	WebArena	logged
`tinker-agentic-smoke`	W&B	10	Semester 3	—	—	—	—	smoke
`huggingface`	W&B	3	Semester 3	—	—	16	—	logged
`skyrl-tinker`	W&B	3	Semester 3	Qwen/Qwen3-8B	grpo	8	—	logged
`tinker-rl-zvf-counterfactual`	W&B	9	3 S3 / 6 S4	—	—	—	—	logged
`zvf-audit`	W&B	368	Semester 4	3	—	4, 8	llama3, qwen3_instruct	logged
`zvf-colab-experiments`	Colab → W&B	16	Semester 4	—	—	—	—	logged
`tinker-new-research`	W&B	9	Semester 4	—	—	—	—	logged

Runs span 8 January to 4 July 2026. The split is not clean with respect to authorship — a single Tinker account was used throughout, with 387 Tinker rows before the boundary and 457 after — which is why the inherited-versus-individual

statement in Appendix B rests on commit history rather than run counts.

A.2 Semester-3 shared benchmark runs

The Semester-3 contribution was the multi-framework benchmark infrastructure: a common harness, a fixed evaluation configuration, and integrations for several independent RL post-training libraries. These runs are the shared baseline measurements, and they are the ones for which the registry is *most* complete, because each was produced under a declared configuration.

Cross-framework comparison at the fixed benchmark configuration. Configuration for all four rows: model `Qwen/Qwen3-8B` (Tinker row uses `Qwen/Qwen3-8B-Base`), algorithm GRPO, group size G = 8, learning rate 1e-5, dataset GSM8K[:500], 30 steps, seed 42. Metric is `last10_avg`, the mean of the final ten reported training rewards. Source: `platform_hybrid/experiments/results/framework_comparison.json`, generated by `aggregate_framework_comparison.py`.

Table A.2: Semester-3 cross-framework shared benchmark runs.

Run identifier	Stack / platform	Model	Algo	G	Dataset	Seed	Steps	Outcome / status
`campaign_v2_w1_qwen3-8b-base`	Tinker	Qwen/Qwen3-8B-Base	GRPO	8	gsm8k-500	42	30	last10 = 0.85625 — **real run**
`modal_trl_trl_qwen3_8b`	TRL on Modal H100	Qwen/Qwen3-8B	GRPO	8	gsm8k-500	42	30	last10 = 0.05, 735.7 s — **real run**
`verl-dryrun`	verl	Qwen/Qwen3-8B	GRPO	8	gsm8k-500	42	30	last10 = 0.5528 — **seeded dryrun, not a training run**
`openrlhf-dryrun`	OpenRLHF	Qwen/Qwen3-8B	GRPO	8	gsm8k-500	42	30	last10 = 0.4785 — **seeded dryrun, not a training run**

The two `dryrun` rows are a documented threat to validity, not a result: the generator falls back to a deterministic seeded trace from `verl.VERLTrainer` / `OpenRLHFTrainer` when the framework is absent from the sandbox, and the JSON

carries the note "Seeded deterministic fallback … (verl not installed in sandbox)". Any comparison reading them as measurements of verl or OpenRLHF is invalid. The Tinker row also initialises from the *base* checkpoint while the open-framework rows use the instruction-tuned one, so even the two real rows are not a clean framework contrast.

Modal multi-framework campaign, 4 frameworks × 5 seeds = 20 runs. Source: `modal_results_all.json` (repository root). All rows: model `Qwen/Qwen2.5-0.5B`, dataset arithmetic/math, 125 training steps, NVIDIA L4, one run per seed. Seeds are 42, 123, 456, 789, 1024. Group size is not recorded in the artefact and is therefore shown as "—".

Table A.3: Semester-3 Modal campaign runs.

Run identifier	Stack / platform	Model	Algo	G	Dataset	Seed	Steps	Outcome / status
`trl_grpo_math`	TRL / Modal L4	Qwen/Qwen2.5-0.5B	GRPO	—	math	42, 123, 456, 789, 1024	125	final acc 0.735 / 0.810 / 0.620 / 0.740 / 0.765; 139.8–183.7 s
`sb3_ppo_math`	Stable-Baselines3 / Modal L4	Qwen/Qwen2.5-0.5B	PPO	—	math	42, 123, 456, 789, 1024	125	final acc 0.003 / 0.014 / 0.011 / 0.012 / 0.010; `steps_to_95` null for all five
`cleanrl_ppo_math`	CleanRL / Modal L4	Qwen/Qwen2.5-0.5B	PPO	—	math	42, 123, 456, 789, 1024	125	final acc 0.010 / 0.008 / 0.014 / 0.009 / 0.004; 25.4–32.4 s
`tianshou_ppo_math`	Tianshou / Modal L4	Qwen/Qwen2.5-0.5B	PPO	—	math	42, 123, 456, 789, 1024	125	final acc 0.011 / 0.002 / 0.005 / 0.002 / 0.009; 1.67–3.30 s

Group-size sweep, shared-harness runs. Source: `platform_hybrid/experiments/results/groupsize_zvf_sweep.tsv`. Model `Qwen/Qwen3-8B` on GSM8K, 3 seeds per group size (12 runs). Held-out accuracy is the mean over seeds; `mean ZVF` is the mean zero-variance fraction.

Table A.4: Semester-3 group-size sweep runs.

Run identifier	Stack / platform	Model	Algo	G	Dataset	Seed	Steps	Outcome / status
`groupsize_g2`	Tinker	Qwen/Qwen8B	GRPO	2	GSM8K held-out	3 seeds	—	held-out acc 0.9817 ± 0.0044; mean ZVF 0.8380
`groupsize_g4`	Tinker	Qwen/Qwen8B	GRPO	4	GSM8K held-out	3 seeds	—	held-out acc 0.9883 ± 0.0017; mean ZVF 0.7635
`groupsize_g8`	Tinker	Qwen/Qwen8B	GRPO	8	GSM8K held-out	3 seeds	—	held-out acc 0.9900 ± 0.0029; mean ZVF 0.6906
`groupsize_g16`	Tinker	Qwen/Qwen8B	GRPO	16	GSM8K held-out	3 seeds	—	held-out acc 0.9783 ± 0.0060; mean ZVF 0.6312

The remaining Semester-3 runs are the exploratory blocks of Table A.1: `tinker-rl-lab-world-class` (174 runs over fourteen base models from Llama-3.2-1B to Qwen3-235B-A22B), `tinker-rl-scaling` (88, the material behind the cross-scale study), `tinker-structural-ceiling` (72), `tinker-rl-webarena` (40) plus the separate `webarena-eval` scoring pass (26), the agentic smoke block (10), three SkyRL-on-Tinker runs, and three Hugging Face runs. These are indexed by identifier in the ledger but given here in aggregate, because for most of them fewer than half the required columns have values.

A.3 Semester-4 P1–P8 study runs

The P1–P8 series is the individual Semester-4 contribution, and its runs are indexed by *claim* rather than by project. The audit ledger at `platform_hybrid/experiments/results/claim_to_run/` maps each published claim to its supporting run identifiers, the W&B URLs where they can be inspected, whether the model named in the paper matches the model actually trained, the seed and step counts, the held-out metric, and an evidence tier. It holds **18 claims** carrying **713 claim-to-run links** — a count of links, not of distinct runs, and the 368 links on P7-C1 are the `zvf-audit` project count rather than 368 separable experiments.

P1–P8 claim-to-run ledger. Tier vocabulary: A = ≥5 seeds and ≥100 steps; B = 3–4 seeds and 50–99 steps; C = completed/descriptive; R = deterministic

resource or schema claim; X = unlinked, contradicted, or provenance-conflicted.

Table A.5: Semester-4 P1-P8 study runs: each claim mapped to its linked runs, stack, model and algorithm.

Claim	Linked runs (representative)	Stack	Model	Algo	G	Dataset	Seeds / steps	Outcome	Tier
P1-C1	20 — `tinker:51a8ef9e…`	Tinker (managed) + W&B	5 GSM8K anchors: Qwen3.5-4B, Qwen3-8B, Llama-3.1-8B-Instruct, Nemotron-120B, DeepSeek-V3.1	—	—	GSM8K	1 per anchor (seed 42) / 20–30	No held-out metric on the exact-linked runs; the claim rests on training-reward traces	C
P1-C2	1 — `tinker:657a920a…` (HF-arbitrated)	Tinker (managed) + HF adapter	`nvidia/NVIDIA-Nemotron-3-Super-120B-A12B-BF16` (MoE, A12B active)	—	—	`frontier_gsm8k`	1 (seed 42) / 20	Peak 0.875, last-10 0.1625, zero-reward step fraction 0.55 (11/20); no held-out metric	C
P2-C1	80 local rows pooled to 23 cells	Local analysis pool (no W&B)	Heterogene~~ous~~; Qwen3-8B at G = 8 among the listed rows		—	GSM8K	up to 3 per source row / 30–600	Mean ZVF versus collapse: Spearman 0.56, point-biserial 0.62 (n = 23)	C
P2-C2	80 (as P2-C1)	as P2-C1	as P2-C1	—	—	GSM8K	as P2-C1	Spearman $\rho = 0.27$, bootstrap 95% CI [−0.37, 0.88] (n = 23)	C

Claim	Linked runs (representative)	Stack	Model	Algo	G	Dataset	Seeds / steps	Outcome	Tier
P3-C1	12 — `wandb:rlvr-openings/l22x3tca`	W&B-managed open trainer; no Tinker identity	Qwen/Qwen3.5-4B	—	2, 4, 8, 16	— (saturated arithmetic regime; dataset not recorded)	3 per G (seeds 0, 1, 2) / 8	Mean held-out Δ: G=2 +0.0000, G=4 +0.0000, G=8 −0.0625, G=16 +0.0000; evaluation size not logged	C
P3-C2	5 — `wandb:zvf-audit-v2/6snvz1qd`	W&B, sampling only	— (G = 32 cells)	—	32	—	5 cells / 0 optimizer steps	No matched G = 4/G = 32 training replication; the comparison cannot be executed	X
P4-C1	6 — `local:drgrpo_gsm8k_cot:grpo:seed42`	Local	Qwen/Qwen2.5-1.5B-Instruct as mapped (abstract says Qwen3-8B) — conflict	GRPO / Dr.GRPO	—	GSM8K CoT	3 per arm (42, 123, 456) / 30	GRPO 0.2017→0.2633 (Δ +0.0617); Dr.GRPO 0.2050→0.2550 (Δ +0.0500); n = 200/seed	X
P5-C1	2 — `wandb:tinker-rl-lab-world-class/xmot42ot`, `…/w83mv3ok`	Tinker-managed versus TRL backend	Qwen/Qwen3-8B-Base versus Qwen/Qwen3-8B — confounded	GRPO / TRL-GRPO	8	GSM8K	1 per backend (seed 42) / 30	Final/last-10 training reward 0.050→0.856; no held-out metric	X
P5-C2	2 — `zvf-colab-experiments/6c7p198f`, `…/l5m9lqij`	Open trainer versus managed stack	Qwen2.5-0.5B-Instruct versus Qwen3.5-4B (declared non-consistent exhibit)	DAPO label versus asymmetric-clip surrogate	—	—	2 open versus 3 managed / not logged (open), 15 (managed)	Mean ZVF 0.00 versus 0.58; no comparable held-out metric	C

Claim	Linked runs (representative)	Stack	Model	Algo	G	Dataset	Seeds / steps	Outcome	Tier
P5-C3	1 — `zvf-colab-experiments/l5m9lqij`	Managed summary	Qwen/Qwen3-4B	F3-5 algorithm labels	—	—	3 per arm (42, 123, 456) / 15	Last-10 reward band 0.710–0.744 (width 0.034); no held-out metric	C
P6-C1	48 registry entries	Registry (schema validation)	N/A	N/A	N/A	N/A	N/A	44/48 entries pass schema validation	R
P6-C2 (new audit)	82 — `tinker:0dfee749…`	Tinker ↔ W&B ↔ HF, three-way	N/A (identity audit)	N/A	N/A	N/A	as logged per underlying run	33/41 exact IDs model-consistent; 5 HF-arbitrated mislinks; 3 unresolved	R
P7-C1	368 — `wandb:zvf-audit/fzforbvg`	W&B only (0 exact Tinker IDs)	Qwen3-4B-Instruct-2507 (113), Llama-3.2-3B (112), Llama-3.2-1B (107), 3 further models (36)	—	4, 8, 16 (also 6)	— (prompt/tokenizer families `llama3`, `qwen3_instruct` recorded, not datasets)	10 logged labels (0–9) / 2–12	ZVF U-shape reproduced; no held-out metric in the W&B summaries	C
P7-C2	1 — `zvf-colab-experiments/ds83rymc`	W&B (Colab)	Qwen/Qwen2.5-0.5B-Instruct	—	—	—	3 (0, 1, 2) in one aggregate / 30	Gradient magnitude versus p(1−p): Pearson r = +0.71	C

Claim	Linked runs (representative)	Stack	Model	Algo	G	Dataset	Seeds / steps	Outcome	Tier
P7-C3	1 — `zvf-colab-experiments/6c7p198f`	W&B (Colab)	Qwen/Qwen2.5-0.5B-Instruct	Adaptive-G versus fixed recipes	—	—	1 per arm / not logged	Adaptive-G held-out Δ +0.575; mean ZVF 0.23; 186 rollouts	C
P8-C1	1 — `local:qp8_fraud:xgboost:seed42`	Local (XGBoost)	Gradient-boosted trees (200 estimators, depth 6)	—	N/A	Synthetic card-fraud: 50,000 rows; 10,000-row held-out split	random_state 42 / 200 estimators	AUC 0.7955 on the 10,000-row held-out split	C
P8-C2	1 — `wandb:tinker-new-research/ek1b2cxn`	Tinker (managed SFT)	Qwen/Qwen3-4B	SFT-row serialisation (no KL, no reference policy)	N/A	Row-to-text serialisations; 500-row positive-enriched held-out subset	1 (seed 0) / 63 mini-batches	AUC 0.482675, accuracy 0.792 (n = 500) — near chance	C
P8-C3	0 — UNLINKED	—	Proposed sensor / scribe / triage architecture	—	—	—	0 / 0	No execution; operational benefit over XGBoost alone not measured	X

Three facts here are load-bearing. First, **no empirical headline row in the P1–P8 ledger reaches Tier A or B** under the statistical-rigor thresholds; the rectified tally is C = 12, R = 2, X = 4, and the registry's only Tier A block belongs to P11, an extension described below. Second, the four Tier X rows are X for specific, checkable reasons: **P3-C2** rests on G = 32 *sampling* cells with zero optimizer steps, so no matched G = 4/G = 32 training replication exists; **P4-C1**'s abstract names Qwen3-8B while every mapped run is `Qwen/Qwen2.5-1.5B-Instruct`; **P5-C1** compares `Qwen/Qwen3-8B-Base` against `Qwen/Qwen3-8B`, so the base checkpoint changes together with the backend and the pair must not be cited as a backend-only causal effect; and **P8-C3** is a proposed architecture with no execu-

tion, hence zero linked runs. Third, of 41 exact Tinker identifiers in the identity audit, 33 are model-consistent with their W&B counterpart and 8 conflict; five conflicts were resolved as W&B mislinks by three-way arbitration against the Hugging Face adapter listing, and three remain unresolved (`0ef59237`, Qwen3.5-4B recorded against a Nemotron run; `78e2d35b`, Qwen3.5-27B; `0dfee749`, Llama-3.1-8B-Instruct recorded against a DeepSeek V3.1 run). Those three `model` fields are contested.

Two column caveats travel with the table. The dataset cell for P7-C1 is `-` because the ledger's only dataset-adjacent field for those 368 runs is a prompt/tokenizer family (`llama3`, `qwen3_instruct`, `gpt_oss_low_reasoning`, `nemotron3_disable_thinking`), not a dataset name. And the P3 sweep here is **not** the Semester-3 sweep of Table A.4: it is a different project (`rlvr-openings`), a different model (Qwen3.5-4B), and it reports deltas of 0.0000 rather than held-out accuracies near 0.99. The two share a group-size grid and nothing else.

The extension ledger added in August 2026 carries five rows besides P11: P10-C1 is Tier R (an algebraic identity, not an empirical result), P10-C2 and P12-C2 are Tier X (placeholder figures and unlinked prospective claims, zero runs), and P12-C1 is Tier R as a lemma and Tier X as an empirical headline.

P11 is the only block in the registry meeting the report's strictest evidence bar: 40 arm–seed units at `local:zvf-program/audit/results/full/{grpo,dapo,gspo,drgrpo,aero}-seed-{11,23,37,53,71,89,107,131}.json`, five arms × eight seeds on one stack (Qwen3-8B, GSM8K, 30 steps, Colab A100, held-out n = 500). Its headline contrast, DAPO versus GRPO, is $\Delta = +0.001$, CI $[-0.0045, +0.00675]$, $\mathrm{MDE}_{80} = 0.01012$, all raw $p > 0.21$, with Benjamini–Hochberg rejecting nothing. The mechanism is visible in the diagnostic: DAPO's mean ZVF is 0.000 against GRPO's 0.693, bought with 3.61× the rollouts (1,734 versus 480) and 1.44× the wall clock. It is the only place in the work where a claim rests on five or more seeds over a hundred or more steps, and it supports a null.

A.3.1 Opening and null-result runs

Chapter 6 reports four null families — a mixed-variance curriculum, a matched token budget, hard-prompt filtering, and step-1 layer selection (§6.4.3) — and each rests on runs that are not in the ledger. They are absent for a checkable reason: the string `rlvr-openings` occurs nowhere in the ledger, and no file in the fourteen-project `wandb_inventory/` covers it, although every Tinker-managed block below is logged to that project under the groups named in the table. The rows are counted from the per-run `name` (or mode) fields of each

artefact's JSON — `results.json` for the curriculum, token-budget, hard-prompt and loss-formulation blocks, `result.json` and its siblings for P1 — not from an inventory API, which is why this cohort is reported as its own total. Three neighbouring blocks add nothing to it: the P3 group-size artefacts are already indexed — `groupsize_zvf/results.json` reproduces the per-group deltas of Table A.5's P3-C1 row exactly, and the second sweep file is discussed below — and the P2, P7 and P8 "openings" (`p2_openings/collapse_baseline_analysis.json`, `p7_controller/sim.json`, `p8_openings/*.json`) are zero-compute re-analyses that launch no runs.

Semester-4 opening and null-result runs (46 runs). Model is `Qwen/Qwen3.5-4B` and dataset GSM8K unless a row says otherwise. All rows are Tinker-managed except the P1 layer-freeze block, which ran on Colab for per-layer gradient access. "Curriculum" means training only on non-collapsed (mixed-variance) groups. Group sizes marked *default* are the launch defaults of the scripts in `platform_hybrid/experiments/openings/`; only the campaign artefact records a group size per run. Δ is the change in held-out accuracy.

Table A.6: Semester-4 opening and null-result runs: the curriculum, token-budget, hard-prompt, loss-formulation and layer-freeze blocks.

Run identifier	Stack / platform	Model	Algo	G	Dataset	Seed	Steps	Outcome / status
`campaign/baseline-G4-s0…s5`	Tinker → W&B `rlvr-openings` (`campaign`)	Qwen/Qwen3.5-4B	GRPO	4	GSM8K	0–5	8	6 runs; mean Δ +0.0111 (0.0000–0.0333)
`campaign/curriculum-G4-s0…s5`	Tinker → W&B `rlvr-openings` (`campaign`)	Qwen/Qwen3.5-4B	GRPO + curriculum filter	4	GSM8K	0–5	8	6 runs; mean Δ +0.0167 (-0.0667 to +0.0667); oversample 3.3–7.2×; zero-loss fraction 0.0
`campaign/baseline-G2-s0`	Tinker → W&B `rlvr-openings` (`campaign`)	Qwen/Qwen3.5-4B	GRPO	2	GSM8K	0	8	1 run; Δ 0.0000; zero-loss fraction 0.7
`campaign/baseline-G8-s0`	Tinker → W&B `rlvr-openings` (`campaign`)	Qwen/Qwen3.5-4B	GRPO	8	GSM8K	0	8	1 run; Δ -0.1333

Run identifier	Stack / platform	Model	Algo	G	Dataset	Seed	Steps	Outcome / status
`campaign/baseline-G16-s0`	Tinker → W&B `rlvr-openings` (`campaign`)	Qwen/Qwen3-4B	GRPO	16	GSM8K	0	8	1 run; Δ +0.0333
`curriculum_opening/baseline`	Tinker → W&B `rlvr-openings` (ungrouped)	Qwen/Qwen3-4B	GRPO	4 *default*	GSM8K	0	8	1 run; Δ +0.0500; held-out 20; zero-loss fraction 0.50
`curriculum_opening/curriculum`	Tinker → W&B `rlvr-openings` (ungrouped)	Qwen/Qwen3-4B	GRPO + curriculum filter	4 *default*	GSM8K	0	8	1 run; Δ +0.0500; held-out 20; oversample 4.81×; zero-loss fraction 0.00; superseded, see below
`token_budget/baseline-b30000-s0…s2`	Tinker → W&B `rlvr-openings` (`token-budget`)	Qwen/Qwen3-4B	GRPO	4 *default*	GSM8K	0–2	30k-token budget	3 runs; mean Δ +0.0278; no groups skipped
`token_budget/curriculum-b30000-s0…s2`	Tinker → W&B `rlvr-openings` (`token-budget`)	Qwen/Qwen3-4B	GRPO + curriculum filter	4 *default*	GSM8K	0–2	30k-token budget	3 runs; mean Δ +0.0278; 11–20 groups skipped per seed
`hard_curriculum/baseline-hard-s0…s2`	Tinker → W&B `rlvr-openings` (`hard-curriculum`)	Qwen/Qwen3-4B	GRPO	6 *default*	GSM8K hard-learnable pool (40 of 300 probed)	0–2	10	3 runs; mean Δ +0.0278; zero-loss fraction 0.1–0.2
`hard_curriculum/curriculum-hard-s0…s2`	Tinker → W&B `rlvr-openings` (`hard-curriculum`)	Qwen/Qwen3-4B	GRPO + curriculum filter	6 *default*	GSM8K hard-learnable pool (40 of 300 probed)	0–2	10	3 runs; mean Δ 0.0000; zero-loss fraction 0.0
`p4_surprise/{sum,mean,surprise}-s0…s1`	Tinker → W&B `rlvr-openings` (`p4-surprise`)	Qwen/Qwen3-4B	GRPO, loss variant	4 *default*	GSM8K	0–1	8	6 runs; mean Δ +0.1250 (sum) / 0.0000 (mean) / +0.0417 (surprise)

Run identifier	Stack / platform	Model	Algo	G	Dataset	Seed	Steps	Outcome / status
`p1_layerfreeze/result.json`	Colab L4	Qwen/Qwen2.5-1.5B-Instruct	(gradient probe, no optimizer steps)	—	—	—	5	1 run; step-1 predicts final top-layer set at overlap 1.0; top-25% concentration 0.476
`p1_layerfreeze/scaled_4seed_result.json`	Colab L4	Qwen/Qwen2.5-3B-Instruct	GRPO	4	GSM8K (24 problems)	0–3	10	4 runs; overlap 0.0833 ± 0.0481 — approximately chance; concentration 0.3908
`p1_layerfreeze/freeze_flop_result.json`	Colab L4	Qwen/Qwen2.5-1.5B-Instruct	GRPO (step-1-frozen arm)	—	—	0–1	—	4 runs, full and frozen arms; frozen arm gains 0.0000 against 0.0625 at 60.7% of the parameter count
`p1_layerfreeze/emergence_result.json`	Colab L4	Qwen/Qwen2.5-1.5B-Instruct	GRPO (step-1 layer selection)	—	—	0–1	—	2 runs; mean step-1 overlap 0.643

Three provenance notes belong with this table, because two of its blocks carry superseded siblings and one carries a refuted claim. First, `campaign/FINDINGS.md` describes a nine-run, three-seed version of the campaign on a twelve-prompt held-out set with arm means of $+0.028$ and -0.028; `campaign/results.json` holds fifteen runs over six seeds on a thirty-prompt held-out set ($0.8667 = 26/30$), and the table follows the per-run file. The FINDINGS warning that `zero_loss_frac` was invalid (an empty-metrics-dict bug, "since fixed") belongs to the nine-run batch, not to that file. Second, the P1 layer-freeze block has the same shape: the two-seed 3B pass (overlap 0.1111 ± 0.0000) is superseded by the four-seed pass (0.0833 ± 0.0481), and both sit near chance, so the toy-scale overlap of 1.0 must not be read as a transferable layer-selection signal. Third, the two single-seed

training blocks flagged SUSPECT by independent verification (kimi, 2026-07-06) are superseded by the multi-seed blocks above: the curriculum opening's +0.05 (campaign curriculum and baseline arms separate by 0.006 over six seeds, at 3.3–7.2× oversample cost), and a single-seed G = 4 gain of +0.125 in `p3_groupsize/sweep_results.json` — the campaign's G4 seed-0 baseline gains 0.0000, not 0.125, and that sweep file retains only two of its twelve entries, so the W&B group is its record.

A.4 E1–E14 held-out campaign lanes

The fourteen evaluation lanes are not training runs; they are evaluations of a single frozen actor, and they carry the registry's strictest receipt discipline. The actor is base `Qwen/Qwen3.6-35B-A3B` (commit `995ad96e…`) with adapter `arvindcr4/pavlov-portfolio-qwen36-seed809-stepfinal-tinker-cf0ad8c1…` (commit `64444133…`), bfloat16, trained on Tinker, run seed 809 taken from the adapter name. The training step count behind that adapter survives in no artefact and is shown as "—"; group size and algorithm are properties of the single actor rather than of any lane, and are shown as "—" rather than repeated. Terminal states are from the ledger of record, `outputs/PES_Phase2_Review_2026-09-12/finish/Pending_Experiments.md`; all scores are from `outputs/E1_E14_FINAL_RESULTS_2026-09-19.md`.

E1–E14 campaign lanes.

Table A.7: E1-E14 held-out campaign lanes with their run identifiers, stacks, platforms, models, algorithms and group sizes.

Lane / run identifier	Stack / platform	Model / actor	Algo	G	Dataset / suite	Seed	Steps	Outcome / terminal state
E1 SWE-bench Pro	native harness, Tinker actor	Qwen3.6-35B-A3B + adapter	—	—	SWE-bench Pro	809	—	2/731 = 0.274% pass@1 (97.54% coverage, 713/731 native evals); originals retained

Lane / run identifier	Stack / platform	Model / actor	Algo	G	Dataset / suite	Seed	Steps	Outcome / terminal state
E1 replacement (wave10)	native harness	as above	—	—	SWE-bench Multilingual	809	—	35/300 graded; wave10 v6 sealed then source lost — `REBUILD_READY_LAUNCH_PENDING`
E2 FrontierSWE	native harness	as above	—	—	FrontierSWE	809	—	1/17 tasks; replay normalized 0.8628
E2 replacement	native harness	as above	—	—	CORE-Bench	809	—	45/45 capsule setups, 0 graded — `AMENDMENT_ACCEPTED_LAUNCH_PENDING`
E3 SDAB	—	as above	—	—	SDAB (private bundle)	809	—	no local or paid path — `CLOSED_EXTERNAL`
E4 BankerToolBench	Tinker→Modal bridge	as above	—	—	BankerToolBench	809	—	1/100 tasks, recovery 0.3115 — `CLOSED_PARTIAL`
E4 base-model rerun	Modal `/v1/responses`	base model, zero-init LoRA	—	—	BankerToolBench	809	0	100/100 trials, mean reward 0.0, $15.07 of $55.91 cap; measures tool-dialogue collapse
E5 APEX-Agents	native harness	as above	—	—	APEX-Agents	809	—	7/480 native-scored; prefix mean 0.050505

Lane / run identifier	Stack / platform	Model / actor	Algo	G	Dataset / suite	Seed	Steps	Outcome / terminal state
E5 replacement (successor27)	native harness	as above	—	—	Tau3	809	—	20/97 = 20.62% cleaned — `REBUILD_READY_LAUNCH_PENDING`
E6 WebArena	AWS-hosted	as above	—	—	WebArena	809	—	0/812 run; 65 files, IDs 0–811 inventory-verified — `PENDING_QUOTA`
E7 BinaryAudit	native harness	as above	—	—	BinaryAudit (private payload)	809	—	1/46 attempted, verifier reward 0.0, no grade — `CLOSED_EXTERNAL`
E8 LAB-Bench (public)	native harness	as above	—	—	LAB-Bench public, 8 categories	809	—	**1967/1967 COMPLETE**
E8 LifeSciBench (original)	—	as above	—	—	LifeSciBench private	809	—	`CLOSED_EXTERNAL`
E9 MLE-bench	Modal streaming arm	as above	—	—	MLE-bench	809	—	40/75 natively graded (53.33%); suite score null
E9 replacement	AWS runtime builder	as above	—	—	MLDevBench	809	—	0/34 graded; image incomplete — `PENDING_QUOTA`
E10 AgentDojo (benign)	native harness	as above	—	—	AgentDojo benign utility	809	—	**97/97 COMPLETE**
E10 AgentHarm (original)	—	as above	—	—	AgentHarm private	809	—	`CLOSED_EXTERNAL`

Lane / run identifier	Stack / platform	Model / actor	Algo	G	Dataset / suite	Seed	Steps	Outcome / terminal state
E11 VerilogEval	native harness	as above	—	—	VerilogEval (both framings)	809	—	**312/312 = 129/312 pass@1 (41.35%)**; 67/156 completion + 62/156 spec-to-RTL
E12 AppBench	—	as above	—	—	AppBench deployment	809	—	`CLOSED_EXTERNAL`
E13 BALROG	hosted supervisor	as above	—	—	BALROG	809	—	13/255 episodes; 21 never-started candidates — `AMENDMENT_ACCEPTED_LAUNCH_PENDING`
E13 OpenReward (original)	—	as above	—	—	OpenReward held-out games	809	—	`CLOSED_EXTERNAL`
E14 Omni-MATH	native scorer	as above	—	—	Omni-MATH	809	—	4426/4428 accepted (99.95%); official accuracy 2271/4428 = 51.31% — `COMPLETE_TERMINAL_NOTE`
E14 FrontierMath (original)	—	as above	—	—	FrontierMath hosted	809	—	`CLOSED_EXTERNAL`

Two source documents disagree on two lanes, and both observations are retained rather than resolved into one, because the disagreement is itself evidence: the historical snapshot records E7 as "1/28 errored, no grade" against the terminal-state table's "1/46 attempted" — both real observations at different points in the lane's life. Likewise E11's canonical figure is 129/312 = 41.35%; a 129/311 sensitivity figure survives in non-canonical receipts and is not used as the headline.

A.5 Completeness limitations

The registry is complete as an *index* and incomplete as a *dataset*, and the distinction governs how it may be used. Coverage is measured over the 1,662 ledger rows only — the 46 opening runs of §A.3.1 are logged per run in their own artefacts and are not part of this table:

Field coverage across the 1,662 ledgered runs.

Table A.8: Field coverage across the 1,662-run ledger: rows carrying a value against rows blank or UNKNOWN.

Field	Rows with a value	Rows blank or `UNKNOWN`	Notes
`model`	1,632	30	all 30 gaps are W&B rows
`task` (dataset)	544	1,118	only 188 are real dataset names; 356 are tokenizer-family labels (`llama3` 225, `qwen3_instruct` 131)
`algo`	139	1,523	only 139 rows name an algorithm at all
`G`	626	1,036	distribution where known: G=8 → 270, G=4 → 181, G=16 → 151, G=6 → 10, G=32 → 6, G=64 → 5, G=2 → 3
`steps`	628	1,034	bimodal: 5 steps → 318 rows, 30 steps → 207 rows, then a long tail
`headline_metric`	28	1,634	98.3% of runs carry no recorded outcome value

The largest gap is structural: **all 844 Tinker-source rows record `task`, `algo`, `G` and `steps` as `UNKNOWN`.** The Tinker pagination API returns training-run identity, base model, LoRA rank and checkpoint metadata but not the training configuration, so those runs' configurations cannot be recovered from the surviving inventory, and every algorithm label in the registry comes from the W&B half of the ledger. The raw snapshot `tinker_runs.txt` shows the same boundary differently: run ID, base model truncated to "Qwen…", LoRA rank 4, timestamp and status "OK" — no configuration at all.

Four further limits apply.

1. **Aggregate presentation.** This appendix names 105 runs individually: the 4 cross-framework runs of Table A.2, the 20 Modal campaign runs of A.3, the 12 sweep runs (in 4 rows) of A.4, the 46 opening runs (in 16 rows) of A.6, and the 23 E-lane rows of A.7 — plus the 18 claim rows of Table A.5, which index claims rather than runs. The remaining ~1,600 runs are

indexed by identifier and project in `experiment_ledger.tsv`, presented here in aggregate because for most of them fewer than half the required columns have values.

2. **Non-runs inside the count.** Three entries are not training results: `verl-dryrun` and `openrlhf-dryrun` are seeded deterministic fallbacks, and the E4 base-model rerun performs zero training steps; to these add §A.3.1's P1 gradient probe, which takes no optimizer steps at all. They are counted because they are runs in the registry's sense, and flagged so no reader treats them as measurements.

3. **Identity conflicts.** Eight Tinker identifiers were linked to W&B runs whose recorded model did not match; five were resolved against the Hugging Face adapter listing, three remain unresolved, and those `model` fields are contested.

4. **Snapshot disagreement.** The 844/818 split used here, the 949/940 split in the audit manifest, and the "983 runs" implied by `tinker_runs.txt` are three observations of the same platforms at three times. No reconciliation was attempted and none should be inferred.

None of this changes a reported result, because every result in Chapters 6–9 is bound to named artefacts rather than to the registry as a whole. It does bound what the registry can be used for: it establishes provenance and identity, shows that the run population is far larger than the reported claims, and shows that those claims rest on a small, explicitly tiered subset — one Tier A block, a majority Tier C, and four Tier X rows that must not be cited as evidence.

Appendix B

Reproducibility and Provenance

B.1 What this appendix establishes

This appendix records three things that should not be confused with one another. The first is what a reviewer can independently re-run: a pinned container, a pinned dependency set, a headline training command, and the tolerance within which a rerun is expected to land. The second is what the repository verifies about itself without re-running any model: an audit suite driven by a single entry point, and a deterministic ledger check that recomputes the reported arithmetic directly from surviving evidence files. The third is provenance: which parts of this work were produced by the Semester 3 group, which by the individual contribution reported here, and what was lost along the way.

B.2 The pinned runtime

The container definition is `Dockerfile`, built from `nvidia/cuda:12.4.1-cudnn-devel-ubuntu22.04`. It installs Python 3.12 through the deadsnakes archive rather than the distribution default, with the in-file comment that Python 3.12 "matches the frozen runtime validated 106/106". It sets `PYTHONHASHSEED=42` and a fixed `LANG/LC_ALL` of `C.UTF-8`; stamps the build commit into an OCI `org.opencontainers.image.revision` label and a `.build_info` file; fixes `SEED=42`, `WANDB_MODE=offline`, `HF_HUB_DISABLE_TELEMETRY=1` and `OMP_NUM_THREADS=8`; self-checks by importing the core stack at build time; and runs under `tini`.

The authoritative pin set is `requirements.txt`, headed as "pinned to the frozen E1 runtime lock … validated 106/106 on this stack". Its exact pins include `torch==2.7.1`, `numpy==2.2.6`, `transformers==5.5.4`,

`trl==1.2.0`, `datasets==4.8.4`, `accelerate==1.13.0`, `peft==0.19.1`, `huggingface-hub==1.11.0` and `wandb==0.21.0`, with bounded ranges for scipy, safetensors, tokenizers, rliable, math-verify, latex2sympy2-extended and pytest, alongside the framework clients `tinker`, `tinker-cookbook` and `atroposlib`.

The 106/106 figure is the recorded outcome of the frozen-runtime validation, which `execution-notes.md` states as passing the complete pinned pilot/preregistration suite on Python 3.12.13 with the pinned versions listed above, and the focused amendment, protocol, launcher and verifier suite passing 55/55. The same record binds five artefacts by SHA-256 — protocol document, screening manifest, launch manifest, launch fingerprint and source binding — and states that a mismatch in package, Python, A100, FLOP, token or gradient conditions fails closed rather than proceeding.

B.3 The reviewer's entry point

`REPRODUCE.md` is the reviewer-facing document. Its fastest path is the structural smoke test at `platform_modal/scripts/smoke_test.sh`, which runs in about two minutes offline and about eight when a live Tinker key is supplied so that the wire protocol is exercised; the offline path needs no GPU and no account.

The headline result is GRPO on Qwen3-8B over GSM8K for 30 steps at LoRA rank 32, group size 8 and seed 42, recorded as a peak accuracy of 62.5 % with a last-10 average of 34.4 %. Reviewer-side reruns are accepted within ±5 percentage points on the last-10 figure and ±10 points on peak, justified in the same document by the residual non-determinism discussed in §B.10. Per-seed reference values and W&B run identifiers are tabulated for seeds 42, 123, 456 and 789, with seed 1024 marked as a reviewer-only rerun; the canonical multi-seed set is `{42, 123, 456, 789, 1024, 2048, 4096, 8192, 16384, 32768}`. The group-size ablation is expected to reproduce a peak of 50.0 % at G=2, 75.0 % at G=4, 100 % at G=8 and 71.9 % at G=16, with last-10 values of 37.5 %, 52.1 %, 84.4 % and 38.0 %. Five-seed aggregation runs through `utils/stats.py` with rliable intervals and 10,000 bootstrap samples; `utils/verify_results.py` checks a rerun against the recorded expectations at those tolerances.

The cost is stated plainly: about five GPU-hours on a single A100-80GB for the headline run, against roughly 446 GPU-hours for a full paper reproduction. Dataset provenance is tabulated with pinned revisions — GSM8K at `740312a` with 7,473 training and 1,319 test rows, plus MATH-500, HumanEval, ArenaHard

Auto, NoRobots and OpenThoughts3 each at a named snapshot — with exact shas in `ARTIFACT.md` §3.4.

B.4 Where the documentation has drifted from the code

Four discrepancies exist between the repository's prose and its live tree, and a reviewer following the prose literally would hit them. The training driver has moved: `REPRODUCE.md` and `ARTIFACT.md` both invoke `grpo_gsm8k_base.py` at the repository root, and the file now lives at `platform_tinker/grpo_gsm8k_base.py`, though the documented command still works because the CLI in `platform_tinker/tinkerrl/grpo_cli.py` carries a compatibility entry point, `legacy_main`, that translates the documented flag spellings onto the current ones — what changed is the path a reviewer must type, not the flags the documents specify. `ARTIFACT.md` names the smoke test at `scripts/smoke_test.sh` rather than at `platform_model/scripts/smoke_test.sh`, though `REPRODUCE.md` gives the correct path. The pins quoted in prose are stale: `REPRODUCE.md` §9 states that `requirements.txt` pins `transformers>=4.56.2,<4.60.0` and a `datasets<3.2` bound, whereas the live file pins `transformers==5.5.4` and `datasets==4.8.4`, and `ARTIFACT.md` §3.3 quotes a third, older set; similarly, `ARTIFACT.md` describes the environment as Python 3.10 from the Ubuntu 22.04 archive, whereas the `Dockerfile` installs Python 3.12 via deadsnakes and the frozen-runtime record validates on Python 3.12.13. The `ARTIFACT.md` items are explained by that document's frozen status — it records a Semester 3 artefact at commit `9019b5c946f17ffe4761b5f4eac13b86d1e51cc0`, dated 19 April 2026 — but `REPRODUCE.md` §9 is live guidance, not a frozen record, and carries the same defect.

B.5 The audit apparatus

The integrity suite is driven by a single entry point, `platform_local/run_all_audits.py`, which imports a shared result interface from `utils/audit_utils.py` and registers its audits as a tuple of name-and-function pairs. As it stands in this tree, that tuple contains **nine** audits: `claim_issues`, `sync_issues`, `anon_issues`, `strength_issues`, `package_issues`, `workflow_issues`, `export_guard_issues`, `caveat_issues` and `scientific_issues`. Each is implemented by a corresponding `platform_local/` module carrying a `__main__` block, so each can also run on its own; `tests/test_audit_runner.py` asserts that the registered modules and

the platform_local/*_audit.py files are the same set.

Some repository documents give different counts, and the difference is historical. At the blind-review sweep the suite was thirteen audits, registered as a list of script filenames at the repository root and executed as subprocesses; the committed log of that run records METRIC audits_total=13 with ten passing and three failing, and the same sweep states that all seventeen *_audit.py scripts then at the root were executed, with the suite covering thirteen of them. The registry now holds nine audits, all passing: a pruning commit removed obsolete audits, and a later restructuring replaced the subprocess runner with the in-process interface, so the seventeen-script description has narrowed by the same route. No script now prints a metric line directly; the line is constructed in one place, AuditResult.metric_line() in utils/audit_utils.py, and emitted by the nine audit modules plus the runner. A repository-wide scan finds several dozen files matching *_audit.py, but that population is not a suite: it includes unrelated analyses, test modules and a third-party file in the virtual environment.

The output contract is uniform. AuditResult.metric_line() emits METRIC <name>=<issue_count>, so every audit reports one machine-readable line whose right-hand side is the number of issues it found, and an audit passes exactly when that count is zero. render_suite prints each audit's metric line and issue codes, then three suite-level lines — METRIC suite_issues, METRIC audits_total and METRIC audits_passing — before either listing the failing audits by name or stating that all audits are passing. The entry point exits 0 when the suite passes and 1 otherwise, which makes it usable as a gate; the packaging step runs the suite before exporting and refuses to package if it fails. The runner's own test asserts this rendering against a synthetic two-audit suite, checking for METRIC audits_total=2 and METRIC suite_issues=1.

B.6 The deterministic ledger check

Where the suite above audits the submission package's internal consistency, a second instrument audits the campaign's reported figures. outputs/verification/LEDGER_CODE_CHECK_2026-09-19.json, under schema ledger-code-check-v1, records eleven checks of which eleven passed and none failed. Each recomputes a claim from the underlying evidence rather than trusting a summary of it, and the four classes of check prove four different things.

Table B.1: The deterministic ledger check by class, listing each specific check and what it establishes.

Class of check	Specific checks	What it proves
Exact divisions	E1 `2/731`; E11 `129/312`; E11 `67+62=129`; E14 `44 26/4428` and accuracy `0 .5131043831902395`; E9 `40 of 75`	That each reported ratio is the quotient of the two integers the report claims, and that a partition sums to its whole
Receipt presence	E8 eight categories summing to 1967 with every `evaluated == expected_total`; E9 graded-row counts; the strict-completion set	That every claimed unit of coverage has a surviving graded row behind it, and that the set of strictly complete suites is exactly `[E8, E10, E11]`
Sealed-artefact hash equality	E1 v6 sealed request and fixture receipt	That the sealed request and the validation receipt are byte-identical to the artefacts recorded at sealing time
Lost-source absence probes	`codex_run_execution_sources_absent`	That four `.codex-run` paths are genuinely absent, so a later re-creation cannot be mistaken for the lost original

The distinction matters under questioning. An exact division proves the arithmetic is not a rounding convenience or a transcription error; it cannot prove the underlying evidence is sound. A receipt-presence check proves coverage claims are backed by artefacts; it cannot prove those artefacts were produced correctly. A hash-equality check proves identity between two artefacts — here, between what was sealed and what was later validated — and is the strongest of the four because it does not depend on re-running anything. An absence probe proves a negative: it prevents a later reconstruction from being passed off as the original. Two instances show the classes in use. The E1 records show the sealed request's

SHA-256 beginning `7adc6e2616f61bc7` matching `validation_v6.json`, with the fixture receipt matching on the same basis. The E9 records show 41 graded rows resolving to 40 unique competitions against a denominator of 75 in the streaming arm and exactly one valid grade in the merged-VLLM arm, which is the check that confirms the two serving arms were never pooled.

B.7 The receipt model

A receipt in this campaign is a durable JSON artefact binding an outcome to the inputs that produced it, and the binding carries the evidential weight rather than the file's existence. The clearest statement of the principle is the receipt-bundle auditor's docstring, which records that the campaign and the receipt builders are kept deliberately separate from the auditor, and that the auditor "never imports a provider SDK, contacts a service, launches a job, or treats `status`/`online`/a URL/ID alone as provenance". The auditor reconstructs contract coverage, receipt predicates, canonical hash rules, budget caps and cross-artefact bindings, carrying its expectations as constants rather than reading them from what it checks — suite counts of 12 training, 14 primary-evaluation and 26 total, and budget constants of 18.00 USD maximum, 16.50 USD operational cap and a 1.50 USD safety reserve — so a receipt set that quietly loses a suite fails the audit rather than redefining it.

Receipts bind four dimensions of a run: protocol, runtime, source and event order. The frozen-runtime record applies that discipline to the runtime dimension alone, binding five artefacts — protocol document, screening manifest, launch manifest, launch fingerprint and source binding — by SHA-256 and failing closed on any mismatch. The judgment layer for subjectively scored questions follows the same pattern: one durable receipt per judgment binds the state bytes, the question text and the raw answer, and battery execution is gated on a health check rather than the service's availability, so a degraded answer cannot be laundered into a valid one. Where a lane could not be scored, its terminal state is recorded in the ledger of record with a named reopen condition, so the absence of a result is itself receipted.

B.8 Evidence loss in the finish era

One loss of evidence occurred and is recorded rather than concealed. During post-review disk cleanup the untracked `.codex-run/` working tree was deleted. Four

paths were confirmed absent by the absence probes described above: `finish_20260912/e1_recovery_v6/bounded_sdk.py`, holding the E1 wave10 repair code; `finish_20260912/e9_completion/restored/`, holding 1,613 hash-verified restored dependency sources from the OpenHands and Composio forks used to build the MLDevBench runtime; `finish_20260912/e6_continuation/checks_v1.py`, an offline receipt validator the reconciliation record describes as structure-only and of low impact; and `public_native_improve_v2/e1/metadata_wave08.py`, the driver for the E1 multilingual run that graded 35 of 300 tasks.

The loss does not invalidate any figure reported in this thesis, for three reasons that hold independently. First, none of the four paths is the source of any reported score: the E1 wave10 repair and the E9 runtime build are execution sources for lanes that remain unlaunched, whose status is `REBUILD_READY_LAUNCH_PENDING` and whose values appear in no results table, and the E1 multilingual figure of 35 of 300 graded is reported as an incomplete replacement-scope state rather than as a suite score. Second, the sealed requests, validation receipts, regression fixtures and the 10-of-10 targeted-test log from the lost E1 code all survive, so a faithful re-implementation is constrained by surviving fixtures rather than free to invent behaviour; that re-implementation has since landed under a tracked path and reproduces the lost suite's record exactly, four process-group regressions and six cleanup regressions. Third, the loss is confined to finish-era iteration code: the replacement-lane native runners survive in tracked locations, including a 515-line multilingual runner carrying no reference to the deleted tree. The policy adopted in response — that execution sources live in tracked repository paths and never in untracked scratch trees — is the durable outcome of the incident.

B.9 The Semester 3 and Semester 4 provenance boundary

The boundary between inherited and individual work is a git tag. `capstone-final-2026-04-25` resolves to commit `21a99ef766543a1ca86bfbad188445d3596ea73c`, dated 23 April 2026, whose subject line records the final metric-separation rewrite that segregated training rewards, held-out capability and proxy diagnostics. Two hundred and fifty commits precede that tag. A reviewer can reproduce the state at the boundary by adding it as a worktree, `git worktree add ../tinker-rl-lab-sem3 capstone-final-2026-04-25`, which materialises the Semester 3 tree without disturbing the current checkout.

The Semester 3 group comprised six students — Arvind C R, Sandhya Jeyaraj,

Madhu Kumara L, Mohammad Rafi, Dhruva N Murthy and Arumugam Chetty K — under guides Anwesh Reddy Paduri and Narayana Darapaneni. Their work is the multi-framework benchmark infrastructure and the initial GRPO baseline experiments. Semester 4 was carried out individually under Ramesh Prakash Guledgudd, and the declaration facing this report assigns to that contribution the research direction, the experimental expansion, the statistical analyses, the P1–P8 paper series and the E1–E14 campaign. The solo-authorship commit `282e6b4f0f158de11b3f5bb4656e2d871f68f5c0`, dated 7 July 2026, records that division at paper level. The provenance record states that 584 commits and 5,050 changed files separated the boundary from the tree as it then stood, and that the eight paper PDFs were compiled at commit `006e90a38e4dabe2d0b4778be01dc7b27f2d3462` on 9 July 2026, one recompiled the following day after a findings-ledger rename. Those counts carry an as-of date rather than describing the current tree, so a reviewer recomputing them will obtain a larger number.

One consequence of this history shapes the repository's structure: the implementation remains shared, with experiment drivers, result tables, figures, LaTeX sources and audit scripts referencing one another across the boundary, and the provenance record is explicit that this is why the code is not split into separate trees. What the boundary separates is authorship of the research contribution, not access to the machinery. Two reporting rules follow and are enforced throughout this report: original-contract and replacement-scope results are never pooled, with no cross-suite aggregate, which the ledger check independently verifies through the arm-separation and component-split checks; and no replacement benchmark's score is presented as the score of the benchmark it replaces, with every substitution named as such at its point of use.

B.10 Residual non-determinism and the limits of reproduction

Not everything here is reproducible in the strict sense, and the tolerances exist because of it. Three sources of residual non-determinism survive even with fixed seeds and deterministic cuDNN settings: remote LoRA training is not bit-deterministic across scheduler decisions, with runs replayed months apart drifting by two to four percentage points on GSM8K last-10, documented through three reruns of the same seed-123 configuration; some matmul operations on H100 and A100 hardware introduce non-determinism through CUDA and cuBLAS; and sampling at temperature 0.8 with top-p 0.95 executes in a remote virtual machine, so a

fixed seed does not fix the sampled trajectory. The record states the consequence itself: run-to-run variance on this setup is of the same order as the effect reported in the paper's PPO-versus-GRPO comparison, which is why that comparison is reported as statistically indistinguishable rather than as a difference.

Three limits remain. Lanes blocked on external access, cloud quota or authorisation cannot be reproduced from this repository, because the blocking resource is not in it. The evidence-loss paths cannot be reproduced in their original form, only re-implemented against surviving fixtures, and that distinction should be preserved in any future citation. And figures originating at the Semester 3 boundary describe the tree at that tag, so reproducing them requires the worktree checkout rather than the current HEAD.

Appendix C

Notation, Symbols and Abbreviations

C.1 Scope and convention

This appendix records the symbols and abbreviations used across this report and the papers behind it, as the sources actually use them rather than as a house style would prefer. Where a symbol carries more than one meaning across documents, both are listed and the collision flagged, because a reader carrying one definition into another chapter will misread a figure.

C.2 Symbols

Table C.1: Notation: symbols used in this report, their meaning, and any divergence in how the underlying sources use them.

Symbol	Meaning	As used in the sources
G	Group size: completions sampled per prompt	Headline runs $G = 8$; sweeps $G \in \{2, 4, 8, 16\}$
K	Rollouts per group in the ZVF formalisation; **separately**, successes within a group	Elsewhere $K = \sum_i R_i \sim \text{Binomial}(G, p)$
B_t, $\mathcal{P}_t$	Batch at optimizer step t; prompts at step t	Competing denominators of ZVF

Symbol	Meaning	As used in the sources
p, $\hat{p}$	Latent success probability; sample rate K/G	Bernoulli(p) model; $\hat{p}$ centres the advantage
r_i, R_i	Scalar reward of completion i; binary reward	Verifiable reward; derivations appendix
$h_G(p)$	$p^G + (1-p)^G$	Bernoulli null for ZVF
ZVF_t, GU_t	Zero-Variance Fraction: share of identical-reward groups; Gradient Utilization $1 - \mathrm{ZVF}_t$	Principal diagnostic and its complement
A_i	Advantage of completion i	Unnormalised $R_i - \hat{p}$; normalised $(r_i - \mu_g)/(\sigma_g + \varepsilon)$; clipped Dr.GRPO
μ_g, σ_g	Group mean and standard deviation of reward	$\mu_g = \frac{1}{G}\sum_j r_j$, population form
π_θ, $\pi_{\theta_{\mathrm{old}}}$, π_{ref}	Trained, sampling and frozen reference policy; per-token importance ratio $\rho_{i,t}$	Ratio against the sampling policy; KL penalty against the reference
$\mathcal{L}(\theta)$	Clipped surrogate plus KL penalty	Main runner omits ratio, clip and reference policy
β	KL coefficient; **separately**, a regression slope	TRL and veRL 0.04; OpenRLHF 0.02; Tinker managed; also a slope in M1/M2
ε	Three distinct meanings — see §C.3	Clip range 0.2; ZVF tolerance 10^{-6}; Adam term 10^{-8}
γ, λ, T	Discount factor; GAE parameter; sampling temperature	0.99; 0.95; canonical 0.8
SI, PTD	Stability index; peak-to-tail drift	Reported in place of direct KL

Symbol	Meaning	As used in the sources
PCD	Pairwise-contrast density	$\mathbb{E}[\hat{p}(1-\hat{p})] = \frac{G-1}{G}\, p(1-p)$
pass@1	Pass rate under the suite's native scorer	E11 $129/312 = 41.35\,\%$; E1 $2/731 = 0.274\,\%$
r, ρ	Correlation coefficients, Pearson and rank	$r = -0.769$, $p = 0.0008$; $\rho \approx 0.65$

C.3 Four collisions worth flagging

The symbol ε carries three unrelated meanings. In the hyperparameter mapping it is the PPO/GRPO clip range, default 0.2, swept over $\{0.05, 0.1, 0.2, 0.3, 0.5\}$. In the ZVF definition it is a numerical tolerance, 10^{-6}, deciding whether a group's sample variance counts as degenerate. In the optimiser configuration it is the Adam epsilon, 10^{-8}. It also appears additively in the advantage denominator, $\sigma_g + \varepsilon$.

K is used two ways: in the formal treatment of ZVF it denotes the rollouts in a group, a synonym for the group size elsewhere called G, whereas in the derivations appendix G is the group size and K is the number of successes — a design parameter in the one case, an outcome in the other. The letter r is similarly the reward, the LoRA rank (swept over $\{4, 8, 16, 32, 64\}$) and the Pearson correlation.

Finally, ZVF is expanded two ways in the sources. The formal definition and the abstract both read "Zero-Variance Fraction", the expansion used here; one figure caption reads "Zero-Value-Filtered". The caption variant is noted only so that a reader meeting it is not misled.

C.4 Abbreviations

Table C.2: Abbreviations and their expansions, with names that no source expands recorded as identities rather than glossed.

Abbreviation	Expansion or identity
GRPO	Group Relative Policy Optimization — critic-free, group-relative
PPO	Proximal Policy Optimization
DPO	Direct Preference Optimization
RLHF	Reinforcement learning from human feedback
IS Loss	Importance-sampling loss, the distillation objective
SFT	Supervised fine-tuning
LoRA	Low-rank adaptation of large language models
KL	Kullback–Leibler divergence; never spelled out in the sources
CoT	Chain-of-thought; spelled out in prose, written "GSM8K CoT" for the variant
ZVF, GU	Zero-Variance Fraction; Gradient Utilization
PCD, SI, PTD	Pairwise-contrast density; stability index; peak-to-tail drift
Dr.GRPO	The advantage-clipping variant
GAE	Generalised advantage estimation; $\lambda = 0.95$
MoE	Mixture-of-experts
BF16	bfloat16, the serving precision for the LAB-Bench, AgentDojo and Omni-MATH lanes
IQM	Interquartile mean, reported via `rliable`
CI	Confidence interval; 95 %, 10,000 bootstrap resamples
W&B	Weights & Biases, the experiment-tracking service

Abbreviation	Expansion or identity
MLE-bench	The suite forming lane E9's original contract; MLE never appears standalone
MATH-500	The mathematics evaluation suite
GSM8K	The grade-school arithmetic suite; 7,473 training and 1,319 test rows
HumanEval	The code-generation suite used for the code arm
ArenaHard, NoRobots, OpenThoughts3	Dataset names used as-is, each at a pinned revision
MiniWoB	Browser micro-task suite used for the proxy smoke test
SWE-bench (Pro, Multilingual)	The repository-repair suite forming lane E1
LAB-Bench, AgentDojo, VerilogEval, Omni-MATH, BALROG, WebArena, APEX-Agents, CORE-Bench, Tau3, MLDevBench, SDAB, BankerToolBench, BinaryAudit, LifeSciBench, AgentHarm, WebBench, AppBench, OpenReward Games, FrontierSWE, FrontierMath	Suite names used as names, with no expansion in any source; see §C.5
Tinker	The remote training and sampling API used to train the adapter
TRL	HuggingFace's LLM post-training library, in both framework rosters
SkyRL, verl (veRL), OpenRLHF, Atropos	Launchers in the cross-launcher roster; only Tinker and TRL produced completed runs, the veRL and OpenRLHF entries being dry-run placeholders

C.5 The E1–E14 lane identifiers

Each lane's short code names one original benchmark contract and is not a rank or a score. Lanes E8, E10 and E14 each hold two states at once — a completed

replacement scope and a `CLOSED_EXTERNAL` contract not released — so the code alone does not say which scope a figure comes from; the terminal state and scope name do.

Table C.3: The E1-E14 lane identifiers, their original contracts, replacement scopes and terminal states.

Lane	Original contract	Replacement scope	Terminal state
E1	SWE-bench Pro	SWE-bench Multilingual	REBUILD_READY_ LAUNCH_PENDING
E2	FrontierSWE	CORE-Bench	AMENDMENT_ ACCEPTED_LAUNCH_ PENDING
E3	SDAB (private bundle)	—	CLOSED_EXTERNAL
E4	BankerToolBench	—	CLOSED_PARTIAL
E5	APEX-Agents	Tau3	REBUILD_READY_ LAUNCH_PENDING
E6	WebBench	WebArena	PENDING_QUOTA
E7	BinaryAudit	—	CLOSED_EXTERNAL
E8	LifeSciBench	LAB-Bench (public split)	COMPLETE (public)
E9	MLE-bench	MLDevBench	PENDING_QUOTA
E10	AgentHarm	AgentDojo (benign utility)	COMPLETE (benign)
E11	VerilogEval	two native framings (no replacement)	COMPLETE
E12	AppBench	—	CLOSED_EXTERNAL
E13	OpenReward Games	BALROG	AMENDMENT_ ACCEPTED_LAUNCH_ PENDING
E14	FrontierMath	Omni-MATH	COMPLETE_ TERMINAL_NOTE