BEYOND VALID OUTPUTS: AUDITING SUCCESS PRESERVATION IN TEST-TIME LLM FUSION

Anonymous authors
Paper under double-blind review

ABSTRACT

Combining specialist language models without further training does not guarantee preservation of their existing successes. We study this question through an empirical comparison of frozen expert groups, pairing native task accuracy with retained, lost, and additional successes. WordGame provides deterministic text-state-machine tasks and a public task-to-expert mapping. On its 768-input development panel, syntax constraints increase parseable outputs from 256 to 754, but final successes only from one to two. A separate, frozen 96-input confirmation with the same specialist bank reproduces a large weighting-policy gap: fixed task-aware weighting solves 93 inputs and dynamic weighting 46, despite all responses being parseable and uncapped. A four-policy contrast recovers 93 successful cases when the main weight is fixed at 0.75; equalizing auxiliary weights while retaining the dynamic main weight recovers 73 cases. These are effects of complete generation policies, including their fallback behavior, rather than universal evidence against dynamic fusion. With common-parent math, coding, and instruction specialists, EMFuse, CoRE, and Mean solve 5–15 code problems outside the saved specialist union yet underperform the coding specialist by 3.75–6.75 percentage points. Together, these results show that admissible outputs and additional coverage can coexist with poor preservation. The study identifies concrete policy-dependent tradeoffs without proposing a new fusion algorithm or imposing a universal method ranking.

1 INTRODUCTION

Alongside general-purpose language models, domain-specialized models fulfill task-specific requirements while retaining their versatility. Independently trained expert models and their modular reuse in multi-expert systems further illustrate this direction (Tan et al., 2025; Li et al., 2022). However, their growing availability poses a fundamental question that goes beyond single-model improvement: how can we harness the collective capabilities of diverse specialists while reducing the risk of losing their individual competencies?

Assembling multiple specialists and leveraging them effectively are two separate challenges. Existing approaches operate at different levels, including jointly trained expert modules within a single architecture (Shazeer et al., 2017), routing among complete models (Ong et al., 2025), test-time distributional fusion (He et al., 2026), and offline weight averaging (Wortsman et al., 2022). In this paper, we specifically study combination after specialist formation: source-model assets remain fixed, and no additional collaboration module is trained. Our primary focus is inference-time output combination. We use task-informed selection and fixed-recipe parameter merging as reference baselines; accordingly, this work is a comparative empirical study rather than a proposal for a new fusion algorithm.

Training-free fusion is already supported by established methods. PackLLM determines or optimizes model weights using input perplexity (Mavromatis et al., 2024); EMFuse combines model outputs through additive energies (He et al., 2026); and CoRE uses model- and token-level confidence and consistency (Zeng et al., 2026). We investigate how these choices impact the retention of correct predictions already achieved by individual specialists. Aggregate accuracy alone does not reveal which successes are preserved, which are lost, or whether the combined model succeeds on examples outside the saved expert-success set. Negative flips can occur even when aggregate perfor-

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

mance improves (Yan et al., 2021). To study these behavioral shifts, we evaluate frozen specialists under autoregressive output combination with executable task-level scoring.

Our central question is: *when specialists are combined, which of their observed successes are preserved, which are lost, and what additional successes are obtained?* WordGame provides a controlled suite to evaluate separately acquired specialist behaviors under matched input conditions. Its stack, rewriting, and grid tasks have deterministic reference executions, allowing output admissibility, terminal-state correctness, and reported-transition validity to be examined separately. Public family labels provide a task-informed selection reference without revealing correct answers. We complement this controlled setting with published math, coding, and instruction-following specialists derived from a common SFT checkpoint, evaluating that checkpoint separately on external tasks (Wang et al., 2026). These two settings test the scope of the observations rather than isolate the causal effect of their different training histories.

Our contributions address three aspects of preservation, combining development evidence with a frozen 96-input confirmation using the same specialists:

- **Executable preservation analysis.** WordGame separates output format, final-state correctness, and reported transitions, with a public task-informed expert reference. Matched success accounting is an analytical tool, not a new metric.
- **Independent-input confirmation and policy separation.** Fixed and dynamic weighting solve 93 and 46 of 96 inputs with identical expert assignments and no parsing or output-cap failures. A four-policy contrast separates main-weight protection from auxiliary allocation, subject to fallback and own-prefix effects.
- **Complementary external evidence.** Successes outside the saved specialist union can coexist with net losses to a deployable expert; equal scores can conceal different successful inputs. These observations support policy-specific conclusions rather than a universal ranking.

2 Related work

Test-time distributional combination constructs a generation policy from existing models without necessarily training an additional collaboration module. PackLLM calculates or optimizes weights using input perplexity, while EMFuse formulates policy fusion through additive energies (Mavromatis et al., 2024; He et al., 2026). CoRE adjusts contributions through model- and token-level consistency and confidence; SAFE selects ensembling positions by using tokenization alignment and predictive consensus, with probability sharpening to address excessive smoothing (Zeng et al., 2026; Yun et al., 2026). These studies already address disagreement, compatibility, and generation degradation. We examine their established rules and identified adaptations through the correctness they retain.

An adjacent approach selects experts or learns their collaboration. RouteLLM trains request-level routers with preference data; DLLG learns token-level fusion weights from response-level supervision; FusionRoute augments expert selection with a trainable complementary generation signal (Ong et al., 2025; Li et al., 2026; Xiong et al., 2026). These mechanisms operate at different granularities and do not necessarily retrain the source experts. Our comparison excludes training a new collaboration module after specialist formation, while allowing disclosed input-dependent computation, test-time weight optimization, and development calibration. Combining experts after a relevant main expert has been specified is distinct from identifying that expert without a task prior.

Parameter merging constructs a deployable model with existing weights before generation starts. Model Soups studies averaging among models fine-tuned with different hyperparameter configurations; TIES trims task updates, resolves signs, and merges updates agreeing with the selected signs (Wortsman et al., 2022; Yadav et al., 2023). Wang et al. (2026) compare joint multi-domain reinforcement learning with separate specialization followed by merging or multi-teacher on-policy distillation from a common initialization. We instead begin with fixed specialists, compare inference-time output combination, and include fixed-recipe Mean and TIES references. The published common-parent specialists provide a controlled source relationship within our external group, not a guarantee of equal training budgets or compatible behavior.

Behavioral preservation specifically examines whether individual specialists maintain their previously demonstrated correctness after combination. Positive-Congruent Training studies negative flips, where examples correctly solved by a reference model become incorrect under a replacement model (Yan et al., 2021). Prior work sharing a common parent checkpoint also evaluates instance-level gains over the SFT baseline alongside an oracle-style reference obtained by taking the coordinatewise maximum across specialists (Wang et al., 2026). Unlike gain-oriented analyses, our analysis focuses on the absolute success union and its associated losses after combination. The four-cell accounting and set decomposition used here are analytical tools rather than new evaluation metrics. Output admissibility represents a separate evaluation dimension. DOMINO and Grammar-Aligned Decoding study subword alignment and distributional fidelity when enforcing grammatical constraints (Beurer-Kellner et al., 2024; Park et al., 2026). In contrast, we isolate syntactic admissibility from functional task success and focus our evaluation on emitted execution records rather than hidden internal reasoning.

Executable evaluation has established foundations. Learning to Execute studies neural execution of short programs, and Neural Networks and the Chomsky Hierarchy uses structured computational tasks including stack manipulation (Zaremba & Sutskever, 2015; Delétang et al., 2023). Show Your Work supervises intermediate computations and evaluates program outputs and traces; CLRS-Text represents algorithmic trajectories as text (Nye et al., 2021; Markeeva et al., 2024). SymBench, Reasoning Gym, and Karel-based program learning provide precedents for symbolic tasks, procedurally generated verifiable training, and grid-program induction (Chen et al., 2025; Stojanovski et al., 2025; Devlin et al., 2017). CCB studies depth-dependent reliability and final-answer/trace discrepancies rather than retention of a fixed specialist's item-level successes (Chapra et al., 2026).

WordGame organizes these task traditions around a different experimental unit: fixed specialists and combination policies evaluated on the same inputs. Deterministic reference executions connect available specialist successes to the requirements that combined outputs satisfy or violate. A public family-to-expert mapping separates task-informed selection from behavior preservation after combination; different transition mechanisms test whether observations are confined to one state representation. Thus the suite supports controlled specialization and executable retention analysis, rather than a claim that trace supervision is new, that the families are statistically independent, or that a single question requires all three skills. Detailed rule definitions and provenance appear in Appendix A.

3 Problem setup

3.1 Fixed specialists and complete strategies

Within an expert group, let $\mathcal{M} = \{m_{\theta_1}, \ldots, m_{\theta_n}\}$ denote the source specialists. Each group studied here has three specialists; the external SFT parent model m_{θ_0} is evaluated separately and is not automatically included in $\mathcal{M}$. Source parameters remain unchanged during comparison. "Training-free" refers to the combination stage, not specialist training, test-time computation, or development-set calibration. A fixed merging recipe may construct another parameter model from these assets; this does not turn parameter merging into output-distribution fusion.

For input x, a fixed template T and verified tokenizer τ produce $z_x = \mathrm{Enc}_\tau(T(x))$. Before token t, fusion strategy $\mathcal{F}$ uses

$$h_t^{\mathcal{F}} = (z_x, y_{<t}^{\mathcal{F}}), \qquad p_{i,\mathcal{F}}^t(v) = p_{\theta_i}(v \mid h_t^{\mathcal{F}}), \quad v \in \mathcal{V}. \tag{1}$$

All experts within a fusion strategy share its actual generated prefix. Different strategies may diverge; distributions at the same token index on independently generated histories are not common-prefix observations. Token axes are aligned within each group, not across the two groups. Game transitions and language-model tokens are different units. Templates, special tokens, and output-axis handling belong to the implementation specification.

Let $c_{\mathcal{F}}$ contain fixed choices such as expert order, the public main-expert mapping, support, and fallback rules. Input-derived state is initialized as $a_0^{\mathcal{F}} = A_{\mathcal{F}}(\mathcal{M}, z_x; c_{\mathcal{F}})$ and may subsequently incorporate observed prefix statistics. A general combination interface is $q_{\mathcal{F}}^t = \mathcal{F}(p_{1,\mathcal{F}}^t, \ldots, p_{n,\mathcal{F}}^t; h_t^{\mathcal{F}}, a_t^{\mathcal{F}}, c_{\mathcal{F}})$. Neither initialization nor decisions can access gold answers, future correctness, reference traces, or hidden execution feedback. For example, candidate-independent

nonnegative weights summing to one define logarithmic pooling:

$$q^t_{\mathcal{F},\log}(v) = \frac{\exp\big(\sum_i w_i^t \log p^t_{i,\mathcal{F}}(v)\big)}{\sum_{u\in\mathcal{V}} \exp\Big(\sum_i w_i^t \log p^t_{i,\mathcal{F}}(u)\Big)}, \qquad w_i^t \geq 0, \quad \sum_i w_i^t = 1. \tag{2}$$

This example assumes positive distributions on the common support. Truncation, masks, EOS decisions, and fallback are specified separately. Algebraic equivalence to weighted logits does not establish finite-precision equivalence across implementations, or describe every component of CoRE.

A complete evaluation strategy s returns $\hat{y}_s(x;\xi) = \Gamma_s(x;\xi)$, where ξ denotes any actual randomness. This covers a single model, generation from fused distributions, a merged model, and an input-level selector reusing a matched saved answer. Saved-answer routing is not answer-aware reranking or measured end-to-end online generation. A versioned native endpoint supplies $Y^E_s(x;\xi) = E(x, \hat{y}_s(x;\xi)) \in \{0,1\}$. Unavailable evaluations are distinguished from completed failures. Main-expert information is also explicit: HOME uses a public task mapping, not a per-example correct-answer oracle. Methods receiving the same prompt need not exploit that mapping in the same way.

3.2 Paired retention and additional coverage

For endpoint E, compare reference r and strategy s on their common labeled input set $\mathcal{D}^E_{r,s}$, with size N. Define $n^E_{ab} = \sum_{x\in\mathcal{D}^E_{r,s}} \mathbf{1}\big[Y^E_r(x) = a,\, Y^E_s(x) = b\big]$ for $a, b \in \{0,1\}$. We report all four counts and their denominators, together with

$$\mathrm{Retention}^E = \frac{n^E_{11}}{n^E_{11} + n^E_{10}}, \qquad \mathrm{Harm}^E = \frac{n^E_{10}}{n^E_{11} + n^E_{10}}, \tag{3}$$

$$\mathrm{Rescue}^E = \frac{n^E_{01}}{n^E_{00} + n^E_{01}}, \qquad \Delta\mathrm{Accuracy}^E = \frac{n^E_{01} - n^E_{10}}{N}. \tag{4}$$

Zero-denominator rates remain undefined. A separately generated expert, task-informed HOME, and hindsight answer-selection oracle are different references. Fixed comparison configurations do not retrospectively preregister development decisions. Multiple answers or methods do not create independent training lineages.

Union analysis uses its own common set $\mathcal{D}^E_*$, on which every designated source expert and compared strategy has a valid label. Write $C^E_s = \{x \in \mathcal{D}^E_* : Y^E_s(x) = 1\}$ and $U^E = \bigcup_{i=1}^n C^E_i$. Additional answer coverage is $|C^E_s \setminus U^E|/|\mathcal{D}^E_*|$, and

$$|C^E_s| = |U^E| - |U^E \setminus C^E_s| + |C^E_s \setminus U^E|. \tag{5}$$

This is set accounting, not a new theoretical result. U^E is the coverage of the saved complete answers, not an upper bound on a strategy generating a different response. Rescue relative to HOME can reuse another specialist's success and need not be union-external. Union-external success does not establish new knowledge or within-question skill composition. Gains relative to the parent require its actual matched outputs and remain separate from absolute success coverage (Wang et al., 2026).

4 Experimental design and evaluation

4.1 Expert groups and task panels

WordGame is a controlled, execution-verifiable specialization stress test. DualStack (DS) executes a supplied program on two capacity-five stacks and a mode bit; GlyphRewrite (GR) applies six prioritized rules to a string of length at most twelve with control state and remaining budget; GridFlux (GF) executes a supplied program on a 2–5 by 2–5 glyph grid with a cursor, orientation, and polarity. Every instance requires a complete canonical transition sequence and final state, predicted in one response rather than through interactive planning. The families vary transition mechanisms while sharing serialization, copying, and sequential state tracking. Each input belongs to one family. The public prompt provides the family, format, initial state, and program or budget, but withholds

Table 1: Task-informed main-expert mapping. A check marks the designated main expert, not measured success; every specialist is evaluated on every task within its group.

WordGame specialists

Family	DS	GR	GF
DualStack	✓		
GlyphRewrite		✓	
GridFlux			✓

Common-parent specialists

Task	Math	Code	IF
GSM8K / AIME2024	✓		
IFEval			✓
LiveCodeBench		✓	

operation semantics and the reference trajectory. Simulation provides supervised targets and offline evaluation, not corrective feedback during decoding.

The three WordGame specialists are trained on a common Qwen2.5-7B Base artifact using LoRA adapters with rank $r = 32$ and scaling $\alpha = 64$ (Hu et al., 2021). Each family contributes 8,000 unique training records; DS uses one pass and GF/GR four passes, corresponding to 1,000 versus 4,000 updates. These are not token- or compute-matched budgets (Appendix B.4); their effect on fusion compatibility is not isolated by equal home-family accuracy. The primary development panel contains 768 inputs in 384 complete parent–twin components, with 256 inputs per family. All three specialists are evaluated on all inputs. The earlier 192-input panel is a subset, not independent corroboration. We label these analyses as exposed development evidence. A separate frozen confirmation uses 96 inputs (48 complete components, 32 inputs per family) with the same bank. Selection precedes these results and uses no outcome-dependent expansion; its finite exposure checks and custody limits are documented in Appendix E. The panels are never pooled.

For external evaluation, we reuse published Qwen3-4B math, coding, and instruction-following descendants of one SFT checkpoint, retaining that checkpoint as a separate reference (Wang et al., 2026). The common parent supports within-group comparisons without implying equal domain training budgets or behavioral compatibility. We evaluate GSM8K (Cobbe et al., 2021), IFEval (Zhou et al., 2023), AIME2024, and LiveCodeBench (LCB) (Jain et al., 2024) under the frozen panels in Table 10. The domain main expert is Math for GSM8K/AIME, IF for IFEval, and Coding for LCB; it is not selected using observed benchmark accuracy. PackLLM-opt is evaluated on a nested, matched 318-input supplement (96/96/30/96 across these tasks), not an independent test or a complete new benchmark inventory.

Generation budgets are fixed within each task across compared methods, but differ between tasks. All settings generate one greedy answer per input. Appendix B.3 lists the panel sizes, output caps, context limits, and precision settings.

4.2 Combination conditions and information access

We compare task-informed expert routing (HOME), energy-based test-time fusion (EMFuse), syntax-constrained EMFuse (SynFuse), fixed task-aware weighting (FixFuse), dynamic task-aware weighting (DynFuse), the full CoRE configuration (CoRE), SAFE fusion (SAFE), and parameter merging (Mean and TIES). A separate logarithmic-pooling control, LogPool-0.75, must not be conflated with FixFuse. WordGame EntRoute and NLLRoute select a specialist once from input statistics and reuse its frozen answer. PackLLM-opt uses its declared prompt-based optimization rather than hard NLL routing. Related-work coverage does not imply that every referenced method was run on every panel. Appendix C maps readable names to exact conditions and preserves supplementary controls.

The focused intervention compares FixFuse with DynFuse. Both identify the same home specialist and share per-expert top-10 softmax support, FP32 reductions, $\epsilon = 10^{-12}$, $\beta = 2$, main-EOS short-circuiting, token-string-prefix fallback, and lowest-token-ID tie-breaking. Only the effective model-weighting rule changes. Fixed weights are $(0.75, 0.125, 0.125)$ in home-first order. Dynamic weights use consistency totals divided by entropy plus ϵ, with a main-model mass clamp giving it at least half the normalized weight. The confirmation adds two policies: D75 fixes the main mass at 0.75 but retains the dynamic auxiliary ratio; DEQ retains the dynamic main mass but splits auxiliary mass equally. These complete policies run on their own generated prefixes (Appendix E). CoRE also uses auxiliary token multipliers; comparing it with the fixed control changes more than

one component. The distinct LogPool-0.75 condition uses full-support logarithmic pooling, not this truncated arithmetic combination.

The syntax intervention constrains policy scores after fusion at each decoding step; it is not post-hoc answer repair and supplies no correct semantic values. WordGame EMFuse uses independent expert caches, BF16 backbone execution, FP32 fusion reductions, EOS token ID 151643, and a 4,096-token output cap. External conditions use the frozen FP16 generation configurations and EOS token ID 151645. Equality of mathematical formulas does not imply bitwise equality across execution paths. We do not attribute cross-method performance differences solely to one component when support, fallback, or information conditions differ.

4.3 Native endpoints and statistical analysis

WordGame Parse tests the canonical output contract. Final (F) compares the separately declared terminal state, excluding the halt label. Reported-transition joint (called Historical Joint in the archived records) is $J_{\mathrm{rep}} = F\mathbf{1}[V_{\mathrm{reported}} = 1]$, requiring every reported transition to satisfy the historical family-specific predicate. It does not certify complete coverage, requested actions in every family, correct stopping, or linkage from the last reported state to FINAL. Appendix B.1 specifies these distinctions. Final is the principal correctness endpoint; reported-transition joint is an auxiliary diagnostic.

External tasks retain their native answer or functional-test endpoints. IFEval strict/loose prompt success is binary, whereas instruction accuracy aggregates applicable constraints; its four metrics remain separately reported. LCB answers are tested under the declared isolated scoring policy (Appendix B.5). An output reaching the generation cap remains in the denominator and is evaluated normally. Infrastructure failures are not automatically scored as model errors. The single TIES–LCB scoring-container OOM is explicitly adjudicated as a policy failure and marked †, without rewriting the original 399 normal scoring records.

WordGame intervals use 10,000 paired bootstrap resamples of complete parent–twin components, stratified by family. External binary proportions use Wilson 95% intervals; method differences use 10,000 paired problem resamples. IFEval instruction intervals cluster constraints by prompt where flags were retained; unavailable flags do not justify treating instructions as independent samples. Actual seeds and endpoint coverage are recorded in Appendix B.6. The focal exploratory contrasts are FixFuse versus DynFuse, EMFuse versus SynFuse, FixFuse versus CoRE, and full-panel LCB combinations versus Coding. Remaining contrasts and routing diagnostics are secondary. This development exposition is not preregistration. Separately, the frozen confirmation specifies FixFuse minus DynFuse Final as its primary contrast and reported-transition joint as a companion. Other contrasts, including the four-policy interaction, are unadjusted exploratory analyses. Resampling problems does not remove development-selection bias or quantify variation across independently trained specialist groups.

5 Results

5.1 Available success and what survives combination

We first establish the opportunity available in the saved specialist answers. Table 2 establishes specialization directly: each WordGame specialist solves 232/256 home-family inputs, and every off-family Final cell is zero. Their disjoint success union covers 696/768 inputs and equals public-family HOME. Table 3 shows that neither output fusion nor parameter merging automatically retains this coverage. The external capability matrix has a different structure (Table 4). On the full 400-input LCB panel, the three-specialist union covers 164 inputs, compared with 130 for Coding (Table 6); this full-panel analysis is separate from the matched 96-input comparison. Thus there is additional saved-answer coverage, but no deployable selector is implied by the union.

Newly generated successes do not by themselves establish a net benefit. On the full 400-input LCB panel, EMFuse, CoRE, and Mean respectively produce 5, 11, and 15 successes outside the 164-input specialist union, while losing 66, 60, and 66 successes within it (Table 6). Relative to deployable Coding, their harm/rescue counts are 44/17, 35/20, and 44/27: net differences of -6.75, -3.75, and -4.25 percentage points, with exploratory paired 95% intervals $[-10.50, -3.00]$, $[-7.50, -0.25]$,

Table 2: WordGame specialist capability: Final accuracy (%), 256 inputs per family. Counts and 95% CIs are in Appendix D.

Specialist	DS	GR	GF
DS specialist	90.6	0.0	0.0
GR specialist	0.0	90.6	0.0
GF specialist	0.0	0.0	90.6

Table 3: WordGame strategies on 768 inputs (%). Joint denotes reported-transition joint. Final/Joint show upper/lower 95% CI offsets; full intervals are in Appendix D. Cap denotes scored output-cap hits, not memory overflows.

Strategy	Final	Joint	Parse	Cap
Task-informed routing / fusion				
HOME	$90.6^{+2.3}_{-2.5}$	$87.8^{+2.7}_{-2.9}$	99.9	0.0
LogPool-0.75	$83.3^{+3.1}_{-3.3}$	$79.3^{+3.5}_{-3.5}$	100.0	0.0
FixFuse	$90.8^{+2.5}_{-2.5}$	$87.9^{+2.9}_{-2.9}$	100.0	0.0
DynFuse	$44.4^{+4.0}_{-3.9}$	$39.3^{+4.2}_{-3.9}$	98.8	0.1
CoRE	$90.8^{+2.3}_{-2.5}$	$88.3^{+2.7}_{-2.7}$	100.0	0.0
Fusion without task-aware expert weights				
EMFuse	$0.1^{+0.3}_{-0.1}$	$0.0^{+0.0}_{-0.0}$	33.3	8.1
SynFuse	$0.3^{+0.4}_{-0.3}$	$0.0^{+0.0}_{-0.0}$	98.2	1.8
SAFE	$0.1^{+0.3}_{-0.1}$	$0.0^{+0.0}_{-0.0}$	63.9	0.7
Input-statistic routing				
EntRoute	$74.6^{+3.1}_{-3.3}$	$72.4^{+3.3}_{-3.3}$	82.3	0.4
NLLRoute	$30.6^{+1.4}_{-1.6}$	$28.1^{+2.0}_{-2.1}$	33.7	38.9
Parameter merging				
Mean	$0.3^{+0.4}_{-0.3}$	$0.1^{+0.3}_{-0.1}$	21.0	51.8
TIES	$6.8^{+1.8}_{-1.8}$	$6.8^{+1.8}_{-1.8}$	30.5	4.3

Table 4: External specialist capability on the matched subsets: 96 / 96 / 30 / 96 inputs. Accuracy (%); IFEval is strict prompt and LiveCodeBench is one-answer pass@1. The Parent is evaluated separately from the expert bank. Counts and 95% CIs appear in Appendix D.3.

Strategy	Fixed GSM8K subset	IFEval benchmark	AIME2024	Matched LiveCodeBench
Parent	83.3	28.1	30.0	27.1
Math specialist	83.3	30.2	36.7	17.7
Coding specialist	80.2	29.2	50.0	31.3
Instruction specialist	82.3	28.1	40.0	25.0

Table 5: EMFuse, CoRE, Mean, TIES, and PackLLM-opt on the same matched subsets as Table 4: accuracy (%). The TIES OOM case is outside these subsets. Full-panel results and matched counts/95% CIs appear in Appendix D.

Strategy	Fixed GSM8K subset	IFEval benchmark	AIME2024	Matched LiveCodeBench
EMFuse	83.3	29.2	43.3	24.0
CoRE	84.4	25.0	36.7	25.0
Mean	81.3	28.1	50.0	26.0
TIES	82.3	27.1	36.7	21.9
PackLLM-opt	83.3	28.1	43.3	24.0

Table 6: LCB success accounting on the full 400-input panel (not the 96-input subset). The specialist union has 164 successes; Coding has 130. Columns using these two references are intentionally separate.

	Relative to specialist union				Relative to Coding	
Strategy	Retained	Lost	External	Final	Harm	Rescue
EMFuse	98	66	5	103	44	17
CoRE	104	60	11	115	35	20
Mean	98	66	15	113	44	27

and $[-8.50, -0.25]$. Union-external successes are included in, not added on top of, these rescues. The other benchmarks do not show a uniformly superior combination strategy or the near-zero collapse seen in some WordGame conditions. Table 4 also shows why the source labels should not be mistaken for benchmark rankings: Math ties the Parent on the matched GSM8K subset and leads the three specialists on IFEval; Coding leads on AIME2024 and LiveCodeBench. These observed rankings do not determine the designated main expert, and the 30-input AIME panel warrants caution about rankings. The combination candidates are defined by their documented training domains, not by selecting the best observed score.

5.2 Matched weighting rules have different preservation effects

FixFuse obtains 697/768 Final successes, compared with 341/768 for DynFuse (Figure 2). Relative to the fixed control, the dynamic condition retains 335 successes, harms 362, and rescues six; 65 inputs fail in both. The net difference is -46.35 percentage points, with a paired component-bootstrap 95% interval of $[-50.26, -42.32]$. Relative to HOME instead, the dynamic condition harms 361 and rescues six: changing the reference changes the four-cell counts. The main-expert mapping is shared, so failure to identify the public family cannot alone explain this contrast.

The frozen confirmation reproduces the gap: FixFuse solves 93/96 and DynFuse 46/96, a paired difference of $+48.96$ points (95% CI $[38.54, 59.38]$). Switching to DynFuse loses 47 successes and rescues none. All six conditions are parseable and uncapped. D75 matches FixFuse's token sequences on all 96 inputs; DEQ solves 73/96. The accuracy-scale interaction is -28.13 points (Appendix E), indicating policy-dependent effects of main-weight protection and auxiliary allocation. However, D75 and DEQ use normal pooling on only 1 and 51 generated steps; most steps take the shared fallback. This is a complete-policy contrast, not an isolated token-level mechanism or evidence for a universally optimal main weight.

5.3 Restoring format does not comparably restore task correctness

The syntax intervention (Figure 1) increases Parse from 256/768 to 754/768, an increase of 498 inputs or 64.84 percentage points (paired 95% interval $[63.67, 65.89]$), with no lost parses. Final success increases only from 1/768 to 2/768, and Reported-transition joint remains zero. Thus parser rejection alone does not account for the severe task failure of this fusion condition. Independently, DynFuse produces 759 parseable responses but only 341 correct final states. These observations distinguish admissibility from successful execution; they neither show that syntax control is useless nor eliminate length, representation, or particular token decisions as contributing factors.

5.4 Equal accuracy does not identify the same successful behavior

FixFuse and CoRE both achieve 697/768 Final successes. Yet relative to HOME, their harm/rescue counts are 2/3 and 6/7, and their correctness labels differ on eight inputs. Reported-transition joint is 675 versus 678. Both have only a one-input net Final gain over HOME, with paired intervals spanning zero. These results do not constitute an equivalence test or establish that one replacement pattern is universally preferable.

The matched external supplement gives another example (Appendix D.3). PackLLM-opt and EMFuse tie at 80/96 on GSM8K, 13/30 on AIME2024, and 23/96 on LCB, while disagreeing on 12, 6, and 16 correctness outcomes. In the 96-input LiveCodeBench subset each method uniquely solves

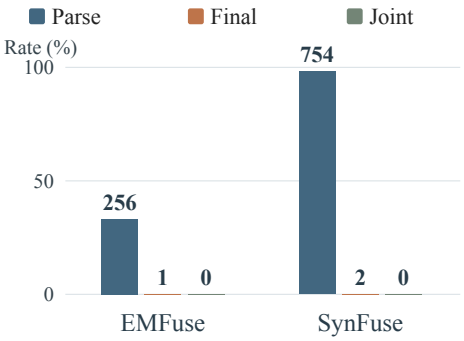


Figure 1: Parseability and correctness on the same 768 WordGame inputs. Numbers are successful inputs; all joint successes are zero. Syntax masking restores format without comparable task recovery.

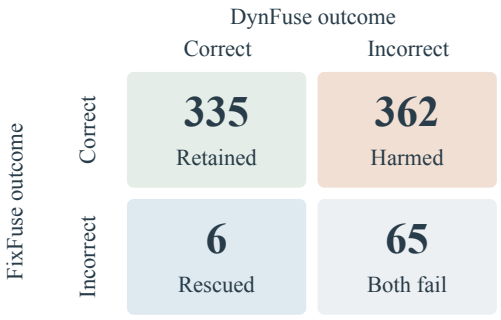


Figure 2: FixFuse → DynFuse paired Final outcomes on 768 inputs. Harmed successes (362) greatly exceed rescues (6); 335 are retained. This compares entire weighting rules.

eight inputs. A tie in aggregate accuracy therefore does not determine which prior successes survive a method substitution. Input-weight and routing diagnostics are reported as secondary observations, not as proof that likelihood-based strategies fail generally.

6 Discussion and Limitations

The expert groups provide complementary evidence, not a controlled comparison of training paradigm or model scale. WordGame's public expert mapping makes it a preservation stress test rather than within-question skill composition. Its development panel is exposed; confirmation tests new inputs with the same bank, not new training seeds. Weighting contrasts include own-prefix divergence and fallback. Saved-answer unions depend on decoding and do not identify latent knowledge. Unequal specialist training budgets may affect calibration despite equal home-family accuracy.

7 Conclusion

Combining fixed specialists without further training requires preserving existing successes as well as obtaining new ones. WordGame shows that restoring output format need not restore task correctness. Independent-input confirmation reproduces the fixed–dynamic weighting gap; the four-policy contrast separates main-weight protection and auxiliary allocation within fallback-equipped generation policies. External code results show that successes beyond the saved specialist union can still accompany larger losses than gains against a deployable expert. These findings support evaluating complete policies through native accuracy and matched success transitions, without establishing a universally best rule, optimal fixed weight, or generalization to new specialist banks.

REPRODUCIBILITY STATEMENT

The evidence accompanying this revision includes the aggregate records underlying the tables and figures, the confirmation statistics and branch counts, their source hashes (Appendix C.2). It supports checking the reported summaries, not a claim of complete end-to-end reproduction. The index distinguishes included files from hash-only references; no public repository or unavailable artifact is represented as released.

AI USE STATEMENT

In this work, we used generative AI tools for sourcing/searching for information and editing a research paper to improve readability. We have not used generative AI tools for generating synthetic data sets, developing theoretical models or conceptual frameworks, or designing research methodology or experiments, and the rest of the required disclosure tasks are not applicable to this work. Additionally, we used generative AI tools for identifying relevant literature and summarizing or analyzing existing literature. We have reviewed all AI-assisted work; specifically, all data collected or sourced with the assistance of AI tools were manually verified and audited before archiving, and all experiments were independently executed, distributed, and confirmed by the authors. We take responsibility for the final content of this work, including text, tables and figures produced with the aid of generative AI.

REFERENCES

Luca Beurer-Kellner, Marc Fischer, and Martin Vechev. Guiding llms the right way: Fast, non-invasive constrained generation, 2024. URL https://arxiv.org/abs/2403.06988.

Shubh Chapra, Dhruv Kumar, Murari Mandal, and Yash Sinha. The complexity ceiling benchmark: A multi-domain evaluation of sequential reasoning under depth scaling, 2026. URL https://arxiv.org/abs/2606.29278.

Yongchao Chen, Yilun Hao, Yueying Liu, Yang Zhang, and Chuchu Fan. CodeSteer: Symbolic-augmented language models via Code/Text guidance. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 8184–8211. PMLR, 13–19 Jul 2025. URL https://proceedings.mlr.press/v267/chen25x.html.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems, 2021. URL https://arxiv.org/abs/2110.14168.

Grégoire Delétang, Anian Ruoss, Jordi Grau-Moya, Tim Genewein, Li Kevin Wenliang, Elliot Catt, Chris Cundy, Marcus Hutter, Shane Legg, Joel Veness, and Pedro A. Ortega. Neural networks and the chomsky hierarchy, 2023. URL https://arxiv.org/abs/2207.02098.

Jacob Devlin, Rudy Bunel, Rishabh Singh, Matthew Hausknecht, and Pushmeet Kohli. Neural program meta-induction, 2017. URL https://arxiv.org/abs/1710.04157.

Kejie He, Yi-Chen Li, and Yang Yu. EMFuse: Energy-based model fusion for decision making. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=6wDp8XRmNI.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models, 2021. URL https://arxiv.org/abs/2106.09685.

Naman Jain, King Han, Alex Gu, Wen-Ding Li, Fanjia Yan, Tianjun Zhang, Sida Wang, Armando Solar-Lezama, Koushik Sen, and Ion Stoica. Livecodebench: Holistic and contamination free evaluation of large language models for code, 2024. URL https://arxiv.org/abs/2403.07974.

Bingnan Li, Zhaoyang Zhang, Xiaoze Liu, Yantao Shen, Shuli Jiang, Shuo Yang, Wei Xia, Zhuowen Tu, and Stefano Soatto. Dllg: Dynamic logit-level gating of llm experts, 2026. URL https://arxiv.org/abs/2606.04378.

Margaret Li, Suchin Gururangan, Tim Dettmers, Mike Lewis, Tim Althoff, Noah A. Smith, and Luke Zettlemoyer. Branch-train-merge: Embarrassingly parallel training of expert language models, 2022. URL https://arxiv.org/abs/2208.03306.

Larisa Markeeva, Sean McLeish, Borja Ibarz, Wilfried Bounsi, Olga Kozlova, Alex Vitvitskyi, Charles Blundell, Tom Goldstein, Avi Schwarzschild, and Petar Veličković. The clrs-text algorithmic reasoning language benchmark, 2024. URL https://arxiv.org/abs/2406.04229.

A. A. Markov. The theory of algorithms. *Trudy Matematicheskogo Instituta imeni V. A. Steklova*, 42:3–375, 1954. URL http://mi.mathnet.ru/tm1178.

Costas Mavromatis, Petros Karypis, and George Karypis. Pack of llms: Model fusion at test-time via perplexity optimization, 2024. URL https://arxiv.org/abs/2404.11531.

Maxwell Nye, Anders Johan Andreassen, Guy Gur-Ari, Henryk Michalewski, Jacob Austin, David Bieber, David Dohan, Aitor Lewkowycz, Maarten Bosma, David Luan, Charles Sutton, and Augustus Odena. Show your work: Scratchpads for intermediate computation with language models, 2021. URL https://arxiv.org/abs/2112.00114.

Isaac Ong, Amjad Almahairi, Vincent Wu, Wei-Lin Chiang, Tianhao Wu, Joseph E. Gonzalez, M Waleed Kadous, and Ion Stoica. Routellm: Learning to route llms with preference data, 2025. URL https://arxiv.org/abs/2406.18665.

Kanghee Park, Jiayu Wang, Taylor Berg-Kirkpatrick, Nadia Polikarpova, and Loris D'Antoni. Grammar-aligned decoding, 2026. URL https://arxiv.org/abs/2405.21047.

Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer, 2017. URL https://arxiv.org/abs/1701.06538.

Zafir Stojanovski, Oliver Stanley, Joe Sharratt, Richard Jones, Abdulhakeem Adefioye, Jean Kaddour, and Andreas Köpf. Reasoning gym: Reasoning environments for reinforcement learning with verifiable rewards, 2025. URL https://arxiv.org/abs/2505.24760.

Zhi-Hao Tan, Zi-Chen Zhao, Hao-Yu Shi, Xin-Yu Zhang, Peng Tan, Yang Yu, and Zhi-Hua Zhou. Learnware of language models: Specialized small language models can do big, 2025. URL https://arxiv.org/abs/2505.13425.

Haoqing Wang, Xiang Long, Ziheng Li, Yilong Xu, Tingguang Li, and Yehui Tang. To mix or to merge: Toward multi-domain reinforcement learning for large language models, 2026. URL https://arxiv.org/abs/2602.12566.

Mitchell Wortsman, Gabriel Ilharco, Samir Yitzhak Gadre, Rebecca Roelofs, Raphael Gontijo-Lopes, Ari S. Morcos, Hongseok Namkoong, Ali Farhadi, Yair Carmon, Simon Kornblith, and Ludwig Schmidt. Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time, 2022. URL https://arxiv.org/abs/2203.05482.

Nuoya Xiong, Yuhang Zhou, Hanqing Zeng, Zhaorun Chen, Furong Huang, Shuchao Bi, Lizhu Zhang, and Zhuokai Zhao. Token-level llm collaboration via fusionroute, 2026. URL https://arxiv.org/abs/2601.05106.

Prateek Yadav, Derek Tam, Leshem Choshen, Colin Raffel, and Mohit Bansal. Ties-merging: Resolving interference when merging models, 2023. URL https://arxiv.org/abs/2306.01708.

Sijie Yan, Yuanjun Xiong, Kaustav Kundu, Shuo Yang, Siqi Deng, Meng Wang, Wei Xia, and Stefano Soatto. Positive-congruent training: Towards regression-free model updates, 2021. URL https://arxiv.org/abs/2011.09161.

Heecheol Yun, Kwangmin Ki, Jung Hyun Lee, and Eunho Yang. When to ensemble: Identifying token-level points for stable and fast LLM ensembling. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=kfPkF2ACDM.

Wojciech Zaremba and Ilya Sutskever. Learning to execute, 2015. URL https://arxiv.org/abs/1410.4615.

Zhichen Zeng, Qi Yu, Xiao Lin, Ruizhong Qiu, Xuying Ning, Tianxin Wei, Yuchen Yan, Jingrui He, and Hanghang Tong. Harnessing consistency for robust test-time LLM ensemble. In Vera Demberg, Kentaro Inui, and Lluís Marquez (eds.), *Findings of the Association for Computational Linguistics: EACL 2026*, pp. 3528–3545, Rabat, Morocco, March 2026. Association for Computational Linguistics. ISBN 979-8-89176-386-9. doi: 10.18653/v1/2026.findings-eacl.182. URL https://aclanthology.org/2026.findings-eacl.182/.

Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. Instruction-following evaluation for large language models, 2023. URL https://arxiv.org/abs/2311.07911.

A WordGame task specifications

A.1 Common contract and notation

The fixed configurations use game version `1.0`, simulator version `0.1.0`, and glyph alphabet $\Sigma = \{\texttt{c}, \texttt{m}, \texttt{q}, \texttt{r}, \texttt{t}, \texttt{v}, \texttt{x}\}$. Glyphs and operation identifiers are textual symbols, not necessarily individual tokenizer tokens. All state components not explicitly updated by an operation remain unchanged. The task specifications below are documentation, not additional inference-time instructions.

For DS and GF, an instance consists of an initial state s_0, a finite program $(u_0, \ldots, u_{L-1})$, and a step limit M. Execution follows $s_{t+1} = T_g(s_t, u_t)$ for $t < \min(L, M)$, where T_g is the family's fixed transition function. GR instead determines its next rewrite from the current state. The reference contains every executed transition and the terminal state with its halt reason. The model predicts this record in a single response; it does not issue actions to an interactive environment.

The public prompt carries `[GAME]`, `[GAME_VERSION]`, and `[TARGET_FORMAT]`, followed by instance fields. Canonical output uses consecutive zero-based `STEP` indices, ordered key–value fields, and one trailing `FINAL` record. DS/GF report the action and full before/after states; GR additionally reports the selected rule and its zero-based match position. Empty stacks and strings are serialized as `_`. In GF, `/` separates rows and `r,c` denotes the cursor. Visual wrapping in the following listings does not introduce logical line breaks into a record. Explanatory prose is not part of the target.

A.2 DualStack (DS)

The state is (A, B, m), with $A, B \in \Sigma^{\leq 5}$ and $m \in \{0, 1\}$. Stacks are written bottom to top, so the rightmost glyph is the top. Write $a \parallel z$ for prefix a followed by top element z; ϵ denotes an empty stack in mathematical notation. Table 7 defines all nine opcodes.

Table 7: DualStack operations. Failed guards leave the entire state unchanged.

Opcode	Meaning	Update
`vek`	Transfer $A \to B$	If $A = a\|z$ and $\|B\| < 5$, set $(A, B) \leftarrow (a, B\|z)$.
`zor`	Transfer $B \to A$	If $B = b\|z$ and $\|A\| < 5$, set $(A, B) \leftarrow (A\|z, b)$.
`lum`	Rotate A top to bottom	If $A = a\|z$ and $\|A\| \geq 2$, set $A \leftarrow z\|a$.
`nax`	Rotate B bottom to top	If $B = z\|b$ and $\|B\| \geq 2$, set $B \leftarrow b\|z$.
`piv`	Copy A top to B	If $A = a\|z$ and $\|B\| < 5$, set $B \leftarrow B\|z$ and preserve A.
`mep`	Delete B top	If $B = b\|z$, set $B \leftarrow b$.
`tal`	Exchange stacks	Set $(A, B) \leftarrow (B, A)$.
`rix`	Toggle mode	Set $m \leftarrow 1 - m$.
`sov`	Mode-dependent update	If $m = 0$, execute `vek`. If $m = 1$, delete A's top when present, without adding it to B.

An empty source or full destination makes a transfer/copy a no-op; a blocked transfer never deletes the source element. Rotating an empty or singleton stack is also a no-op. Unknown opcodes are invalid rather than no-ops. A legal no-op consumes one instruction and must be reported. The solver stops with `program_complete` after the supplied program, or `max_steps` when a smaller step limit is reached. Complete-program admission sets $M = L$, with program-length bands spanning four to thirty-two instructions. This admission range is not an additional limit on the abstract stack state.

DS probes maintenance of ordered content under noncommuting operations, the distinction between movement and copying/deletion, and persistence of the mode across instructions. It does not require inventing or minimizing a program, and bounded success does not establish unbounded pushdown computation. The example below includes rotation, copying, a mode change, conditional deletion, and stack exchange. After `rix`, `sov` deletes from A rather than transferring to B; the final state is $(\texttt{rt}, \texttt{cqm}, 1)$.

Listing 1: DS input.

```
[SUITE] EWG-Lite
[GAME] dualstack
[GAME_VERSION] 1.0
[TARGET_FORMAT] dualstack-canonical-trace-v1
[INSTANCE]
A=qmc
B=rt
MODE=0
PROGRAM=lum,piv,rix,sov,tal,vek
MAX_STEPS=6
[RESPONSE]
Return the canonical trace and final state only.
```

Listing 2: DS complete target, including all six transitions.

```
STEP|INDEX=0|ACTION=lum|BEFORE_A=qmc|BEFORE_B=rt|BEFORE_M=0|
    AFTER_A=cqm|AFTER_B=rt|AFTER_M=0
STEP|INDEX=1|ACTION=piv|BEFORE_A=cqm|BEFORE_B=rt|BEFORE_M=0|
    AFTER_A=cqm|AFTER_B=rtm|AFTER_M=0
STEP|INDEX=2|ACTION=rix|BEFORE_A=cqm|BEFORE_B=rtm|BEFORE_M=0|
    AFTER_A=cqm|AFTER_B=rtm|AFTER_M=1
STEP|INDEX=3|ACTION=sov|BEFORE_A=cqm|BEFORE_B=rtm|BEFORE_M=1|
    AFTER_A=cq|AFTER_B=rtm|AFTER_M=1
STEP|INDEX=4|ACTION=tal|BEFORE_A=cq|BEFORE_B=rtm|BEFORE_M=1|
    AFTER_A=rtm|AFTER_B=cq|AFTER_M=1
STEP|INDEX=5|ACTION=vek|BEFORE_A=rtm|BEFORE_B=cq|BEFORE_M=1|
    AFTER_A=rt|AFTER_B=cqm|AFTER_M=1
FINAL|A=rt|B=cqm|M=1|HALT=program_complete
```

A.3 GLYPHREWRITE (GR)

The state is (w, c, b), where $w \in \Sigma^{\leq 12}$, $c \in \{\texttt{alpha}, \texttt{beta}, \texttt{halt}\}$, and $b \in \{0, \ldots, 12\}$ is the remaining rewrite budget. The data policy uses an initially positive budget. Each rule has a priority, a nonempty pattern, a replacement, and optional control guard/update. Table 8 gives the six fixed rules. Pattern and replacement lengths are at most two.

Table 8: GlyphRewrite rules. A dash means no guard or no control update; ϵ is an empty replacement. Lower priority numbers take precedence.

Priority	Rule	Replacement	Guard	New control
0	glim	$\texttt{vx} \to \texttt{xx}$	—	halt
1	grom	$\texttt{qm} \to \texttt{t}$	—	—
2	grel	$\texttt{mm} \to \texttt{qm}$	—	—
3	garn	$\texttt{rt} \to \texttt{tr}$	alpha	beta
4	glen	$\texttt{c} \to \texttt{vv}$	beta	alpha
5	gvox	$\texttt{x} \to \epsilon$	—	—

At a nonterminal state, choose the applicable rule with smallest priority number and then its leftmost occurrence. This minimizes (priority, match position) lexicographically over guarded matches. Overlapping matches are eligible: the leftmost occurrence across all rules does not override rule priority. One occurrence is replaced, the control update is applied if present, and b decreases by one. Match positions refer to the current pre-state rather than the initial string.

Stopping checks have the following precedence. A control of `halt` or an occurrence of `xx` gives `terminal_state`. Otherwise, $b = 0$ gives `max_rounds`; otherwise, no applicable rule

gives `no_match`. If the selected replacement would exceed length twelve, execution stops with `length_limit`: state and budget remain unchanged, and no lower-priority rule is attempted. Only successful rewrites produce `STEP` records; a stopping check is not a rewrite. This includes the zero-step case of an initially terminal state.

Unlike DS/GF, GR's next operation is content-driven, not read from an explicit program. The model must bind the selected rule to its correct occurrence while tracking string length, control, and remaining budget. A correct final string alone does not establish correct rule positions or termination; the full terminal state also contains control and rounds. In the example, rules `grel`, `grom`, `garn`, `glen`, and `glim` occur at positions $0, 0, 1, 3, 4$. The final rewrite is terminal with one round still available.

Listing 3: GR input.

```
[SUITE] EWG-Lite
[GAME] glyphrewrite
[GAME_VERSION] 1.0
[TARGET_FORMAT] glyphrewrite-canonical-trace-v1
[INSTANCE]
GLYPHS=mmrtcx
CONTROL=alpha
ROUNDS=6
[RESPONSE]
Return the canonical trace and final state only.
```

Listing 4: GR complete target, including all five rewrites.

```
STEP|INDEX=0|RULE_ID=grel|MATCH_START=0|BEFORE_GLYPHS=mmrtcx|
    BEFORE_CONTROL=alpha|BEFORE_ROUNDS=6|AFTER_GLYPHS=qmrtcx|
    AFTER_CONTROL=alpha|AFTER_ROUNDS=5
STEP|INDEX=1|RULE_ID=grom|MATCH_START=0|BEFORE_GLYPHS=qmrtcx|
    BEFORE_CONTROL=alpha|BEFORE_ROUNDS=5|AFTER_GLYPHS=trtcx|
    AFTER_CONTROL=alpha|AFTER_ROUNDS=4
STEP|INDEX=2|RULE_ID=garn|MATCH_START=1|BEFORE_GLYPHS=trtcx|
    BEFORE_CONTROL=alpha|BEFORE_ROUNDS=4|AFTER_GLYPHS=ttrcx|
    AFTER_CONTROL=beta|AFTER_ROUNDS=3
STEP|INDEX=3|RULE_ID=glen|MATCH_START=3|BEFORE_GLYPHS=ttrcx|
    BEFORE_CONTROL=beta|BEFORE_ROUNDS=3|AFTER_GLYPHS=ttrvvx|
    AFTER_CONTROL=alpha|AFTER_ROUNDS=2
STEP|INDEX=4|RULE_ID=glim|MATCH_START=4|BEFORE_GLYPHS=ttrvvx|
    BEFORE_CONTROL=alpha|BEFORE_ROUNDS=2|AFTER_GLYPHS=ttrvxx|
    AFTER_CONTROL=halt|AFTER_ROUNDS=1
FINAL|GLYPHS=ttrvxx|CONTROL=halt|ROUNDS=1|HALT=terminal_state
```

A.4 GRIDFLUX (GF)

The state is (G, q, d, p). The grid $G \in \Sigma^{H \times W}$ has $2 \le H, W \le 5$; $q = (r, c)$ is an in-bounds zero-based cursor; d is `north`, `east`, `south`, or `west`; and $p \in \{0, 1\}$ is polarity. Rows increase downward and columns rightward. Forward offsets are respectively $(-1, 0), (0, 1), (1, 0), (0, -1)$. Grid dimensions are unchanged by execution. The clockwise orientation cycle is north–east–south–west, and the glyph cycle is $c \to m \to q \to r \to t \to v \to x \to c$.

Table 9: GridFlux operations. All guards and right-hand sides use the pre-operation state.

Opcode	Meaning	Update
`bex`	Move forward	Move q by the forward offset of d if the destination is in bounds; otherwise no-op.
`cud`	Move backward	Move by the negative forward offset if in bounds; otherwise no-op.
`dax`	Turn clockwise	Advance d by $90°$ clockwise, without rotating the grid.
`fip`	Turn counterclockwise	Advance d by $90°$ counterclockwise, without rotating the grid.
`huz`	Toggle polarity	Set $p \leftarrow 1 - p$.
`jol`	Cycle current glyph	Change $G[q]$ one step forward in the glyph cycle for $p = 0$, or backward for $p = 1$.
`kem`	Swap forward cells	Exchange $G[q]$ with its forward neighbor if in bounds. Cursor is fixed; a blocked neighbor gives a no-op.
`wiv`	Shift cursor row	Cyclically shift row r right for $p = 0$ and left for $p = 1$; cursor coordinates stay fixed.
`yot`	Shift cursor column	Cyclically shift column c down for $p = 0$ and up for $p = 1$; cursor coordinates stay fixed.
`zun`	Conditional swap/turn	If $G[q] \in \{\texttt{q}, \texttt{x}\}$, execute `kem`; otherwise turn clockwise for $p = 0$ and counterclockwise for $p = 1$.

Writing $\delta = 1 - 2p$, the simultaneous shift assignments are

$$\texttt{wiv}: \quad G'_{r,j} = G_{r,(j-\delta) \bmod W}, \tag{6}$$

$$\texttt{yot}: \quad G'_{i,c} = G_{(i-\delta) \bmod H,c}. \tag{7}$$

Other rows or columns remain unchanged. These are simultaneous cyclic permutations, not sequential in-place overwrites. Movement and neighbor swapping do not wrap at boundaries; row/column shifts do. If `zun` selects a blocked swap, it remains a no-op instead of falling back to a turn. The literal directions of `dax` and `fip` do not depend on polarity.

As in DS, no-ops consume instructions, unknown opcodes are invalid, and the solver distinguishes `program_complete` from `max_steps`. Complete-program admission spans four to thirty-two instructions. GF probes coordination of cursor, orientation, polarity, and content updates, not autonomous navigation toward a goal. Cells outside an operation's permitted footprint must remain unchanged. The example's sixth instruction is a blocked swap at the bottom edge and is still reported; polarity one makes `yot` shift upward, and the final `zun` turns because the current glyph is `m`.

Listing 5: GF input.

```
[SUITE] EWG-Lite
[GAME] gridflux
[GAME_VERSION] 1.0
[TARGET_FORMAT] gridflux-canonical-trace-v1
[INSTANCE]
GRID=cmq/rtv/xcm
CURSOR=1,1
ORIENTATION=east
POLARITY=0
PROGRAM=jol,huz,wiv,dax,bex,kem,yot,zun
MAX_STEPS=8
[RESPONSE]
Return the canonical trace and final state only.
```

Listing 6: GF complete target, including the no-op transition.

```
STEP|INDEX=0|ACTION=jol|BEFORE_GRID=cmq/rtv/xcm|
BEFORE_CURSOR=1,1|BEFORE_ORIENTATION=east|
BEFORE_POLARITY=0|AFTER_GRID=cmq/rvv/xcm|
AFTER_CURSOR=1,1|AFTER_ORIENTATION=east|
AFTER_POLARITY=0

STEP|INDEX=1|ACTION=huz|BEFORE_GRID=cmq/rvv/xcm|
BEFORE_CURSOR=1,1|BEFORE_ORIENTATION=east|
BEFORE_POLARITY=0|AFTER_GRID=cmq/rvv/xcm|AFTER_CURSOR=1,1|
AFTER_ORIENTATION=east|AFTER_POLARITY=1

STEP|INDEX=2|ACTION=wiv|BEFORE_GRID=cmq/rvv/xcm|
BEFORE_CURSOR=1,1|BEFORE_ORIENTATION=east|BEFORE_POLARITY=1|
AFTER_GRID=cmq/vvr/xcm|AFTER_CURSOR=1,1|AFTER_ORIENTATION=east|
AFTER_POLARITY=1

STEP|INDEX=3|ACTION=dax|BEFORE_GRID=cmq/vvr/xcm|BEFORE_CURSOR=1,1|
BEFORE_ORIENTATION=east|BEFORE_POLARITY=1|AFTER_GRID=cmq/vvr/xcm|
AFTER_CURSOR=1,1|AFTER_ORIENTATION=south|AFTER_POLARITY=1

STEP|INDEX=4|ACTION=bex|BEFORE_GRID=cmq/vvr/xcm|BEFORE_CURSOR=1,1|
BEFORE_ORIENTATION=south|BEFORE_POLARITY=1|AFTER_GRID=cmq/vvr/xcm|
AFTER_CURSOR=2,1|AFTER_ORIENTATION=south|AFTER_POLARITY=1

STEP|INDEX=5|ACTION=kem|BEFORE_GRID=cmq/vvr/xcm|BEFORE_CURSOR=2,1|
BEFORE_ORIENTATION=south|BEFORE_POLARITY=1|AFTER_GRID=cmq/vvr/xcm|
AFTER_CURSOR=2,1|AFTER_ORIENTATION=south|AFTER_POLARITY=1

STEP|INDEX=6|ACTION=yot|BEFORE_GRID=cmq/vvr/xcm|BEFORE_CURSOR=2,1|
BEFORE_ORIENTATION=south|BEFORE_POLARITY=1|AFTER_GRID=cvq/vcr/xmm|
AFTER_CURSOR=2,1|AFTER_ORIENTATION=south|AFTER_POLARITY=1

STEP|INDEX=7|ACTION=zun|BEFORE_GRID=cvq/vcr/xmm|BEFORE_CURSOR=2,1|
BEFORE_ORIENTATION=south|BEFORE_POLARITY=1|AFTER_GRID=cvq/vcr/xmm|
AFTER_CURSOR=2,1|AFTER_ORIENTATION=east|AFTER_POLARITY=1

FINAL|GRID=cvq/vcr/xmm|CURSOR=2,1|ORIENTATION=east|POLARITY=1|
HALT=program_complete
```

A.5 Construction, structural strata, and provenance

Legal states, constructor support, structural admission, and partition/exposure grouping are different objects. The preceding bounds do not imply uniform sampling or exhaustive coverage. In the frozen L1–L5 structural policy, DS/GF program lengths are 4–6, 7–10, 11–16, 17–24, and 25–32, with minimum effective steps 1, 3, 5, 8, and 12. GR bands contain 1–2, 3–4, 5–7, 8–10, and 11–12 successful rewrites, with initial-length ranges 1–6, 1–8, 2–10, 2–12, and 3–12. Minimum GF areas are 4, 4, 6, 9, and 12. Admission also depends on family-specific interactions. These are structural labels, not a common empirical difficulty scale or a claim that the evaluated panel uniformly covers every band.

Parent–twin construction changes the initial DS mode, GR alpha/beta control, or GF polarity while preserving other fields. Both members are simulated and admitted under the family policy. Success on both measures robustness to that input intervention, not proof of an internal causal representation. A primitive pair can belong to a larger structural/exposure component; split eligibility follows the data manifest, not merely a pair ID or a fresh seed. Constructor expansion changes sampled support, not the fixed semantics unless a rule actually changes. The three complete examples above are

explanatory fixtures rather than model predictions or held-out benchmark cases. Their previously reported thirty native rule and boundary checks do not establish population-wide test coverage.

DS belongs to neural execution and stack-manipulation traditions (Zaremba & Sutskever, 2015; Delétang et al., 2023). GR is bounded ordered rewriting with finite control, related to classical algorithmic rewriting and contemporary string-transformation tasks (Markov, 1954; Chen et al., 2025; Stojanovski et al., 2025). GF has a grid-program precedent in Karel learning, but predicts a supplied execution instead of inducing a program from examples (Devlin et al., 2017). These references acknowledge related mechanisms, not a documented direct derivation of the particular tables. The operation names are opaque identifiers, not English acronyms; their meanings are the explicit updates. We do not invent letter-by-letter etymologies or assume the names are single tokens or absent from pretraining. The identifiers preserve correspondence to the evaluated prompts and experts.

B Evaluation definitions and data roles

B.1 Historical WordGame predicates

A canonical response contains zero or more STEP records and one FINAL record, with exact family-specific field order, legal enumerations, consecutive indices, and no extra commentary. Parse (P) records syntactic acceptance; it is not a semantic certificate. The native evaluator validates its reference against the simulator. For a parse-valid prediction, let K be its reported transition count, T the reference count, $\hat{a}_t$ its post-state, s_t the reference post-state, $\hat{s}_f$ its separately declared final state, and $\hat{h}$ its halt label. Then

$$F = \mathbf{1}[\hat{s}_f = s_T], \qquad H = \mathbf{1}\Big[K = T,\ \hat{h} = h_{\mathrm{ref}}\Big], \qquad J_{\mathrm{rep}} = F\mathbf{1}[V_{\mathrm{reported}} = 1], \quad (8)$$

$$S = \begin{cases} \dfrac{\sum_{t=1}^{\min(K,T)} \mathbf{1}[\hat{a}_t = s_t]}{\max(K,T)}, & \max(K,T) > 0, \\ 1, & K = T = 0, \end{cases} \quad (9)$$

$$V_{\mathrm{reported}} = \begin{cases} K^{-1} \sum_{t=1}^{K} v_t, & K > 0, \\ 1, & K = T = 0, \\ 0, & K = 0 < T. \end{cases} \quad (10)$$

Final compares all terminal fields, including GR's remaining budget, but not the halt label. Missing/extra steps enlarge the denominator of S. Completed parse-invalid outputs remain failures in unconditional success rates.

The local transition predicate checks legal states and continuity from the initial state through predicted post-states, but its action requirements differ by family. DS replays the reported legal opcode without requiring it to match the input instruction at that position. GR recomputes the deterministic priority rule, leftmost position, and resulting state. GF additionally requires input-action alignment and that changed cells belong to the allowed footprint. Extra GF steps are invalid. A valid nonempty prefix plus a correct final declaration can therefore satisfy Reported-transition joint despite omitted later steps or wrong stopping. This is a consequence of the predicates, not a prevalence estimate.

The retained auxiliary semantic score is $0.50F + 0.35S + 0.10V_{\mathrm{reported}} + 0.05H$, capped at 0.5 when $F = 1$ but $V_{\mathrm{reported}} < 1$. It is not an accuracy or a replacement for the separate endpoints. Native diagnostic categories include missing/extra steps, pre-/post-state and final-state errors, rule/position errors, and halt errors. Their checking order is an implementation-defined diagnostic convention, not the first internal cause of model failure. High P with low F indicates admissible but incorrect responses; F with low Reported-transition joint indicates invalid reported transitions in some final-correct responses. Neither situation identifies hidden reasoning.

B.2 Panels, development exposure, and reuse

The primary WordGame panel has 768 inputs in 384 complete components, balanced across three families. DEV192 contains 192 of those inputs. The registered case-ID overlap between D24 and DEV768 is zero, but both are exposed development material; absence of a case-ID overlap alone

Table 10: Evaluation panels and generation budgets. Context is the configured total sequence limit; all settings use one greedy answer per input. A dash is an unbound field, not an unlimited context.

Panel	Inputs	Output cap	Context	Precision
WordGame development panel	768	4,096	—	BF16 backbone; FP32 adapters
Fixed GSM8K subset	256	2,048	4,096	FP16
IFEval benchmark	541	2,048	4,096	FP16
AIME2024	30	8,192	10,240	FP16
Matched LiveCodeBench panel	400	8,192	10,240	FP16

Table 11: WordGame training budgets and home-family Final scores. Budgets are checkpoint-bound; scores use the 256-input development family. Equal home accuracy does not isolate a causal training-budget effect or imply comparable logits.

Specialist	Unique training inputs	Passes	Updates	Home Final
DS	8,000	1	1,000	232/256 (90.6%)
GR	8,000	4	4,000	232/256 (90.6%)
GF	8,000	4	4,000	232/256 (90.6%)

is not a complete structural independence proof. The saved sensitivity analysis excludes nineteen components, leaving 730 inputs and 365 components, without replacing the primary result. The completed 96-input confirmation is reported separately in Appendix E; it does not change the status of the exposed development panel or establish completion of the larger reservation.

The external main panel uses the benchmark-specific inventories in Table 10. The Pack supplement reuses 96 GSM8K, 96 IFEval, all 30 AIME2024, and 96 LCB inputs, with matching baseline outputs. These 318 inputs do not enlarge the count of independent main-panel questions. Previously saved answers reused in a completed run are counted once. A matched pair analysis needs labels for its reference and candidate; a union analysis needs all designated experts and compared strategies. A separate global intersection is not imposed on every two-strategy comparison, but changing supports must be made explicit. No pooled accuracy over the heterogeneous external benchmarks is used.

B.3 Generation budgets

B.4 Specialist training budget and observed capability

Table 11 binds training accounting to the exact checkpoint identities in the frozen expert bank. The original DS checkpoint/plan and GR/GF training receipts were retrieved read-only through source references in the evidence archive, with archive/member hashes verified. No new training or evaluation was run. DS's single pass is derived from 8,000 consumed examples and 8,000 unique scheduled inputs; GR/GF explicitly record four completed passes. Full optimizer hyperparameters and training/exposure correspondence are not thereby established.

B.5 External scoring and resource-policy endpoint

GSM8K and AIME2024 use their retained native answer-scoring implementations. LCB reports the fraction of one generated solution per question passing the retained functional tests. IFEval retains strict/loose and prompt/instruction metrics: a prompt succeeds when its required checks all pass under the relevant version, whereas instruction accuracy divides the total passed checks by all applicable checks. The full IFEval panel contains 541 prompts and 834 instructions. WordGame Strict and IFEval strict are unrelated predicates.

The retained LCB scoring environment is a uniform Linux/arm64 container image with one CPU and 1 GiB per container, six seconds per test, and a 120-second container limit. This is the study's resource policy, not a claim that the LCB authors mandate that Docker configuration. All 3,200 main-comparison answers were generated. TIES has 399 normally returned native scores, including 85 passes, and one scoring-container OOM at index 105 (question 3091). Our recorded analysis

decision counts that case as a failure, giving 85/400 (21.25%) in a separate analysis-policy layer. The original OOM receipt and historical native ledger remain unchanged; the decision is not retrospectively called preregistered. Treating that one case as correct would yield 86/400 (21.50%), a sensitivity scenario rather than an observation. The case is not part of the matched 96-input LiveCodeBench subset. Other infrastructure failures are not automatically subject to this rule.

B.6 Statistics and interpretation

WordGame analyses resample complete parent–twin components within family, preserving both members and method pairing. The retained analysis uses 10,000 resamples and linearly interpolated quantiles. Main Final/Reported-transition joint intervals use seed 20260923; Parse and paired comparisons use seed 20260926. An empirical all-zero interval is degenerate and does not imply zero population uncertainty. For component-level all-member success, $A_c(E) = \prod_{x \in E_c} Y^E(x)$ is evaluated on the full declared member set; it is not an additional independent input count.

External single-strategy binary intervals are Wilson 95% intervals. Same-question differences use 10,000 paired question bootstrap resamples with seed 20260926. Intervals are exploratory and not corrected for multiple comparisons. Individual strategy intervals are not subtracted as if the samples were independent. IFEval instruction intervals resample prompts with their associated constraint counts when those flags exist. For CoRE, Mean, TIES, and the Pack instruction supplement, missing retained prompt-level flags make those instruction intervals unavailable; treating all constraints as independent or rescoring solely to manufacture a missing interval is not used. Repeated seeds, tokens, shards, and methods do not create additional independent expert groups.

C Implementation distinctions and reproducibility scope

The method descriptions below distinguish the published mechanism, its actual task-specific implementation, and the evaluated condition. WordGame EMFuse uses full-axis normalized log-probability pooling. LogPool-0.75 uses the same type of logarithmic pool with task-specified coefficients. FixFuse instead averages each expert's own top-ten normalized probabilities, with main-expert EOS and token-string-prefix safeguards. The weight number alone therefore does not identify an equivalent policy. Mathematical equivalence between weighted logits and log probabilities is not a certificate of bitwise equality across precisions or execution paths.

For the WordGame CoRE-based controls, local indices are main-first and are distinct from the parent-model index in Section 3. Suppressing the generation-token index, let p_i denote each expert's top-ten softmax probabilities embedded in the common vocabulary axis, with zeros elsewhere, and let $\bar{p}$ be their arithmetic mean. With main expert indexed by 0, the retained rule forms

$$r_{iv} = \mathbf{1}[p_i(v) + p_0(v) > \epsilon] \, \frac{\exp(-\beta |p_i(v) - \bar{p}(v)|) - \exp(-\beta)}{1 - \exp(-\beta)}, \tag{11}$$

$$R_i = \sum_v r_{iv}, \qquad b_i = \frac{R_i}{H_i + \epsilon}, \qquad b_0' = \max\left(b_0, \sum_{j \neq 0} b_j\right), \quad b_j' = b_j \ (j \neq 0), \tag{12}$$

$$w_i = \frac{b_i'}{\sum_j b_j'}, \qquad H_i = -\sum_v \widetilde{p}_i(v) \log \widetilde{p}_i(v), \quad \widetilde{p}_i(v) = \max\left(\frac{p_i(v)}{\max(\sum_u p_i(u), \epsilon)}, \epsilon\right). \tag{13}$$

Here $\epsilon = 10^{-12}$ and $\beta = 2$, and reductions use the retained full output axis. The entropy calculation normalizes before applying the ϵ floor; the floored vector is not renormalized. The support mask excludes jointly absent token positions from the consistency sum. FixFuse computes diagnostic weights but substitutes $(0.75, 0.125, 0.125)$ in the arithmetic mixture. DynFuse uses w_i. CoRE additionally uses released auxiliary factors $r_{iv} N_i / R_i$, where N_i counts nonzero r_{iv}, rather than applying a uniform scalar shrinkage to each auxiliary model. The main distribution is not multiplied by an auxiliary factor. Main-EOS short-circuiting and token-string-prefix fallback remain part of the declared generation policies.

C.1 Method abbreviation mapping

Table 12: Method abbreviation mapping. Short names refer to distinct evaluated conditions, not new proposed algorithms.

Short name	Full meaning	Definition / scope
HOME	Task-informed expert routing	Select the expert indicated by the public family/domain mapping; no answer oracle.
EMFuse	Energy-based test-time fusion	Equal-weight log pooling; precision and execution paths are documented by expert group.
SynFuse	Syntax-constrained EMFuse	Apply the public-syntax mask to EMFuse scores; no semantic feedback.
FixFuse	Fixed task-aware weighting	Own-top-ten arithmetic mixture with weights $(0.75, 0.125, 0.125)$; CoRE EOS/fallback, without token factors.
DynFuse	Dynamic task-aware weighting	Matched per-expert top-ten mixture with consistency/entropy model weights; no token factors.
CoRE	Full CoRE configuration	Task-informed main expert; full model- and token-level factors for the evaluated implementation.
SAFE	SAFE fusion	Retained token-selection/sharpening condition; distinct from the CoRE controls.
EntRoute	Input-entropy routing	Minimum first-output-position entropy; freeze one input-level choice before saved-answer reuse.
NLLRoute	Input-NLL routing	Minimum mean prompt NLL; one input-level choice, not PackLLM optimization.
Mean	Mean parameter merging	Fixed parameter-average recipe, not probability averaging.
TIES	TIES parameter merging	Fixed task-update merging recipe; the LiveCodeBench policy exception is disclosed.
PackLLM-opt	Prompt-optimized fusion	Whole-prompt native-dtype scoring, sequential raw-logit fusion, frozen 21-point grid and tie/order rules.
LogPool-0.75	Task-aware logarithmic pooling	Full-support log pool with main-first weights $(0.75, 0.125, 0.125)$; distinct from FixFuse.
LogPool-0.50	Log-pool coefficient control	Main-first coefficients $(0.5, 0.25, 0.25)$.
LogPool-0.20	Log-pool coefficient control	Main-first coefficients $(0.2, 0.4, 0.4)$.

WordGame expert order is DS/GF/GR before main-first remapping. The source group is identified in the retained registry by `ab2325f3...dd8aa`; its policies use EOS token ID 151643 and an output cap of 4,096 with BF16 backbone execution and original FP32 adapters. The paired controls preserve original model execution and change only effective model weights. The external group uses Math/Coding/IF order, FP16 model forwards, and EOS token ID 151645, with the separately evaluated parent excluded from fusion. Full tokenizer/weight identities, literal external templates, per-recipe construction settings, and source hashes require the corresponding execution manifests; missing fields are not replaced by current library defaults or source-paper hyperparameters. The registry maps retained evidence; it does not imply that all executable dependencies are embedded in the manuscript.

Parameter averaging is described at the level actually implemented, with adapter increments constructed as $\Delta W = (\alpha/r)BA$ where applicable; averaging A and B independently is not assumed equivalent. The retained Mean and TIES labels do not imply equal construction cost or equivalent interference mechanisms. CPLUS/CMINUS score shadows are not proofs of full independent-expert trajectory invariance. The D24 gate-only transfer of CoRE model weights to EMFuse is a separately bounded pilot and is not substituted for any 768-input row.

Only recorded costs are reportable. Expert training, offline merge construction, prompt-feature measurement, generation, and scoring are separate stages. Saved-answer routing avoids rerunning matching answers in this analysis; a deployed input router need not generate all candidate answers. Offline reuse time is not measured online routing latency. Hardware/topology differences and missing lifecycle telemetry preclude a complete cross-method efficiency claim from the present records. These cost records do not support a training-to-deployment efficiency comparison across the two expert groups.

C.2 ANONYMOUS EVIDENCE INDEX

The accompanying `anonymous_reproducibility/INDEX.md` lists relative artifact names and SHA256 identities. Its data files support the displayed aggregate values and paired counts. A separate training-budget binding records read-only retrieval of historical receipts matching the frozen checkpoint identities; those receipts add provenance, not new experiment outcomes. WordGame bank, backbone, tokenizer, and three checkpoint identities are included as hash records.

D ADDITIONAL ACCEPTED RESULTS

D.1 TABLE PROVENANCE

All capability and comparison tables in this revision are generated from the frozen local archive `20260926-v1.zip`. WordGame method-by-family and endpoint records come from `wordgame/main/tables/main_methods.csv`; cap counts come from `wordgame/main/tables/failure_categories.csv`. Full-panel external source capabilities use `m2rl/tables/EXPERT_BY_TARGET.csv`; matched source capabilities and combination counts/intervals use `m2rl/tables/SCORES.csv`. Rows belonging to the same scientific task are aggregated across accepted runs; engineering failures are not additional model outcomes. Specialist matrix entries are read directly from their expert–target records, not inferred from aggregate totals. The nested PackLLM-opt comparison is never substituted for a missing full-panel result.

D.2 Full WordGame outcomes and retained intervals

Table 13 includes all seventeen registered primary-panel strategies, including three single specialists, the two coefficient controls, and both selectors. Intervals reproduce the retained source tables; they are not recalculated to choose favorable endpoints. All denominators are 768, including failed and capped responses. A single specialist's all-family accuracy differs from its home-family accuracy. The 730-input exclusion sensitivity does not turn exposed development data into independent confirmation.

Table 13: Full WordGame development results: successful/total inputs, accuracy (%), and exact retained 95% component-bootstrap intervals.

Strategy	Final: n/N (%) [CI]	Reported-transition joint: n/N (%) [CI]
HOME	696/768 (90.62) [88.15, 92.97]	674/768 (87.76) [84.90, 90.49]
DS specialist	232/768 (30.21) [28.65, 31.64]	229/768 (29.82) [28.12, 31.38]
GR specialist	232/768 (30.21) [28.65, 31.51]	213/768 (27.73) [25.78, 29.56]
GF specialist	232/768 (30.21) [28.78, 31.51]	232/768 (30.21) [28.78, 31.51]
EMFuse	1/768 (0.13) [0.00, 0.39]	0/768 (0.00) [0.00, 0.00]
SynFuse	2/768 (0.26) [0.00, 0.65]	0/768 (0.00) [0.00, 0.00]
LogPool-0.75	640/768 (83.33) [80.08, 86.46]	609/768 (79.30) [75.78, 82.81]
FixFuse	697/768 (90.76) [88.28, 93.23]	675/768 (87.89) [85.03, 90.76]
DynFuse	341/768 (44.40) [40.49, 48.44]	302/768 (39.32) [35.41, 43.49]
CoRE	697/768 (90.76) [88.28, 93.10]	678/768 (88.28) [85.55, 91.02]
SAFE	1/768 (0.13) [0.00, 0.39]	0/768 (0.00) [0.00, 0.00]
LogPool-0.50	124/768 (16.15) [13.54, 18.75]	111/768 (14.45) [11.98, 16.93]
LogPool-0.20	0/768 (0.00) [0.00, 0.00]	0/768 (0.00) [0.00, 0.00]
Mean	2/768 (0.26) [0.00, 0.65]	1/768 (0.13) [0.00, 0.39]
TIES	52/768 (6.77) [4.95, 8.59]	52/768 (6.77) [4.95, 8.59]
EntRoute	573/768 (74.61) [71.35, 77.73]	556/768 (72.40) [69.14, 75.65]
NLLRoute	235/768 (30.60) [29.04, 32.03]	216/768 (28.12) [26.04, 30.08]

Table 14: Full specialist capability matrix underlying Table 2; Final endpoint, 256 inputs per target family.

Specialist	Target	Correct/total	Accuracy (%)	95% CI
DS specialist	DS	232/256	90.62	[85.94, 94.92]
DS specialist	GR	0/256	0.00	[0.00, 0.00]
DS specialist	GF	0/256	0.00	[0.00, 0.00]
GR specialist	DS	0/256	0.00	[0.00, 0.00]
GR specialist	GR	232/256	90.62	[85.94, 94.53]
GR specialist	GF	0/256	0.00	[0.00, 0.00]
GF specialist	DS	0/256	0.00	[0.00, 0.00]
GF specialist	GR	0/256	0.00	[0.00, 0.00]
GF specialist	GF	232/256	90.62	[86.33, 94.53]

Table 15: WordGame output diagnostics. Parse is a separate admissibility rate, not task accuracy; capped outputs remain in the scientific denominator.

Strategy	Parse/total	Parse (%)	95% CI	Capped/total
HOME	767/768	99.87	[99.61, 100.00]	0/768
DS specialist	507/768	66.02	[65.10, 66.67]	10/768
GR specialist	255/768	33.20	[32.94, 33.33]	301/768
GF specialist	256/768	33.33	[33.33, 33.33]	8/768
EMFuse	256/768	33.33	[33.33, 33.33]	62/768
SynFuse	754/768	98.18	[97.01, 99.22]	14/768
LogPool-0.75	768/768	100.00	[100.00, 100.00]	0/768
FixFuse	768/768	100.00	[100.00, 100.00]	0/768
DynFuse	759/768	98.83	[97.92, 99.61]	1/768
CoRE	768/768	100.00	[100.00, 100.00]	0/768
SAFE	491/768	63.93	[62.63, 65.10]	5/768
LogPool-0.50	510/768	66.41	[66.02, 66.67]	0/768
LogPool-0.20	255/768	33.20	[32.94, 33.33]	212/768
Mean	161/768	20.96	[17.58, 24.48]	398/768
TIES	234/768	30.47	[29.04, 31.64]	33/768
EntRoute	632/768	82.29	[79.69, 84.77]	3/768
NLLRoute	259/768	33.72	[33.20, 34.51]	299/768

D.3 PACKLLM-OPT ON THE MATCHED SUPPLEMENT

Every row in Table 16 uses the same frozen subset for that benchmark. A result on the 96-input LiveCodeBench subset is not compared directly with another strategy's full matched LiveCodeBench panel accuracy. All 30 AIME2024 questions are reused rather than counted as a second independent AIME evaluation. The same aggregate scores for PackLLM-opt and EMFuse do not imply identical answers or correctness vectors.

Table 16: Matched PackLLM-opt comparison: counts, accuracy (%), and retained Wilson 95% confidence intervals. Every strategy uses identical membership within each benchmark; AIME2024 is shared with the full study panel.

Strategy	Correct/total	Accuracy (%)	95% CI
Fixed GSM8K subset			
Parent	80/96	83.33	[74.63, 89.47]
Math specialist	80/96	83.33	[74.63, 89.47]
Coding specialist	77/96	80.21	[71.14, 86.95]
Instruction specialist	79/96	82.29	[73.46, 88.64]
EMFuse	80/96	83.33	[74.63, 89.47]
CoRE	81/96	84.38	[75.81, 90.30]
Mean	78/96	81.25	[72.30, 87.80]
TIES	79/96	82.29	[73.46, 88.64]
PackLLM-opt	80/96	83.33	[74.63, 89.47]
IFEval benchmark			
Parent	27/96	28.12	[20.11, 37.83]
Math specialist	29/96	30.21	[21.93, 40.01]
Coding specialist	28/96	29.17	[21.02, 38.92]
Instruction specialist	27/96	28.12	[20.11, 37.83]
EMFuse	28/96	29.17	[21.02, 38.92]
CoRE	24/96	25.00	[17.41, 34.51]
Mean	27/96	28.12	[20.11, 37.83]
TIES	26/96	27.08	[19.20, 36.73]
PackLLM-opt	27/96	28.12	[20.11, 37.83]
AIME2024			
Parent	9/30	30.00	[16.66, 47.88]
Math specialist	11/30	36.67	[21.87, 54.49]
Coding specialist	15/30	50.00	[33.15, 66.85]
Instruction specialist	12/30	40.00	[24.59, 57.68]
EMFuse	13/30	43.33	[27.38, 60.80]
CoRE	11/30	36.67	[21.87, 54.49]
Mean	15/30	50.00	[33.15, 66.85]
TIES	11/30	36.67	[21.87, 54.49]
PackLLM-opt	13/30	43.33	[27.38, 60.80]
Matched LiveCodeBench panel			
Parent	26/96	27.08	[19.20, 36.73]
Math specialist	17/96	17.71	[11.36, 26.54]
Coding specialist	30/96	31.25	[22.85, 41.09]
Instruction specialist	24/96	25.00	[17.41, 34.51]
EMFuse	23/96	23.96	[16.53, 33.39]
CoRE	24/96	25.00	[17.41, 34.51]
Mean	25/96	26.04	[18.31, 35.62]
TIES	21/96	21.88	[14.78, 31.14]
PackLLM-opt	23/96	23.96	[16.53, 33.39]

In the 96-input LiveCodeBench subset, the IF expert receives the largest Pack weight on 95 inputs, and Coding has zero weight on 90; their mean weights are approximately 95.29% and 1.09%, respectively. Coding solves 30/96, compared with Pack's 23/96; the paired difference is -7.29 percentage points with an exploratory interval $[-16.67, 1.04]$. This is an input-weight diagnostic, not evidence that zero Coding weight causes failure or that Pack is conclusively worse. Native-dtype whole-prefix

scoring is distinguished from an older cached-FP32 qualification path; WordGame NLLRoute hard selection is not Pack's optimization algorithm. Only accepted, matched results enter this comparison; no additional WordGame Pack or full external-extension outcomes are imputed.

Pack's IFEval loose-prompt success is 30/96. Its strict-instruction result is 53/140 and loose-instruction result 57/140. Missing retained per-prompt instruction flags preclude a clustered instruction interval. Relative to EMFuse, the counts of discordant correctness labels on the 96-input GSM8K subset, the 30-input AIME2024 panel, and the 96-input LiveCodeBench subset are 12, 6, and 16. LCB's discordance rate is 16.67%, with a retained paired-bootstrap interval $[9.38, 23.96]$; each strategy solves eight cases the other misses. This is not a statistical equivalence test or a universal ranking of risk preferences.

D.4 IFEval's four native endpoints

The full panel contains 541 prompts and 834 instruction checks. Table 17 retains all four metrics. Prompt rates and instruction rates have different denominators; instruction success is not an independent binary observation per constraint for inferential purposes.

Table 17: Complete IFEval results, as passed/total. Clustered instruction intervals are unavailable for CoRE, Mean, and TIES because the corresponding per-prompt flags were not retained.

Strategy	Strict prompt	Strict instruction	Loose prompt	Loose instruction
SFT parent	149/541	355/834	163/541	376/834
Math	150/541	345/834	158/541	359/834
Coding	146/541	341/834	161/541	364/834
IF	147/541	345/834	163/541	366/834
EMFuse	141/541	345/834	151/541	360/834
CoRE	144/541	342/834	159/541	365/834
Mean	144/541	336/834	152/541	354/834
TIES	153/541	347/834	162/541	366/834

D.5 Secondary selection diagnostics

On WordGame, NLLRoute chooses GR on 764/768 inputs and solves 235, while EntRoute solves 573 and HOME 696. All three specialists share the same first-output-position argmax on the 768 prompts despite disjoint final-success sets. This may reflect the common formatting prefix; it is not evidence about agreement at every later decision. Both selectors use input-only frozen measurements and matched saved answers.

A saved-answer subset comparison further separates available coverage from realized selection. With only DS and GF candidates, the union contains 464 successes and NLL routing obtains 275. Adding GR expands coverage to 696 but reduces NLL success to 235. EntRoute increases from 415 to 573 under the same candidate expansion. This is a deterministic re-selection over saved responses, not two-expert fusion generation or an online timing experiment. No exported paired interval accompanies that subset contrast, and it does not establish a scaling law or an optimal expert count.

D.6 FULL-PANEL EXTERNAL RESULTS AND INTERVALS

Table 18 reports the full-panel counts, percentages, and retained Wilson intervals (256 / 541 / 30 / 400 inputs). The matched subsets used in the main-text Tables 4 and 5 are reported separately in Table 16. These marginal intervals are not method-difference tests; paired comparisons use their own resampling analysis.

Table 18: External study panels: correct/total inputs, accuracy (%), and retained Wilson 95% confidence intervals. IFEval uses strict prompt; TIES LiveCodeBench uses the disclosed policy endpoint (†).

Strategy	Correct/total	Accuracy (%)	95% CI
Fixed GSM8K subset			
Parent	216/256	84.38	[79.42, 88.31]
Math specialist	221/256	86.33	[81.58, 90.00]
Coding specialist	214/256	83.59	[78.57, 87.63]
Instruction specialist	219/256	85.55	[80.71, 89.33]
EMFuse	214/256	83.59	[78.57, 87.63]
CoRE	220/256	85.94	[81.15, 89.67]
Mean	215/256	83.98	[78.99, 87.97]
TIES	219/256	85.55	[80.71, 89.33]
IFEval benchmark			
Parent	149/541	27.54	[23.95, 31.45]
Math specialist	150/541	27.73	[24.12, 31.65]
Coding specialist	146/541	26.99	[23.42, 30.88]
Instruction specialist	147/541	27.17	[23.59, 31.07]
EMFuse	141/541	26.06	[22.54, 29.92]
CoRE	144/541	26.62	[23.07, 30.50]
Mean	144/541	26.62	[23.07, 30.50]
TIES	153/541	28.28	[24.65, 32.22]
AIME2024			
Parent	9/30	30.00	[16.66, 47.88]
Math specialist	11/30	36.67	[21.87, 54.49]
Coding specialist	15/30	50.00	[33.15, 66.85]
Instruction specialist	12/30	40.00	[24.59, 57.68]
EMFuse	13/30	43.33	[27.38, 60.80]
CoRE	11/30	36.67	[21.87, 54.49]
Mean	15/30	50.00	[33.15, 66.85]
TIES	11/30	36.67	[21.87, 54.49]
Matched LiveCodeBench panel			
Parent	120/400	30.00	[25.72, 34.66]
Math specialist	71/400	17.75	[14.32, 21.80]
Coding specialist	130/400	32.50	[28.10, 37.24]
Instruction specialist	106/400	26.50	[22.41, 31.03]
EMFuse	103/400	25.75	[21.71, 30.25]
CoRE	115/400	28.75	[24.53, 33.37]
Mean	113/400	28.25	[24.06, 32.85]
TIES†	85/400	21.25	[17.52, 25.52]

E FROZEN INDEPENDENT-INPUT CONFIRMATION AND WEIGHTING CONTRAST

Design and evidence role. The confirmation contains 96 unique inputs in 48 complete parent–twin components, balanced across DS, GR, and GF. A deterministic first-32-input partition per family was frozen before observing these outcomes, with no outcome-dependent expansion. The selection, source identities, and documented exposure/custody checks are retained. This supports an independent-input confirmation relative to the recorded development use, not an omniscient guarantee against undocumented exposure or a test of new specialist-training seeds. The original four conditions are HOME, FixFuse, DynFuse, and CoRE; the two additional policies use exactly the same inputs. Thirty-six shards are merged into six condition-level results, each with 96 answers. Prior engineering startup failures supply no scientific observations. All completed outputs are retained, with no inference recovery or rescoring required.

Four policies. Let $d = (d_0, d_1, d_2)$ be the existing normalized dynamic gate after its main-weight clamp, in main-first order. Define $r_i = d_i/(d_1 + d_2)$ for auxiliary indices $i = 1, 2$ when the denominator is positive, and $r_1 = r_2 = 1/2$ otherwise. The proposed arithmetic-pooling weights are

$$w_{\mathrm{FixFuse}} = (.75, .125, .125), \qquad w_{\mathrm{D75}} = (.75, .25r_1, .25r_2), \tag{14}$$

$$w_{\mathrm{DEQ}} = (d_0, (1 - d_0)/2, (1 - d_0)/2), \qquad w_{\mathrm{DynFuse}} = (d_0, d_1, d_2). \tag{15}$$

All four share support, main-EOS behavior, and token-string-prefix fallback. The proposed weights do not override these branches: actual weights become main-only when fallback or the main-EOS branch bypasses pooling. Gate values are computed on each policy's own generated prefix. Therefore the contrast is a four-policy intervention with common frozen inputs, not a same-prefix intervention at every token. D75 and DEQ are diagnostic policies rather than proposed general-purpose algorithms.

Table 19: Frozen confirmation: native Final and reported-transition joint (J_{rep}) on the same 96 inputs. Final intervals use 10,000 paired component-bootstrap resamples stratified by family. Every condition has 96 parseable outputs and zero capped outputs.

Condition	Final/96	Accuracy (%)	95% CI	J_{rep}/96
HOME	92	95.83	[91.67, 98.96]	92
FixFuse	93	96.88	[93.75, 100.00]	93
DynFuse	46	47.92	[37.50, 58.33]	40
CoRE	92	95.83	[91.67, 98.96]	92
D75	93	96.88	[93.75, 100.00]	93
DEQ	73	76.04	[65.63, 85.42]	68

Primary contrast and policy interaction. The frozen selection identifies FixFuse minus DynFuse Final as the primary contrast, with joint as a companion. Final improves by 47/96, or 48.96 points (paired 95% CI $[38.54, 59.38]$); joint improves by 53/96. In the Final transition from FixFuse to DynFuse, 46 successes are retained, 47 are harmed, none are rescued, and three inputs fail under both. D75 and FixFuse agree on all 96 generated token sequences, not merely their aggregate scores. DEQ rescues 27 inputs relative to DynFuse with no losses, while remaining 20 successes below FixFuse. The exploratory interaction on the accuracy scale, DynFuse $-$ DEQ $-$ D75 $+$ FixFuse, is -28.13 points (95% CI $[-38.54, -18.75]$); its joint counterpart is -29.17 points (95% CI $[-39.58, -18.75]$). Intervals use the same 10,000 paired, family-stratified component resamples with seed 20260926 and linear percentile interpolation. Other pairwise contrasts are exploratory and unadjusted. Identical observed responses do not constitute a population equivalence test.

Family and branch diagnostics. Final counts in DS/GR/GF (each out of 32) are respectively 32/30/30 for HOME, 32/30/31 for FixFuse and D75, 6/22/18 for DynFuse, 31/30/31 for CoRE, and 20/28/25 for DEQ. Thus the policy contrast varies across families. HOME and CoRE both solve 92 inputs but each uniquely solves one; FixFuse gains just one input over HOME (paired difference 1.04 points, 95% CI $[0, 3.13]$), which does not establish a reliable general improvement over task-informed selection.

Table 20: Observed branch exposure in the two added policies. Counts cover every generated token, including terminal EOS. Proposed pooling weights are not applied on bypass branches.

Policy	Tokens	Prefix fallback	Main-EOS bypass	Normal pooling
D75	98,138	98,041	96	1
DEQ	98,208	98,061	96	51

The predominance of fallback is a material qualification of the weighting result. A small number of non-bypassed decisions may alter later trajectories, but these counts alone do not identify the first causal token or establish mediation through a particular internal statistic. The reproducibility subset includes the merged outcomes, paired contrasts, family counts, definitions, and branch summaries; full native source and collection hashes remain in the internal archive. These confirmation observations are not pooled with the 768-input development statistics.