DECOUPLING VERIFICATION AND GENERATION: REASONING CONTENT GOVERNS WHAT FINE-TUNING TEACHES

Anonymous authors
Paper under double-blind review

ABSTRACT

Language models are increasingly asked to both generate solutions and verify them, and a single model is often fine-tuned to learn both at once — on the assumption that judging and solving are one competence that improves together. We show they are separately learned: a single fine-tuning run can move the two abilities in *different* directions. Training on a bare correctness verdict collapses generation almost entirely (math accuracy $63.8\% \rightarrow 0.0\%$) while leaving verification near the base level, and supervised fine-tuning on solutions (SFT), by contrast, improves generation while leaving verification weak. They are separable because they are learned from *different* parts of the signal: verification from the judgment it expresses, generation from the reasoning content it retains. A bare verdict carries only the judgment, so it leaves verification intact but collapses generation; an SFT target carries only the reasoning, so it does the reverse; a critique carries both, which is why it appears to improve both at once — not because critique is a uniquely effective objective, but because it is the one signal that contains both ingredients. Varying this content across model sizes and families, we find that whether verification training preserves generation is governed by the reasoning content of the signal, not its quantity. When generation collapses it does so by disengagement rather than lost skill: the model stops attempting problems, a failure we trace to the single instruction framing shared by all critique examples and confirm causally by mixing in direct-solve instructions. The practical consequence is that a verifier cannot be assumed to remain a capable generator; training one that still solves requires keeping reasoning content in the verification signal, and a bare verdict — the cheapest signal — is the most destructive.

1 INTRODUCTION

Language models are increasingly expected to both solve problems and judge solutions, often within a single model: self-improvement loops have a model verify and then improve its own outputs (Yuan et al., 2024), and models trained to generate are routinely repurposed as judges (Zheng et al., 2023). It is tempting to treat generation and verification as one competence, so that improving one improves the other, and a model fine-tuned for either keeps the other intact. Critique-based fine-tuning, for instance, targets better generation and reports that learning to critique delivers it (Wang et al., 2025).

We show the two abilities are separately learned. To test whether a single fine-tuning objective moves them together or apart, we train the same base model on two extreme signals and measure both abilities. Training on a bare correctness verdict pushes the model to one corner: its verification stays about as good as before[1], yet its generation collapses almost completely. On Qwen3.5-2B-Base[2](Qwen Team, 2026), generation accuracy falls from 63.8% to 0.0%, and the model stops attempting problems entirely. Supervised fine-tuning on solutions pushes it to the opposite corner: generation improves while verification stays weak. The two signals thus move the abilities in opposite directions. One keeps verification and destroys generation, the other keeps generation and leaves verification weak (Figure 1).

[1]Its balanced accuracy does not exceed the base, even where its F1 does; see §4.

[2]https://huggingface.co/Qwen/Qwen3.5-2B-Base

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.

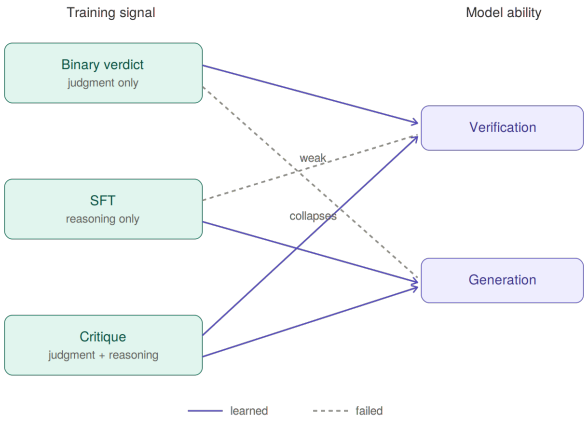


Figure 1: **Verification and generation are learned separately.** Different training signals move the two abilities in different directions: a verdict-only signal keeps verification while collapsing generation, SFT (solutions only) does the reverse, and a critique (both judgement and solutions) keeps both.

If they are separate, they are learned from *different* properties of the training signal: verification needs the judgment the signal expresses, generation needs the reasoning content it retains. A natural-language critique seems to improve both abilities at once. It is not because the critique is a uniquely powerful objective; it is simply the one signal that happens to carry *both* ingredients. A bare verdict carries only the judgment (and so keeps verification while collapsing generation); an SFT target carries only the reasoning (and so does the reverse); a critique carries both, and so preserves both. The practical question is therefore not whether to train a model to verify, but what the verification signal must contain so that doing so does not cost the model its ability to generate.

We answer this by holding the base model, the questions, and the hyperparameters fixed and varying only the target to learn along an axis; from a bare verdict, to per-step critiques, to free-form critiques, and measuring both abilities across two model families (Qwen3.5 (Qwen Team, 2026), Gemma4 (Gemma Team, 2026)) and two sizes, respectively. We find that whether verification training preserves generation is governed by the reasoning content of the signal, not its quantity: a verdict-only signal collapses generation no matter how much data it is given, whereas an otherwise identical signal that retains step-level reasoning preserves generation. This content effect is what governs whether a model trained to verify remains able to generate; we analyze how the collapse unfolds and why it is not simply a loss of skill in §5.

Our contributions are:

- Evidence that verification and generation are **separable capabilities**: a content-free signal collapses generation while sparing verification, and SFT does the reverse, though it degrades verification only weakly rather than collapsing it.
- An account of **why a critique preserves both**: the two abilities are learned from different properties of the signal, and a critique is simply the signal that carries both.
- A **controlled characterization** of when verification training preserves generation together with the failure mechanism and its causal test.
- A **practical takeaway**: a verifier cannot be assumed to remain a capable generator; training one that still solves requires keeping reasoning content in the verification signal, and a bare verdict is the most destructive.

We leave open why the ability to be trained as a verifier tracks the model family rather than its domain prior—a hypothesis we test and reject three ways (§6)—and the reverse direction, when training to generate also confers verification.

2 Related Work

2.1 Learning to critique and verify

A growing body of work trains models to evaluate rather than only produce solutions. Critique Fine-Tuning (CFT) shows that training a model to critique noisy responses, rather than imitate correct ones, improves problem-solving across mathematical reasoning benchmarks (Wang et al., 2025). Reinforcement-learning approaches train critics from outcome signals: Critique-RL finds that optimizing only for the downstream usefulness of feedback leaves the critic's ability to discriminate correct from incorrect solutions poor unless discriminability is rewarded directly (Xi et al., 2025). Others combine test-time and training-time critique supervision (Xi et al., 2024) and add natural-language and numerical feedback to policy optimization (Zhang et al., 2025). These methods report that critique training also helps generation, but establish it on a single strong-base family and a single (free-form) critique format; we vary both and find the benefit depends on the critique's reasoning content and on a competent base (§4).

2.2 Catastrophic forgetting under fine-tuning

That fine-tuning on a narrow objective degrades previously held abilities is well documented as catastrophic forgetting (Luo et al., 2023). Most relevant, an empirical study of continual fine-tuning finds forgetting more severe in *larger* models within the 1–7B range—attributed to their higher initial competence—and in models without general instruction tuning (Luo et al., 2023); both align with our observations. Comparative and mitigation studies extend this via replay, regularization, and parameter-efficient variants (Haque, 2025; Fawi, 2024; Ren et al., 2024). We differ in what we isolate rather than how we mitigate: where these track the joint degradation of prior abilities, we show a *dissociation*—one ability destroyed while a related one is retained—governed by the signal's reasoning content, and use instruction-mixing only as a causal probe of the collapse, not as a proposed remedy.

2.3 What supervised signals transfer

Richer or larger supervision does not reliably transfer capability. Scaling supervised data improves surface form more than underlying reasoning (Gudibande et al., 2023), and gains from more self-generated data depend on quality-based selection rather than volume alone (Costello et al., 2025). Our finding is consistent with this: fine-tuning on short, format-irregular web answers collapses accuracy, whereas regenerating the same answers as clean chain-of-thought restores it, locating the failure in the form of the target rather than its quantity. We extend this from generation quality alone to the joint behavior of generation and verification, identifying reasoning content as the factor that decides whether a verification signal preserves generation.

3 Experimental Setup

We isolate what property of a verification training signal determines the resulting model's generation and verification abilities. We hold everything fixed except the *content of the training target*: base model, underlying questions and candidate solutions, number of epochs, and identical optimization settings.

3.1 Verification Signal

All training data are derived from a single source, `TIGER-Lab/WebInstruct-CFT`(Wang et al., 2025)[3], using the `4k` and `50k` splits. Each example is a *(question, candidate solution, correctness label)* triplet; the candidate solutions are noisy ($\approx 27\%$ correct, 73% incorrect). From these identical triplets we construct training targets (illustrated in Appendix B) that differ only in how much reasoning content they carry:

- **Binary verdict.** Only the conclusion (i.e. `Conclusion: right`/`wrong`), a few tokens, no explanation. Zero reasoning content.

[3] `https://huggingface.co/datasets/TIGER-Lab/WebInstruct-CFT`

- **Structured critique.** A verdict for each solution step in a fixed per-step template. Reasoning content present, one rigid format.
- **Varied critique.** The same per-step judgment, surface format drawn from several styles so no single template dominates. Content present, format diversified.
- **Free-form critique.** The original unconstrained CFT critique (Wang et al., 2025), ending in a verdict. Content present, format unconstrained.

Binary, structured, and varied are all derived from the *same* free-form critiques over the *same* problems; binary by stripping the explanation to its verdict, structured and varied by reformatting it.

Reference conditions. Supervised Fine Tuning (SFT) trains on a correct solution to each question—a pure generation target with no verification signal. Because the original web answers are low quality (16% correct, 72% well-formatted by a GPT-5.4 judge over 200 samples), we regenerate them as clean chain-of-thought with GPT-5.4-mini (70% correct, 95% well-formatted); we call this **SFT-gpt** and use it as the imitation reference, reporting the raw-answer version to show its collapse is a data-quality artifact (§4). The **untrained base** is evaluated zero-shot as the reference for all deltas.

Instruction-mixing intervention. To test whether the collapse is conditioned on the instruction distribution, we train **mix** variants that replace 30% of critique examples with direct-solve instructions on the same questions.

3.2 Data scale

To separate content from quantity, each format is trained at 4K and 50K from the same pool, letting us ask whether a format that harms generation can be saved by more data (§4).

3.3 Models

We use base rather than instruction-tuned checkpoints because critique training on an already instruction-tuned model would conflate our intervention with disruption of the prior tuning. We use pre-trained (base) checkpoints from two families. Our primary analysis is on **Qwen3.5-2B and 9B**, whose bases are competent in the target domain and, therefore, permit an equal-footing comparison of the two abilities. We additionally include **Gemma-4 (E4B, 12B, 31B)** as a *boundary case*: these bases are far weaker in the domain (e.g. E4B solves 6.4% of math-500 vs. Qwen-2B's 63.8%), which lets us ask how the picture changes when the base has little prior to preserve, but makes them unsuitable for a like-for-like comparison. All models are fully fine-tuned for three epochs with identical hyperparameters within each model; only the training target varies. Infrastructure details are in Appendix A.

3.4 Evaluation

Generation. Six reasoning benchmarks (math-500 (Hendrycks et al., 2021), GSM8K (Cobbe et al., 2021), AIME24 (Hugging Face H4, 2024), Minerva-Math (Lewkowycz et al., 2022), MMLU-Pro (Wang et al., 2024), BBH (Suzgun et al., 2023)) under greedy decoding. We report per-benchmark accuracy and summarize with the unweighted mean delta versus the model's own untrained base (*avg solving* Δ). Because base (non-instruction-tuned) models can fail to terminate cleanly, we track answer-extraction/unparsed rates and inspect raw outputs; MMLU-Pro is particularly sensitive and its deltas are read as upper bounds (Appendix D).

Verification. Given a (question, candidate solution) pair, the model judges correctness. The natural label distribution is skewed (84.8% correct), so raw accuracy rewards a constant "correct" classifier. We evaluate on a class-balanced subset (chance $= 50\%$) and report balanced accuracy as the primary metric, because it is insensitive to shifts in prediction frequency. We report incorrect-class F1 alongside it but note that F1 rises with the rate at which a model predicts "incorrect" and can overstate a signal that merely predicts the minority class more often; we accordingly inspect the confusion matrix (TPR/TNR) for every model and flag degenerate near-constant predictors rather

Table 1: **Per-benchmark results across model size** (Qwen3.5-2B-Base and 9B-Base, 50K). Benchmark columns are absolute accuracy (%); "Avg Δ" is the mean generation delta vs. that model's base over the six benchmarks. Verification is incorrect-class F1 (full / balanced) and balanced accuracy; absolute verification values carry a difficulty confound (Appendix E).

		Generation (accuracy %)							**Verification**		
Model	Variant	math-500	AIME24	GSM8K	Minerva	MMLU-Pro	BBH	Avg Δ	$F1_{full}$	$F1_{bal}$	bal. acc
2B	base	**63.8**	0.0	**80.2**	**34.2**	18.1	51.1	–	0.346	0.408	60.5
	SFT-gpt	59.2	3.3	77.1	33.1	31.0	37.5	-1.0	0.378	0.450	62.2
	binary	0.2	0.0	0.0	0.0	11.3	0.0	-39.3	0.289	0.546	56.9
	structured	56.0	3.3	71.9	28.3	18.9	**51.5**	-2.9	0.433	0.585	66.5
	varied	60.0	**10.0**	78.5	33.1	**35.6**	49.4	$+3.2$	**0.446**	**0.612**	**67.6**
9B	base	75.2	16.7	91.4	36.8	41.0	64.6	–	0.631	0.628	72.6
	SFT-gpt	78.0	20.0	93.2	**50.0**	**70.6**	**70.1**	$+9.4$	**0.746**	0.773	81.2
	binary	2.6	0.0	0.0	2.6	11.3	7.2	-50.3	0.596	0.640	73.0
	structured	78.4	20.0	**93.9**	47.8	68.1	68.8	$+8.5$	0.737	**0.785**	**81.8**
	varied	**79.2**	**23.3**	**93.9**	44.5	69.0	57.0	$+6.9$	0.715	0.761	80.0

than crediting their F1. Labels come from comparing each solution's final answer to ground truth, not an LLM judge, and the same prompt is used for every model. Our primary verification set is on mathematical solutions; to probe whether the effect depends on the base model's prior in the evaluated domain, we additionally construct a BBH verification set (a domain where the Gemma bases are comparatively strong; §6).

Robustness. Primary results use greedy decoding; for core experiments, we re-run at temperature 0.7 across seeds to confirm that the large effects are not decoding artifacts. Small differences between adjacent formats should be read with the residual single-seed variance in mind.

4 RESULTS

We report results on the Qwen3.5 family (2B-Base, 9B-Base), whose bases are competent in the target domain and allow for an equal-footing comparison of the two abilities; Table 1 gives the full per-benchmark breakdown of Qwen-3.5 family. We treat Gemma-4 as a boundary case, discussed in §4.2 with full numbers in Appendix H.

4.1 SEPARABILITY, AND CONTENT AS THE GOVERNING FACTOR

We summarize generation by the **average solving delta** (avg Δ): for each fine-tuned model we take its accuracy on each of the six benchmarks, subtract the untrained base model's accuracy on that benchmark, and average the six differences. A positive avg Δ means the training improved generation over the base on average; a large negative value means it degraded it. Verification is balanced accuracy on the class-balanced set.

Verification and generation are learned separately. A single fine-tuning does not learn the two abilities together: it can collapse one while leaving the other intact. Training on a content-free verdict collapses generation; avg $\Delta = -39.3$ pp on 2B and -50.3 pp on 9B, with every math-oriented benchmark falling to near zero (e.g. math-500 $63.8\% \rightarrow 0.0\%$ at 2B), while verification does not collapse with it, staying essentially at the base level (2B: 56.9 vs. 60.5; 9B: 73.0 vs. 72.6). By contrast, supervised fine-tuning shows a different profile, improving generation (avg $\Delta = -1.0$ on 2B but $+9.4$ on 9B) without the collapse—the two abilities clearly do not move as one. That one ability collapses while the other is kept on the same problems, from the same run, is what shows the two are learned separately: a single underlying skill could not be lost and kept at once

Content preserves generation. Among signals that carry reasoning, generation is preserved: on 2B, structured, varied, and free-form critiques give avg $\Delta = -2.9$, $+3.2$, and $+4.2$ respectively versus -39.3 for the verdict-only signal. The determinant of whether generation survives is whether the target contains reasoning the model can generalize from, not whether it carries a verification objective. Raising verification above the base likewise requires content: the verdict-only signal

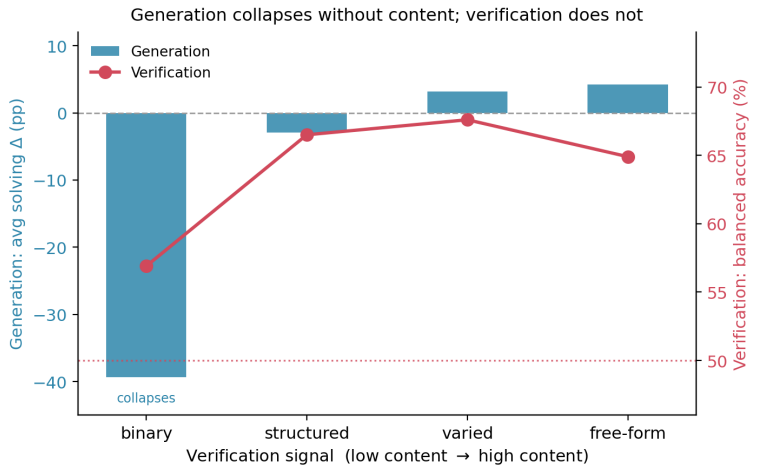


Figure 2: **Generation collapses without content; verification does not** (Qwen3.5-2B, 50K). Bars: generation (avg solving Δ vs base, left axis). Line: verification balanced accuracy (right axis) on the balanced set (random $= 50\%$). A verdict-only signal collapses generation while retaining verification; content-rich signals preserve both. Incorrect samples skew and come from fewer distinct questions, so absolute verification values carry a difficulty confound (§6); compare across formats rather than to chance.

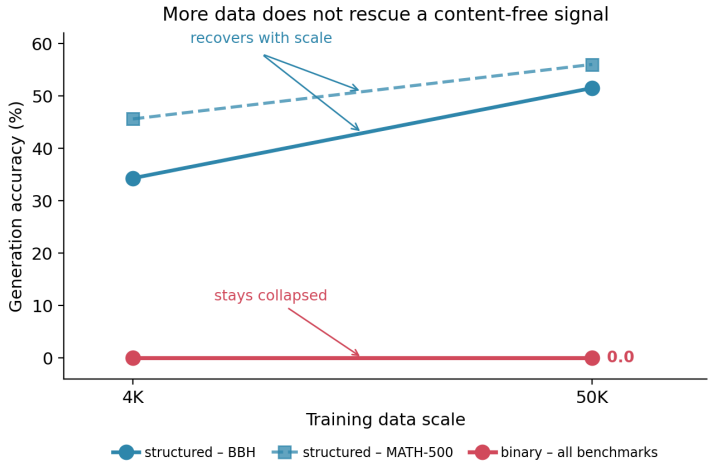


Figure 3: **More data does not rescue a content-free signal** (Qwen3.5-2B). A verdict-only signal stays fully collapsed as data grows $12\times$ (4K$\rightarrow$50K), while an otherwise identical content-bearing (structured) signal recovers. Binary is identical across all benchmarks (all 0.0); GSM8K omitted as its base accuracy is already high.

leaves verification at base (56.9 vs. 60.5 on 2B), while content-rich critiques lift it (structured 66.5, varied 67.6). Content is thus the shared ingredient behind both preserved generation and improved verification(Figure 2).

The pattern strengthens with a stronger base. Across 2B and 9B the ordering holds, and the larger base tolerates content-rich training better: structured and varied are net-positive on generation at 9B (avg $\Delta = +8.5, +6.9$) while roughly neutral at 2B. On verification the gap between critique and plain imitation narrows with scale—at 9B, SFT-gpt (81.2) is within noise of structured (81.8), whereas at 2B content-rich critiques lead imitation by several points. The advantage of critique-style training over clean imitation is therefore largest for smaller bases.

Table 2: **Generation summary across all models.** Each entry is the mean change in accuracy over six benchmarks relative to that model's untrained base. Qwen bases (competent in-domain) are our main comparison; the Gemma bases are far weaker in-domain and serve as a boundary case (§4.2). Per-benchmark accuracy and verification are in Table 1 and Appendix H.

Training target	Qwen-2B	Qwen-9B	Gemma-E4B	Gemma-12B	Gemma-31B
SFT-gpt	-1.0	$+9.4$	-1.6	-9.7	-14.9
binary	-39.3	-50.3	-15.7	-35.7	-47.5
structured	-2.9	$+8.5$	-8.0	-18.9	-22.3
varied	$+3.2$	$+6.9$	-8.7	-24.5	-26.3
free-form	$+4.2$	$+0.5$	—	—	—

Quantity does not substitute for content. The verdict-only collapse is not a data-shortage effect: increasing the verdict-only training set twelvefold leaves generation collapsed to the same degree, whereas the same scale-up substantially recovers a content-bearing (structured) signal (Figure 3). More examples of a signal with no reasoning give the model nothing further to generalize from.

4.2 The picture under a weak base prior

The Gemma-4 base models are far weaker in the domain (e.g. E4B solves 6.4% of math-500 vs. Qwen-2B's 63.8%), so we treat them not as an equal-footing comparison but as a boundary case: what happens when the base has little prior to protect. Two things change. First, the qualitative content ordering still reproduces—verdict-only is the worst, content-rich is better (Table 2)—so the direction of the effect is not specific to one family. Second, the quantitative picture degrades: absolute generation is low across the board, and the verdict-only signal does not even retain verification as it does on Qwen—on Gemma it collapses to a near-constant prediction (a degenerate solution whose F1 is an artifact), so the separability we observe on Qwen does not hold here. Severity also does not simply decrease with scale: the larger Gemma-12B is harmed *more* than E4B by the same training, consistent with prior reports that fine-tuning damage grows with a model's initial competence (Luo et al., 2023). We take Gemma to indicate that our main findings hold where the base is competent, and to mark the regime where they break down; why the base prior governs this is left open (§6).

5 Analysis

Section 4 established that content-free training collapses generation while content-rich training preserves it. Here we ask *how* the collapse happens, and use a causal intervention to identify its cause. The collapse is a failure to *engage* with the task rather than a loss of the underlying skill, and it has two components with different remedies.

5.1 Collapse is disengagement, not lost skill

We classify each failed math-500 output by whether the model engaged with the problem—a genuine attempt (right or wrong)—or did not (critique-mode: judging an absent solution; repetition collapse; or echoing the question). Two facts stand out (Figure 4). First, measurement error is negligible: solutions that were correct but mis-extracted account for 0.0% of outputs across conditions, so low accuracy is not an extraction artifact. Second, the *signature* of failure differs sharply by objective. Critique-trained models fail overwhelmingly by *not engaging*: on Gemma-4-E4B, non-engagement (critique-mode + repetition + echo) accounts for 99.8% of binary outputs, 98.8% of structured, and 96.8% of varied, while genuine attempts make up at most 0.2%. The supervised fine-tuning fails the opposite way: it attempts to solve problems, but genuine wrong attempts making up the bulk of its failures (32.6% on E4B, 52.6% on 12B) and essentially no critique-mode contamination. A model that had lost the skill would attempt and fail.

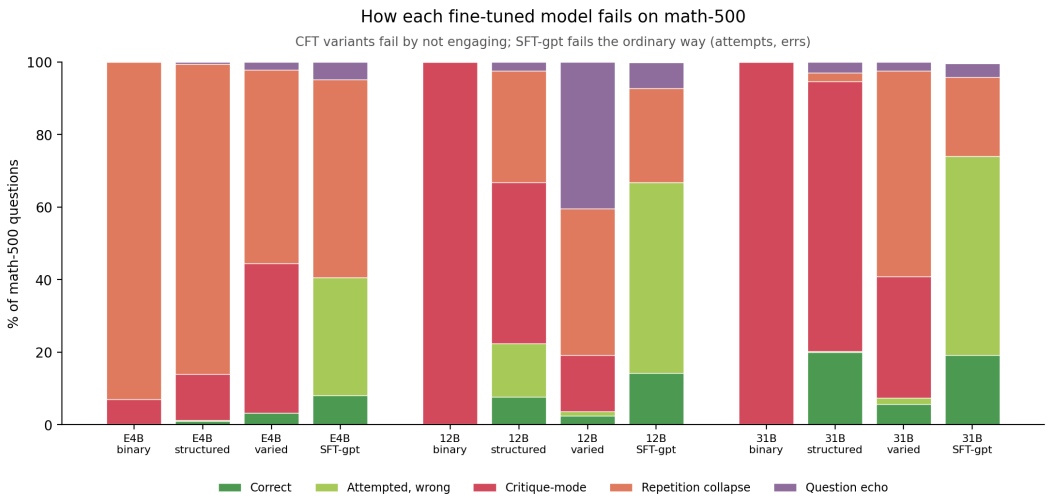


Figure 4: **Failure-mode breakdown on math-500.** Critique-trained models fail by not engaging (critique-mode, repetition, echo); the SFT control fails by attempting and erring. Categories are mutually exclusive and sum to 100% per column; "correct" is included so columns sum cleanly.

5.2 The cause: a single instruction framing

Every critique training example shares one instruction framing, so the model is never asked, during training, to respond to a bare problem by solving it. If this framing is what suppresses solving, then diversifying it should restore engagement. We test this directly by replacing 30% of critique examples with direct-solve instructions (`+mix`). Mixing drives critique-mode to near zero for every content-rich target: on Gemma-4-E4B, varied's critique-mode falls from 41.2% to 0.0% and structured's from 12.8% to 5.2%; on 12B, structured falls from 44.4% to 4.2% and varied from 15.6% to 0.0% (full per-benchmark results in Appendix I, Table 4). The disengagement is thus conditioned on the instruction distribution, not an inevitable consequence of the verification objective.

5.3 Content determines whether engagement can be restored

Restoring engagement is not the same as restoring the ability to solve. When we mix direct-solve instructions into a *content-rich* target, the model both engages and solves: E4B varied+mix reaches 18.2% correct with 75.6% genuine attempts (avg solving Δ recovering from -8.8 to $+6.5$ pp). When we do the same for the *content-free* target, the model engages but cannot solve: critique-mode disappears and genuine attempts rise to 61.4%, yet correct answers reach only 8.8% (Table 4). Because binary training never exposed the model to reasoning it could learn to produce, opening solve-mode reveals little solving ability behind it. We run this intervention on Gemma, where critique training induces the clearest disengagement; on the competent Qwen bases disengagement is mild, and the analogous test there is scale—adding twelvefold more verdict-only data likewise fails to recover generation (§4.1)—so both families point the same way. This is the sharpest evidence that reasoning content is the *source* of the preserved generation ability, not merely a trigger for the right response mode.

6 Discussion

Reconciling prior reports. Our results place seemingly conflicting prior findings on a single axis: critique training helps generation when the target carries reasoning content and the base is competent, and harms it otherwise. This subsumes both the positive reports (Wang et al., 2025) and the negative ones as different regimes rather than a contradiction. The effect also narrows with scale: on the larger Qwen-9B base, content-rich critique variants are statistically indistinguishable from the untrained base and from plain imitation on verification, echoing on a second axis what we observe for generation that the benefit of critique training is largest for smaller bases.

The verdict-only collapse is domain-general and scale-general. The degenerate behavior of verdict-only training reproduces across both domains, both families, and every scale: on BBH, Gemma-12B binary predicts "incorrect" for 100% of items, and its mixed counterpart flips to 100% "correct", the same constant-predictor signature we observe on the math verification set. Because a constant predictor is degenerate regardless of how the evaluation set is built, this conclusion does not depend on the set's composition.

Instruction mixing targets generation, not verification. Our mixing intervention removes the critique-mode disengagement that suppresses *generation* (§5); we use it only to show that this disengagement is conditioned on the instruction distribution. It is not a verification remedy—on BBH it does not improve verification (Appendix G)—so we frame it strictly as a generation-side causal probe.

Practical implications. Training a model to verify should not be assumed to leave it a capable generator. If both abilities are wanted from one model, the verification signal must carry reasoning content; a bare verdict is the worst case, collapsing generation while, at best, learning to verify degenerately. The benefit of critique-style training over plain high-quality imitation is, moreover, largest for smaller bases and narrows as the base strengthens—on our 9B bases, clean imitation matched or exceeded every critique variant.

Limitations. Our primary domain is mathematical reasoning; the BBH results add a second domain but a broad multi-domain study remains future work. Our verification sets pair correct solutions with incorrect ones produced by a strong model failing naturally, which correlates incorrectness with difficulty; we therefore restrict verification claims to those robust to this confound (the degenerate/non-degenerate distinction and relative family comparisons), and report balanced accuracy rather than F1 for the reasons detailed in Appendix E. Most runs use a single training seed, mitigated by temperature-resampling checks on the core experiment. The mechanism by which model family governs verifiability is unresolved, and repetition collapse (Appendix F) is a second failure mode our intervention does not address.

Future work. Our data suggests that generation training can confer verification ability, but with strong dependence on scale and family (e.g. SFT-gpt gains verification on Qwen but not on Gemma); characterizing this reverse transfer, and the family-level factor that governs verifiability, is left to future work.

7 Conclusion

Judging and solving are not one skill. A content-free critique tuning can keep a model's ability to verify while collapsing its ability to generate which is impossible if the two are a single underlying competence. They are separated because they are learned from different parts of the training signal: verification from the judgment it expresses, generation from the reasoning content it retains. This is why a critique appears to improve both at once while a bare verdict collapses generation. Whether verification training preserves generation is therefore decided by the reasoning content of the signal, not its quantity and not the mere presence of a verification objective. When generation collapses it does so by disengagement rather than lost skill and reasoning content is what restores a solving ability rather than merely the response mode: neither more content-free data nor a re-opened solve-mode recovers generation without it. Practically, a verifier obtained by fine-tuning should not be assumed to retain generation ability; preserving both requires that the verification signal carry reasoning content. Our study leaves open why the ability to be trained as a verifier tracks the model family rather than its domain prior and the reverse direction, when training to generate also confers verification. We see the former as the most worthwhile question to pursue next.

AI Use Statement

In this work, we used generative AI tools for research ideation and execution and for generating synthetic datasets. The initial research direction and several experimental designs were developed with the assistance of an AI system used as a brainstorming and planning aid, and the synthetic training

data were produced using large language models as described in Section 3. We used generative AI tools to draft and to polish sections of the paper.

We have reviewed all AI-assisted work. AI-suggested research ideas and framings were checked by the authors against the literature and against our own empirical results; all AI-generated synthetic data was validated against ground-truth labels (e.g., regenerated solutions were kept only when their final answers matched the reference, and verification labels were assigned by comparison to ground truth rather than by an AI judge); and all AI-assisted text was read, edited, and verified by the authors for accuracy, including every numerical claim, table entry, and citation. We take responsibility for the final content of this work, including all text, claims, and artifacts produced with the aid of generative AI.

Reproducibility Statement

We have aimed to make our study reproducible from public resources. The full experimental setup is described in Section 3, with additional training and infrastructure details in appendix A. All training data derives from the publicly available WebInstruct-CFT dataset (Wang et al., 2025) using its `4k` and `50k` splits; the binary, structured, and varied critique targets are constructed from the same source by the procedures described in Section 3 and illustrated with examples in Appendix B, and the SFT-gpt regeneration procedure is described in Section 3. All models we fine-tune are publicly released base checkpoints (Qwen3.5 (Qwen Team, 2026) and Gemma-4 (Gemma Team, 2026)). Our evaluation protocol is given in Section 3 and Appendix E, and the complete per-benchmark results underlying the summary tables are reported in Appendix H. We also note the known limitations of our verification set (a difficulty confound; Appendix E) so that our verification claims can be interpreted and reproduced appropriately.

References

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. *arXiv preprint arXiv:2110.14168*, 2021.

Caia Costello, Simon Guo, Anna Goldie, and Azalia Mirhoseini. Think, prune, train, improve: Scaling reasoning without scaling models. *arXiv preprint arXiv:2504.18116*, 2025.

Muhammad Fawi. Curlora: Stable llm continual fine-tuning and catastrophic forgetting mitigation. *arXiv preprint arXiv:2408.14572*, 2024.

Gemma Team. Gemma 4 technical report, 2026. URL `https://arxiv.org/abs/2607.02770`.

Arnav Gudibande, Eric Wallace, Charlie Snell, Xinyang Geng, Hao Liu, Pieter Abbeel, Sergey Levine, and Dawn Song. The false promise of imitating proprietary LLMs. *arXiv preprint arXiv:2305.15717*, 2023.

Naimul Haque. Catastrophic forgetting in llms: A comparative analysis across language tasks. *arXiv preprint arXiv:2504.01241*, 2025.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset. *arXiv preprint arXiv:2103.03874*, 2021. NeurIPS 2021 Datasets and Benchmarks.

Hugging Face H4. Aime 2024, 2024. URL `https://huggingface.co/datasets/HuggingFaceH4/aime_2024`.

Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, Yuhuai Wu, Behnam Neyshabur, Guy Gur-Ari, and Vedant Misra. Solving quantitative reasoning problems with language models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022. arXiv:2206.14858.

Yun Luo, Zhen Yang, Fandong Meng, Yafu Li, Jie Zhou, and Yue Zhang. An empirical study of catastrophic forgetting in large language models during continual fine-tuning. *arXiv preprint arXiv:2308.08747*, 2023.

Qwen Team. Qwen3.5: Towards native multimodal agents. `https://qwen.ai/blog?id=qwen3.5`, 2026. Model series release; repository: `https://github.com/QwenLM/Qwen3.5`.

Weijieying Ren, Xinlong Li, Lei Wang, Tianxiang Zhao, and Wei Qin. Analyzing and reducing catastrophic forgetting in parameter efficient tuning. *arXiv preprint arXiv:2402.18865*, 2024. ICASSP 2024.

Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc Le, Ed Chi, Denny Zhou, and Jason Wei. Challenging big-bench tasks and whether chain-of-thought can solve them. In *Findings of the Association for Computational Linguistics: ACL 2023*, 2023. arXiv:2210.09261.

Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, Tianle Li, Max Ku, Kai Wang, Alex Zhuang, Rongqi Fan, Xiang Yue, and Wenhu Chen. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark. In *Advances in Neural Information Processing Systems (NeurIPS) Datasets and Benchmarks Track*, 2024. arXiv:2406.01574.

Yubo Wang, Xiang Yue, and Wenhu Chen. Critique fine-tuning: Learning to critique is more effective than learning to imitate. *arXiv preprint arXiv:2501.17703*, 2025. COLM 2025.

Zhiheng Xi, Dingwen Yang, Jixuan Huang, Jiafu Tang, Guanyu Li, Yiwen Ding, Wei He, Boyang Hong, Shihan Dou, Wenyu Zhan, Xiao Wang, Rui Zheng, Tao Ji, Xiaowei Shi, Yitao Zhai, Rongxiang Weng, Jingang Wang, Xunliang Cai, Tao Gui, Zuxuan Wu, Qi Zhang, Xipeng Qiu, Xuanjing Huang, and Yu-Gang Jiang. Enhancing llm reasoning via critique models with test-time and training-time supervision. *arXiv preprint arXiv:2411.16579*, 2024.

Zhiheng Xi, Jixuan Huang, Xin Guo, Boyang Hong, Dingwen Yang, Xiaoran Fan, Shuo Li, Zehui Chen, Junjie Ye, Siyu Yuan, Zhengyin Du, Xuesong Yao, Yufei Xu, Jiecao Chen, Rui Zheng, Tao Gui, Qi Zhang, and Xuanjing Huang. Critique-rl: Training language models for critiquing through two-stage reinforcement learning. *arXiv preprint arXiv:2510.24320*, 2025.

Weizhe Yuan, Richard Yuanzhe Pang, Kyunghyun Cho, Sainbayar Sukhbaatar, Jing Xu, and Jason Weston. Self-rewarding language models. *arXiv preprint arXiv:2401.10020*, 2024.

Xiaoying Zhang, Hao Sun, Yipeng Zhang, Kaituo Feng, Chaochao Lu, Chao Yang, and Helen Meng. Critique-grpo: Advancing llm reasoning with natural language and numerical feedback. *arXiv preprint arXiv:2506.03106*, 2025. ICML 2026.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in Neural Information Processing Systems (NeurIPS)*, 2023. arXiv:2306.05685.

A Training and Infrastructure Details

All models are fully fine-tuned for three epochs with a fixed set of hyperparameters shared across every training target within a model, so that the only variable is the training text. Training uses the LLaMA-Factory framework. We fine-tune the following pre-trained-only base checkpoints: `Qwen3.5-2B-Base`, `Qwen3.5-9B-Base`, `Gemma-4-E4B`, `Gemma-4-12B`, and `Gemma-4-31B`.

Data. All training data derives from `TIGER-Lab/WebInstruct-CFT`, using the official `4k` and `50k` configuration splits (not custom subsampling). Each example is a *(question, candidate solution, correctness label)* triplets; candidate solutions are noisy, with roughly 27% labeled correct and 73% incorrect. The binary, structured, and varied targets are all derived from the same free-form critiques over the same problems—binary by reducing the critique to its verdict, structured and varied by reformatting it.

Supervised fine-tuning reference. The learning target trains on a correct solution to each question. Because the original web answers are low quality—16% correct and 72% well-formatted, as judged by GPT-5.4 over 200 samples—we regenerate them as clean chain-of-thought with GPT-5.4-mini (70% correct, 95% well-formatted). The raw-answer version is reported only to show that the SFT collapse is a data-quality artifact.

B Training Data Format Examples

To make the content axis concrete, we show how one (question, candidate solution) pair is turned into each training target. All four critique targets are derived from the same critique over the same problem; only the amount and format of the retained reasoning differs. Long outputs are truncated (`[...]`) for space.

Shared input (identical across all critique formats)

Question. Find the sum of the series $\sum_{n=0}^{\infty}(n+1)(n+2)x^n$.

Candidate solution. Writes $(n+1)(n+2) = n^2+3n+2$, splits into $\sum n^2x^n + 3\sum nx^n + 2\sum x^n$, substitutes the standard closed forms, and simplifies to $S = \frac{x(1+4x+x^2)}{(1-x)^3}$. *(This final simplification is wrong.)*

Only the target text below differs across conditions:

Binary verdict — zero reasoning content

```
Conclusion: wrong
```

Structured critique — a verdict on every step, fixed template

Step 1: expands $(n+1)(n+2) = n^2+3n+2$. — Correct.
Step 2: splits into three sums. — Correct.
Step 3: applies the standard power-series formulas. — Correct.
Step 4: combines and simplifies to $\frac{x(1+4x+x^2)}{(1-x)^3}$. — Incorrect; the simplification does not follow.
`Conclusion: wrong`

Varied critique — same per-step judgment, non-fixed surface form

• Correct — expanding $(n+1)(n+2)$ is valid.
• Correct — splitting the sum is valid by linearity.
• Incorrect — the closed forms are not applied consistently. [...]
• Incorrect — the final simplification does not produce the stated expression.
`Conclusion: wrong`

Free-form critique — unconstrained prose

"Let us analyze the solution step by step. Step 1 [...] is correct because expanding $(n+1)(n+2)$ gives n^2+3n+2. [...] Step 5: combining the terms over a common denominator [...] so the student's expression $\frac{x(1+4x+x^2)}{(1-x)^3}$ is incorrect. [...] Conclusion: wrong"

SFT-gpt target — no verdict; used by the -mix variants

$S = \sum(n^2 + 3n + 2)x^n = \sum n^2 x^n + 3\sum nx^n + 2\sum x^n$ [...] The answer is $\boxed{\frac{2}{(1-x)^3}}$.

SFT-gpt target. For the same question, the SFT-gpt target is a solution with no verdict, e.g. "$S = \sum(n^2 + 3n + 2)x^n = \sum n^2 x^n + 3\sum nx^n + 2\sum x^n$ [...] The answer is $\boxed{\cdot}$." The `-mix` variants replace 30% of critique examples with such direct-solve targets on the same questions.

Instruction-mixing. The `-mix` variants replace 30% of the critique examples above with a direct-solve instruction on the same questions, i.e. the imitation-style target rather than a critique.

C Failure-Mode Output Examples

The following are representative *model outputs* on math-500 after training, illustrating the failure modes classified in §5. Each is the model's response to a normal math problem.

Critique-mode (verdict-only / binary-trained model)

```
Given a bare problem, the model judges a solution that was never
    provided:
Conclusion: wrong  -- it never attempts the problem.
```

Repetition collapse (small Gemma, critique-trained)

```
Step 6: The model states the polar form as (3, 30 deg). -- Correct.
Step 7: The model states the polar form as (3, 30 deg). -- Correct.
Step 8: The model states the polar form as (3, 30 deg). -- Correct.
    [...]
```

Wrong attempt (SFT-gpt)

```
n = 2 (mod 7), so the remainder when n is divided by 7 is 2. Now
    consider the
product (n+2)(n+4)(n+6) divided by 7. Since n = 2 (mod 7), ... So the
    remainder
when the product is divided by 7 is 2.  Answer: 2.
```

D Evaluation Details

Generation. We evaluate on math-500, GSM8K, AIME24, Minerva-Math, MMLU-Pro, and BBH under greedy decoding, reporting per-benchmark accuracy and summarizing with the unweighted mean delta versus the model's own untrained base. Because non-instruction-tuned base models can fail to terminate cleanly, we track answer-extraction and unparsed rates and inspect raw outputs. MMLU-Pro is particularly sensitive to this: its multiple-choice format credits an answer if the target letter appears anywhere in the output, so a non-terminating base model can be scored generously. We therefore read MMLU-Pro deltas as upper bounds.

The verdict-only 0.2% artifact. The verdict-only model emits only `Conclusion: right/wrong` for all 500 math-500 items, and no answer is ever extracted (500/500 empty predictions). A single item is scored correct by our grader because its ground-truth answer normalizes to the empty string, matching the model's empty prediction; the true accuracy is 0.0%. The same 0.2% appears for the Gemma verdict-only runs and is likely the same artifact, though we verified it only for Qwen3.5-2B.

Verification. Given a (question, candidate solution) pair, the model judges correctness. The natural label distribution is skewed (84.8% correct), so we evaluate on a class-balanced MATH Level 4–5 subset of $5{,}954$ items ($2{,}977$ correct / $2{,}977$ incorrect, chance $= 50\%$) and report balanced accuracy as the primary metric, with incorrect-class F1 and separated TPR/TNR alongside. Labels come from comparing each solution's final answer to ground truth, not from an LLM judge. Because incorrect solutions are produced by a strong model failing naturally, they correlate with problem difficulty (correct samples are 48% Level 5, incorrect samples 67%), so absolute verification values carry a difficulty confound; we therefore rely on relative comparisons and on the degenerate/ non-degenerate distinction, which are robust to it (see §6).

E Verification Metric: Balanced Accuracy vs. F1

We report balanced accuracy on the class-balanced verification set as the primary metric and treat incorrect-class F1 as secondary. The reason is that F1 responds to the *rate* at which a model predicts the minority class, independently of whether its discrimination improves. On the balanced set we observe both failure directions this can cause. First, a signal can raise F1 while balanced accuracy is flat or lower: the Qwen-2B verdict-only model reaches F1 0.546 (above the base's 0.408) yet balanced accuracy 56.9% (below the base's 60.5%), because it predicts "incorrect" far more often ($17\% \rightarrow 45\%$) rather than discriminating better. Second, an apparently reasonable F1 can come from a near-constant predictor: the Gemma verdict-only models attain F1 values in a plausible range while predicting one class for essentially all inputs (TPR or TNR near 0). We therefore inspect the confusion matrix (TPR/TNR) for every model and flag such degenerate predictors rather than crediting their F1. Because the balanced set has equal class sizes, its balanced accuracy equals ordinary accuracy.

F Additional Failure-Mode Analysis

Instruction-mixing removes the framing-induced disengagement (critique-mode) but does not reach a second, framing-independent failure mode: repetition collapse. On Gemma-4-E4B, repetition drops sharply with mixing (binary $93.0 \rightarrow 27.8$, structured $85.4 \rightarrow 22.2$, varied $53.4 \rightarrow 3.2$); on the larger 12B it barely moves (structured $30.8 \rightarrow 32.6$, varied $40.4 \rightarrow 37.2$). This second component dominates at larger Gemma scale and explains why mixing fully recovers E4B's generation but only partially recovers 12B's. We distinguish it from the framing-induced disengagement our intervention targets and do not resolve it here.

G Testing the Base-Prior Hypothesis

We considered the hypothesis that whether a base can be trained to verify while remaining a generator is governed by its prior competence in the target domain, and tested it three ways. All three fail to support it.

(1) Within-model, across benchmarks. Comparing, within a single model, how well the base solves each benchmark against how much fine-tuning preserves it, we find no relationship: the correlation between base benchmark accuracy and retained accuracy is null and changes sign across training variants.

(2) A strong-generation domain (BBH). The Gemma bases are strong *generators* on BBH—Gemma-12B solves 72.4%, more than Qwen-9B's 64.6%—yet when trained to verify BBH solutions they never exceed chance (balanced accuracy 50–56% across all formats), and the verdict-only variants are degenerate constant predictors. A base that generates well in a domain is therefore not thereby a good verifier in that domain. (E4B solves 61.8% of BBH as a generator but its BBH verification is at chance, 50.3%.)

(3) Cross-family. Under identical evaluation, the weaker-generating Qwen bases learn to verify well (57–61% balanced accuracy at 2B, $\approx 82\%$ at 9B) while the stronger-generating Gemma-12B verifies at chance. Verifiability tracks the model family, not the domain prior.

Table 3: **Per-benchmark results, Gemma-4 (boundary case, unmixed).** Benchmark columns are absolute accuracy (%); "Avg Δ" is the mean generation delta vs. that model's base. Verification F1 for the verdict-only rows (†) reflects a degenerate near-constant predictor, not genuine judgment (Appendix E): E4B/12B binary predict essentially one class for all inputs. Gemma bases verify near chance, so verification is reported as F1 only.

		Generation (accuracy %)							Verification F1	
Model	Variant	math-500	AIME24	GSM8K	Minerva	MMLU-Pro	BBH	Avg Δ	full	balanced
E4B	base	6.4	0.0	5.8	2.6	29.0	61.8	–	0.004	0.004
	SFT-gpt	8.0	0.0	16.0	5.9	18.7	47.1	−1.6	0.038	0.044
	binary	0.2	0.0	0.0	0.0	11.3	0.0	−15.7	0.267†	0.666†
	structured	1.0	0.0	2.3	1.5	13.0	39.6	−8.0	0.351	0.669
	varied	3.2	0.0	6.4	2.2	12.1	29.2	−8.7	0.335	0.655
12B	base	35.0	0.0	69.1	21.7	27.7	72.4	–	0.031	0.042
	SFT-gpt	14.2	0.0	43.0	11.0	33.9	65.4	−9.7	0.093	0.096
	binary	0.2	0.0	0.0	0.0	11.3	0.0	−35.7	0.263†	0.667†
	structured	7.6	0.0	26.6	3.7	16.7	58.1	−18.9	0.365	0.658
	varied	2.4	0.0	5.8	2.2	16.1	52.3	−24.5	0.344	0.637

On the difficulty confound. These comparisons are relative statements under identical evaluation, so the difficulty confound in the verification sets (incorrect solutions skew toward harder problems; §3) is shared by both sides and cannot by itself explain the family gap. Likewise, the degenerate/non-degenerate distinction is robust to set composition, since a constant predictor is degenerate under any labeling. Controlling difficulty directly—pairing correct and incorrect solutions to the *same* problem, matched for length—is a cleaner design we leave to future work.

Instruction mixing does not improve verification. Mixing in direct-solve instructions removes the critique-mode disengagement that suppresses generation, but does not help verification: on BBH it is flat to slightly lower (structured $56.3 \rightarrow 53.2$, varied $53.8 \rightarrow 52.2$, within noise). The earlier impression that mixing "preserves" verification rested on math F1, which as noted in Appendix E can move for reasons unrelated to discrimination.

The role of the base prior: a hypothesis we reject. It is natural to suppose that whether a base can be trained to verify (and remain a generator) depends on its prior competence in the domain. Our data do not support this. We tested it three ways, all reported in Appendix G: a within-model comparison across benchmarks; a direct test in a domain (BBH) where the Gemma bases are strong *generators* yet still fail to verify above chance; and a cross-family comparison in which the weaker-generating Qwen bases learn to verify well while the stronger-generating Gemma-12B does not. A base that is a stronger generator in the domain is thus not a better verifier: verifiability tracks the model family, not the domain prior. The mechanism behind this family difference we leave open.

H Per-Benchmark Results

The main text reports generation as an averaged delta (avg solving Δ) and verification as balanced accuracy, for compactness. Here we give the full per-benchmark breakdown behind those summaries: absolute accuracy on all six generation benchmarks, the averaged delta, and all three verification metrics (incorrect-class F1 on the full and balanced sets, and balanced accuracy). Table 1 covers the Qwen3.5 models (2B and 9B) that form our main analysis; Table 3 covers the Gemma-4 boundary-case models.

A few entries warrant care. The verdict-only (binary) rows collapse generation to zero on essentially every benchmark; the apparent 11.3% on MMLU-Pro is the multiple-choice chance floor rather than solving ability, and the 0.2% on math-500 is a grading artifact (the model emits no extractable answer; Appendix D). For the Gemma models, the verdict-only verification F1 values (marked †) come from degenerate near-constant predictors—E4B binary predicts "incorrect" almost always, and 12B binary predicts one class for all inputs—so they reflect the label distribution, not judgment ability (Appendix E); we therefore do not report balanced accuracy for the Gemma verification columns, which is uninformative under such degeneracy.

Table 4: **Instruction-mixing (`+mix30sft`) on Gemma-4, per-benchmark.** Replacing 30% of critique examples with direct-solve instructions. Benchmark columns are absolute accuracy (%); "Avg Δ" is the mean change vs. that model's untrained base. Binary rows (†/‡) have degenerate verification (Appendix E): E4B binary-mix predicts "incorrect" almost always, 12B binary-mix predicts "correct" for all inputs (F1 = 0). Base and SFT-gpt rows repeated for reference.

		Generation (accuracy %)							**Verification F1**	
Model	Variant	math-500	AIME24	GSM8K	Minerva	MMLU-Pro	BBH	Avg Δ	full	balanced
E4B	base	6.4	0.0	5.8	2.6	29.0	61.8	—	0.004	0.004
	SFT-gpt	8.0	0.0	16.0	5.9	18.7	47.1	−1.6	0.038	0.044
	binary-mix	8.8	0.0	11.8	4.8	11.5	41.7	−4.5	0.263†	0.667†
	structured-mix	11.6	0.0	20.1	7.4	17.1	49.8	+0.1	0.343	0.671
	varied-mix	18.2	0.0	46.6	10.3	17.7	51.6	+6.5	0.324	0.666
12B	base	35.0	0.0	69.1	21.7	27.7	72.4	—	0.031	0.042
	SFT-gpt	14.2	0.0	43.0	11.0	33.9	65.4	−9.7	0.093	0.096
	binary-mix	14.4	0.0	34.5	9.9	18.4	63.9	−14.1	0.000‡	0.000‡
	structured-mix	12.6	0.0	31.3	10.7	25.3	61.1	−14.2	0.351	0.668
	varied-mix	8.8	0.0	22.4	10.7	22.2	64.5	−16.2	0.340	0.617

I INSTRUCTION-MIXING: PER-BENCHMARK RESULTS

Section 5.2 and §5.3 use the instruction-mixing intervention (`+mix30sft`, replacing 30% of critique examples with direct-solve instructions) to (i) confirm that the critique-mode disengagement is caused by the single instruction framing, and (ii) show that mixing recovers generation only where the training target already carried reasoning content. Table 4 gives the full per-benchmark numbers for the Gemma-4 models, alongside the untrained base and the SFT-gpt reference.

Two points are visible in the table. First, mixing removes the critique-mode failure for all critique variants (critique-mode percentages in the main text), and for the content-rich targets (structured, varied) this translates into recovered generation—avg Δ moving from clearly negative to near zero or positive on E4B. Second, the same intervention does *not* rescue the content-free binary target: although mixing makes the model attempt problems, its generation stays well below the content-rich variants (E4B avg $\Delta = -4.5$; 12B -14.1), because verdict-only training left no reasoning for the model to generalize from. The binary verification F1 values remain those of degenerate near-constant predictors (E4B always "incorrect"; 12B always "correct", hence F1 = 0), unchanged in kind by mixing.