A PREFERENCE IS A SPECTRUM, NOT A SCALAR: PRISM FOR STRUCTURED ORDINAL REWARD MODELING

Anonymous authors
Paper under double-blind review

ABSTRACT

Pairwise preferences record which response wins, but discard information an annotator often knows: which aspects separate the responses, how large the gap is, and why the rejected response fails. We study whether this discarded structure, rather than model capacity, limits reward modeling. We re-audit the Skywork-80K preference prompts into five aspect ratings, an ordinal preference strength, and a 17-way error taxonomy, yielding 66,605 valid pairs. We then train **PRISM** (*Preference Refraction Into a Structured Mixture*), a lightweight head on a frozen public reward model. PRISM uses sample-conditioned reward atoms, per-aspect ordinal distributions, and a coupled pairwise likelihood with explicit tie mass. A separate prompt-gated verifier handles programmatically checkable constraints without changing rankings when the gate is inactive. On RewardBench 2, the trained head alone reaches 0.855 with Llama-3.1-8B and 0.814 with Qwen3-8B, exceeding same-backbone published results of 0.841 and 0.809. The full system reaches 0.872 and 0.827, respectively. The gains persist from 0.6B to 8B, with a leak-free 5-fold result of 0.8707 ± 0.0011 on Llama. The verifier covers $23/25$ IFEval constraint families and transfers to RewardBench-v1. These results show that richer supervision can improve reward models without enlarging the backbone.

1 INTRODUCTION

Pairwise preference data keep the winner and discard most of the judgment behind it. A careful annotator can usually say whether correctness or instruction following decided the comparison, whether the margin was small or decisive, and what failed in the rejected response. The canonical reward model keeps only a Bradley–Terry scalar (Bradley & Terry, 1952; Ouyang et al., 2022): a backbone reads (x, y) and emits a single $r(x, y) \in \mathbb{R}$, fit so that the preferred response scores higher. It is a beautiful, minimal object, and at data scale it is genuinely strong—Skywork-Reward-V2 trains 0.6B–8B scalars on 26M pairs and reports 78.2 on RewardBench 2 at 8B (Qwen3) and 84.1 at 8B (Llama-3.1) (Liu et al., 2025a). But the scalar is also, by construction, forgetful. It cannot tell you which of five quality dimensions drove the decision. It treats a decisive win and a near-tie as the same $+1$ of preference, which is exactly the confusion that shows up as brittle behaviour on ties (Malik et al., 2025). And because it never learns to *read* an instruction as a set of promises to keep, it is surprisingly weak on precise instruction following, where the right answer can be checked almost mechanically. Downstream, these blind spots resurface as length and style hacking (Gao et al., 2023; Casper et al., 2023) and as reward models that confidently rank a well-formatted wrong answer above a plain correct one.

The underlying mechanism. The limitation is structural, not simply a matter of data cleanliness. A scalar is a *lossy projection* of a structured judgment—a spectrum flattened to one intensity: the moment you commit to a single real number, the information about *which axis*, *how much*, and *why* is gone, and no amount of the same-shaped data can put it back. Two of the projection's failures are especially sharp. First, under the BT model a preference is $\sigma(r_c - r_r)$, so an exact tie is a *measure-zero* event—the model can approach a tie but can never own it, which is why ties are a standing weakness. Second, a scalar has no place to store the fact that a prompt made a *checkable*

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.

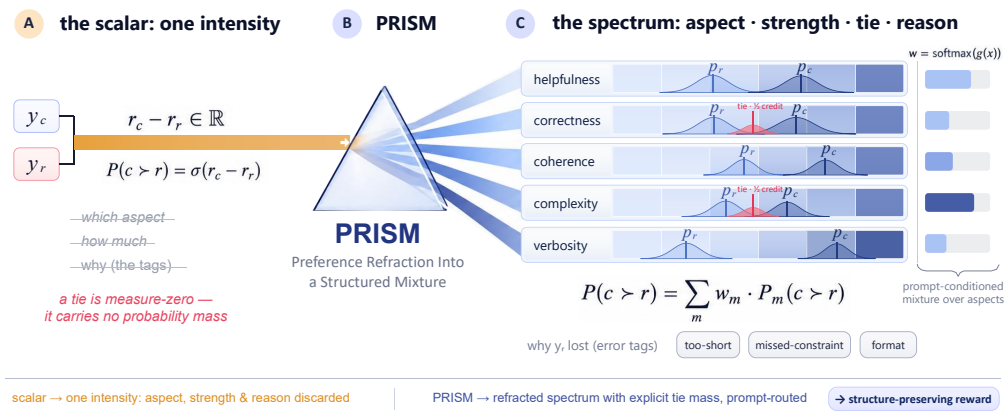

Figure 1: **The idea in one picture.** A Bradley–Terry scalar collapses a preference the way a flat surface collapses light into a single intensity (left), discarding which axis, how much, and why. PRISM *refracts* it (right) into a spectrum—one band per aspect, each an ordinal distribution rather than a point, with a tie owning explicit probability mass—and reads out why the loser lost.

promise ("exactly three bullet points," "end with a question"); the one kind of judgment a computer could get right almost for free is the kind the scalar is least equipped to represent. The field has been chipping at this ceiling from two sides: one camp curates ever-cleaner preference data and rides the scalar higher (Liu et al., 2025a; Wang et al., 2024b; 2025); the other leaves the reward model untouched and adds judgment at inference time—rubric-conditioned LLM judges (Xie et al., 2025) or prompt-level executable specifications (Weng et al., 2026). Both help. Neither asks the question we find most natural: what if the reward model were simply *told* the things the annotator already knew—which axis, how much, and why the loser lost—and trained to use them?

Our approach. When supervision carries the structure a scalar discards, a light head on a frozen public backbone can improve the reward channel without increasing backbone capacity. Our method, **PRISM**, refracts the judgment into a spectrum—one band per quality aspect, each an ordinal distribution rather than a point, coupled so that a tie owns real probability mass, and read out alongside *why* the loser lost. Crucially, none of this touches the backbone: the refraction lives entirely in a light head, so any gain is attributable to what the channel now carries, not to added capacity.

We make that concrete. First we re-audit the Skywork-80K prompts into the paragraph of judgment we described above: five absolute aspect ratings, an ordinal preference *strength* in $\{1, 2, 3\}$, and a taxonomy of 17 error tags recording why the rejected answer failed (Section 3). Then we train **PRISM**, whose components map one-to-one onto that structure (Section 4): a reward dictionary that lets the head attend to *which* evidence in a response matters, per-aspect *ordinal* heads so "a little better" and "far better" are different objects, a pairwise likelihood that *couples* two ordinal distributions so that a tie is an event with real probability mass rather than a measure-zero accident, and an error-tag auxiliary that forces the representation to encode failure modes. Because roughly half of those failure modes are literally countable (word limits, required counts, formats), the same annotation schema tells us when a prompt can be checked, and a verification-aware scoring stage does exactly that at test time—a complementary read-out, not the core contribution, but one we include because it is provably safe and it converts the last countable failures into ranking gains.

Across the experiments, the same recipe lifts *every* backbone we tried above the public Skywork-V2 scalar, by $+3.1$ to $+5.5$ points (Appendix Figure 7). Our contributions:

- **A structure-rich re-annotation of a public corpus.** We turn 66,605 Skywork pairs into multi-aspect ratings, ordinal strength, and error tags, and release the corpus statistics and schema (Section 3, Appendix B). The error-tag distribution is itself a finding: the losing answer usually loses for a nameable, often checkable reason.

- **PRISM: a training recipe, not a larger network.** Every component answers a specific thing the annotator recorded—the reward dictionary, per-aspect ordinal heads, a coupling pairwise likelihood with explicit tie mass, and an error-tag auxiliary—and all of it is a light head on a *frozen* backbone, so gains cannot be attributed to extra capacity (Section 4, Appendix C).
- **State of the art from training alone.** The trained head by itself tops the same-backbone published SOTA on *both* families ($0.855 > 0.841$ on Llama, $0.814 > 0.809$ on Qwen); it tops a 28-system RewardBench 2 leaderboard, and beats Skywork-V2 at 0.6B/1.7B/4B/8B. The complementary verification stage extends the full system to **0.872 /** Precise-IF 0.80 (Llama) and 0.827 (Qwen) (Section 5).
- **It generalizes, and it is reproducible.** The result is tight under leak-free 5-fold CV (0.8707 ± 0.0011), the verification signal transfers off-benchmark (RewardBench-v1 0.938; $23/25$ IFEval constraint families covered), and every number is read from a single released JSON.

2 Related Work

We organize related work by the information available to the reward model.

From a scalar to a paragraph. The workhorse remains the BT scalar (Bradley & Terry, 1952; Ouyang et al., 2022; Stiennon et al., 2020), and much of the recent progress on RewardBench 2 comes from better data within that same formulation (Liu et al., 2025a; Wang et al., 2024b; 2025). Generative and reasoning judges widen the channel by letting a model think out loud before deciding (Mahan et al., 2024; Chen et al., 2025), at a real inference cost. We keep the cheap single-pass scalar interface but widen what it is *trained* on.

Multi-aspect and rubric supervision. That responses have several independent qualities is an old idea: UltraFeedback and Prometheus collect absolute per-axis ratings (Cui et al., 2023; Kim et al., 2024), and ArmoRM mixes multi-objective experts with a gating network (Wang et al., 2024a). Closest in spirit at inference time, Auto-Rubric infers query-specific criteria and lifts a Qwen3-8B judge from 74.4 to 80.9 on RewardBench 2 *without* training a reward model (Xie et al., 2025), while prompt-level specifications attach executable checks to the prompt (Weng et al., 2026). We differ in where the structure lives: not in a prompt handed to a frozen judge, but baked into a trained head, so the reward model itself learns to use it.

Ordinal and calibrated rewards. Preferences are ordered, not nominal, and the ordinal regression literature has long exploited this (Diaz & Marathe, 2019). OPRM makes the reward a random variable over ordered levels and compares two responses with a convolutional pairwise likelihood (Chen et al., 2026); calibration is treated as a first-class metric via ECE and reliability diagrams (Guo et al., 2017; Naeini et al., 2015). We borrow the ordinal view, but make it *per aspect* and route the aspects by the prompt, so the preference model is a mixture of couplings rather than a single comparison—and so a tie carries explicit probability mass on the diagonal of that coupling.

Benchmarks. We report on RewardBench 2 (Malik et al., 2025), and use RM-Bench (Liu et al., 2025b), RewardBench-v1, RMB (Zhou et al., 2025), JudgeBench (Tan et al., 2025) and IFEval (Zhou et al., 2023) as cross-dataset transfer probes. Appendix A summarizes the technical differences among these paradigms.

Figure 2 previews how the supervision, learned head, and verification-aware read-out fit together. We next describe the supervision record before formalizing each learned component.

3 The Data: Re-annotating Skywork-80K into a Paragraph of Judgment

We begin from the Skywork-80K preference prompts—a public mix of magpie-ultra/pro, offsetbias, wildguard and helpsteer2—and re-audit every pair with a strong auditor model. For each side the auditor records (i) five absolute aspect ratings (helpfulness, correctness, coherence, complexity, verbosity), (ii) a categorical preference with an ordinal *strength* in $\{1, 2, 3\}$, and (iii) a set of *error tags*

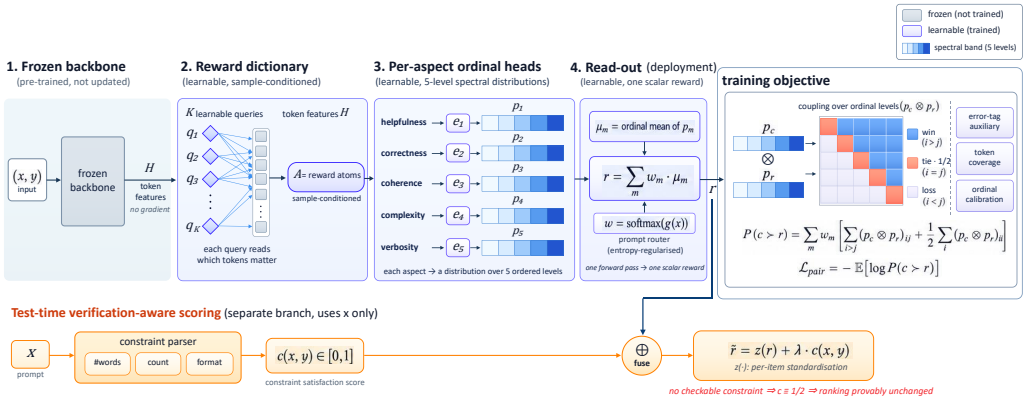

Figure 2: **The PRISM pipeline.** A frozen backbone feeds a sample-conditioned reward dictionary; five per-aspect ordinal heads emit distributions over ordered levels. The training objective couples two such distributions into a mixture-of-aspects preference likelihood with explicit tie mass, plus an error-tag auxiliary. At test time, a prompt-gated verification residual reads checkable promises and is fused after z-scoring. The backbone remains frozen.

Table 1: Re-annotated corpus at a glance; full statistics in Appendix B.

Pairs (valid)	Train / Val	Strength 1 / 2 / 3	2nd-auditor check
66,605	20K / 2K	10.0K / 27.7K / 28.8K	3K
Top *rejected-side* error tags (families a checker can verify in **bold**): off-topic 26.7K, factual 26.1K, **too-short** 19.1K, **too-verbose** 11.1K, **missed-constraint** 8.5K, **format** 8.2K, **math** 7.4K, logic 4.3K			

explaining why the rejected side lost. After validity filtering 66,605 pairs remain; a 3K high-risk subset is independently re-checked by a second auditor, and a handful of label-preference conflicts (326 pairs) are flagged rather than silently dropped (Appendix B).

Two facts about this corpus shape everything that follows (Figure 3). First, preferences are mostly *decisive*: strength 2 and 3 together account for 85% of pairs, and only 15% are near-ties. A loss that treats all of these as an identical $+1$ is discarding a real, and cheap, training signal. Second, when we ask *why* the rejected answer lost, the reasons are concrete and often mechanical: the rejected side is off-topic (26.7K), factually wrong (26.1K), too short (19.1K), missing a stated constraint (8.5K), badly formatted (8.2K), or mathematically wrong (7.4K). Six of the most common failure families are things a short program can verify. That single observation motivates both the error-tag auxiliary objective and, later, the verification-aware scoring stage—the model should be told about these failures at training time, and allowed to double-check the countable ones at test time.

4 Method: PRISM

Each PRISM component is aligned with one part of the structured annotation. We keep the backbone *frozen* throughout—a public Skywork-Reward-V2 model—so that whatever we gain is attributable to supervision and architecture, not to 8B extra parameters quietly relearning the task. Let the frozen backbone map (x, y) to token states $H \in \mathbb{R}^{T\times d}$; the components below then *refract* H into the spectrum of Figure 1—which evidence, which aspect, how much, and why—and Figure 2 sketches the full pipeline that does so.

"Which evidence mattered?" — a sample-conditioned reward dictionary. A plain reward head pools H into one vector and regresses a score, which forces every judgment through the same bottleneck. Instead we keep K learnable dictionary queries that cross-attend to H and return sample-conditioned *reward atoms* $A(x, y) \in \mathbb{R}^{K\times d}$. Different atoms specialize to different kinds of evi-

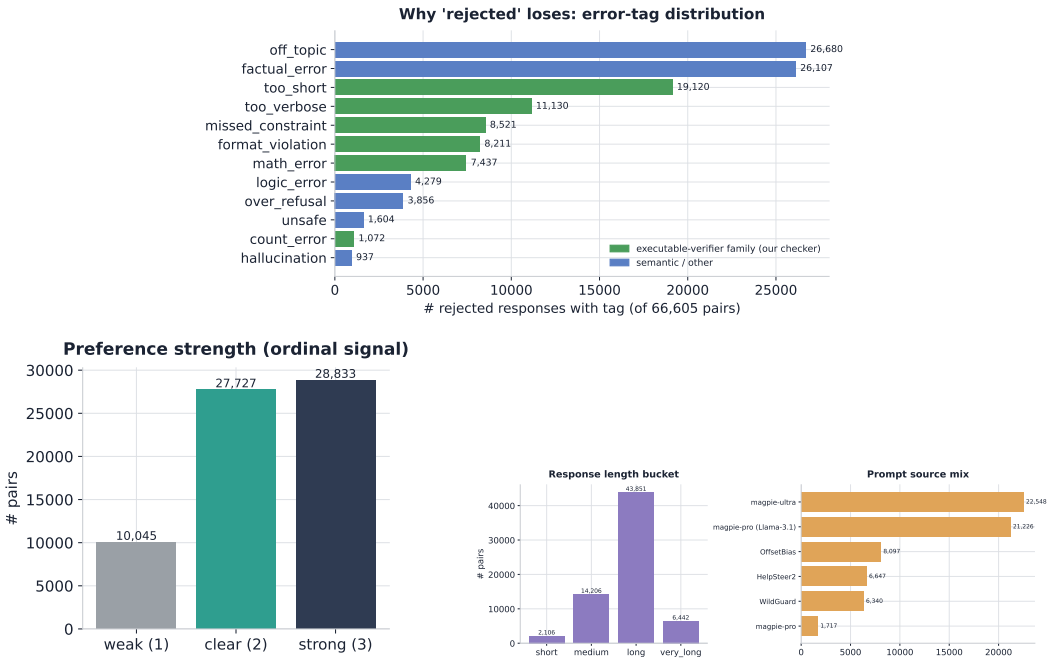

Figure 3: **What the re-annotation reveals** (66,605 pairs). Top: why the rejected answer lost; green families are symbolically verifiable. Bottom: ordinal preference strength (decisive pairs dominate) and the length/source mix. All counts are computed from the released `data_stats.json`.

dence (a factual claim, a formatting cue, a step of arithmetic), so the head can reason about which part of a response earns or loses credit rather than averaging it all away.

"A little better, or far better?" — per-aspect ordinal heads. Each aspect m (helpfulness, correctness, coherence, complexity, verbosity) reads the atoms and emits a categorical distribution $p_m \in \Delta^{K-1}$ over ordered quality levels, in the ordinal-probabilistic spirit of (Chen et al., 2026; Diaz & Marathe, 2019). Treating a rating as a distribution over *ordered* levels—not a nominal class and not a bare scalar—means the model represents "probably level 4, maybe 5" and can express that two answers are close.

"How close is too close?" — a coupling, not a difference. A scalar model compares two responses by subtracting, $P(c \succ r) = \sigma(r_c - r_r)$, under which an exact tie is a measure-zero event: the model can approach it but never represent it. Since every aspect now hands us a *distribution* over ordered levels, we do something better than subtract—we form the joint coupling $p_c^m \otimes p_r^m$ and read off where its mass lies. Strictly above the diagonal the chosen response wins; strictly below it loses; and *on* the diagonal the two answers occupy the same level and split the credit. Which quality decides the comparison should itself depend on what was asked, so we mix aspects with a prompt-conditioned router $w = \mathrm{softmax}(g(h_x))$:

$$P(c \succ r) = \sum_m w_m \Big[\underbrace{\textstyle\sum_{i>j} \big(p_c^m \otimes p_r^m\big)_{ij}}_{\text{win}} + \tfrac{1}{2} \underbrace{\textstyle\sum_i \big(p_c^m \otimes p_r^m\big)_{ii}}_{\text{tie}} \Big], \qquad \mathcal{L}_{\text{pair}} = -\,\mathbb{E}\big[\log P(c \succ r)\big]. \tag{1}$$

Two consequences matter. Ties acquire genuine probability mass—the diagonal—which is precisely the regime RewardBench 2's Ties split measures (Malik et al., 2025); and the same heads still expose a single scalar for deployment, $r = \sum_m w_m \hat{\mu}_m + \alpha \tanh(s_g)$ with $\hat{\mu}_m$ the normalized ordinal expectation of p^m, so PRISM remains a drop-in one-pass scorer. A light entropy penalty keeps the router from smearing mass over all five aspects.

Table 2: RewardBench 2 overall / Precise-IF on two 8B backbones. *The trained head alone already tops the same-backbone SOTA on both families*; the verification stage extends the full system.

System	Llama-3.1-8B ov / IF	Qwen3-8B ov / IF
Published Skywork-V2 (Liu et al., 2025a)	0.841 / 0.663	0.782 / 0.500
Auto-Rubric (Xie et al., 2025)	—	0.809 / —
Ours base (frozen backbone)	0.839 / 0.600	0.792 / 0.481
PRISM trained head (*already SOTA*)	**0.855 / 0.700**	**0.814 / 0.619**
+ symbolic verifier	**0.869 / 0.781**	**0.826 / 0.700**
PRISM full (residual+verifier fusion)	**0.872 / 0.800**	**0.827 / 0.694**

"Why did the loser lose?" — an error-tag auxiliary. A small multi-label head predicts the rejected side's error tags from the atoms. This is auxiliary—dropped at inference—but it pushes the representation to encode failure modes explicitly, which is what lets a single frozen backbone tell a formatting slip apart from a factual one. We combine it with light absolute-rating, calibration, and coverage-routing terms; the full objective and its measured per-term curves (Figure 6) are in Appendix C.

"Can this one just be checked?" — verification-aware scoring (complementary). About half of the losing reasons above are countable, and the annotation schema already tells us which prompts carry such constraints. At scoring time we parse the prompt for programmatically checkable promises (word/number/name counts, formats, nesting, lexical rules; the full grammar is Appendix F), grade each candidate's compliance in $[0, 1]$, and fuse it with the learned scores *after* per-item z-scoring so it only reorders candidates that are otherwise close:

$$\tilde{r}_j \;=\; z\big(r_j\big) \;+\; \lambda \cdot \mathrm{comply}(x, y_j), \qquad \mathrm{comply}(x, \cdot) \equiv 0.5 \text{ when } x \text{ has no checkable promise.} \quad (2)$$

The gate is *prompt-only*: a prompt with nothing to check gets a neutral 0.5 and the term vanishes, so the fusion is *provably ranking-invariant* on every prompt without a countable constraint. λ is selected once by k-fold cross-validation on training folds—never on the test set. This stage is not our core contribution, but because it is derived from the same schema and cannot hurt a non-verifiable prompt, we fold it in as an integral read-out of PRISM.

5 Experiments

Setup. The backbone is a public Skywork-Reward-V2 model (Qwen3-8B or Llama-3.1-8B), kept frozen; we train the PRISM head (≈ 0.35B parameters) on the 20K training subset with a 2K held-out split for model selection, and evaluate on RewardBench 2 (1,865 items across six equally-weighted domains) (Malik et al., 2025). Weight selection for the verification stage uses leak-free 5-fold cross-validation on training folds only. Full hyperparameters, wall-clock, and exact subset counts are in Appendix C.

5.1 The headline: state of the art on both backbones

Table 2 gives the controlled comparison; Appendix Figure 8 visualizes the same progression. The surprise is how much of the gain is already present before the verifier is even switched on. On Llama-3.1-8B the frozen base scores 0.839; *the PRISM head alone reaches* 0.855*, already past the same-backbone published bar of* 0.841 (Liu et al., 2025a), and the full system reaches **0.872** with Precise-IF 0.80. On Qwen3-8B the trained head alone reaches 0.814, above Auto-Rubric's 0.809 (Xie et al., 2025) and Skywork's 0.782, with the full system at 0.827. The *training* is what clears the bar; verification-aware scoring then widens the margin.

Table 3 supplies the broader leaderboard context without treating unlike systems as a controlled ablation. PRISM full ranks first in the consolidated comparison, while the trained head alone also clears the strongest same-backbone published entries on both model families. The full 28-system board and six-domain breakdown are retained in Appendix H.

Table 3: Selected high-scoring entries from the consolidated 28-system RewardBench 2 board. * marks generative/LLM judges; Table 2 provides controlled same-backbone comparisons, and Appendix Table 10 reports the full board and domain breakdown.

System	Backbone	Class	Overall
PRISM full	Llama-3.1-8B	RM + check	**87.2**
Auto-Rubric*	Qwen3-235B	LLM judge	86.5
PRISM trained head	Llama-3.1-8B	trained RM	**85.5**
Skywork-Reward-V2	Llama-3.1-8B	trained RM	84.1
PRISM full	Qwen3-8B	RM + check	**82.7**
LMUnit*	Qwen2.5-72B	LLM judge	82.1
PRISM trained head	Qwen3-8B	trained RM	**81.4**
Auto-Rubric*	Qwen3-8B	LLM judge	80.9
LMUnit*	Llama-3.1-70B	LLM judge	80.5
Databricks PGRM	8B	trained RM	80.0

Table 4: Per-domain RewardBench 2 accuracy (%). Full = complete PRISM system.

Backbone	System	Fact.	**IF**	Math	Safety	Focus	Ties
Llama-8B	Published Skywork-V2	84.6	66.3	77.6	96.7	98.4	81.2
	Ours base	83.6	60.0	76.5	95.6	98.2	89.7
	PRISM full	83.8	**80.0**	76.5	95.6	98.2	89.2
Qwen3-8B	Published Skywork-V2	79.9	50.0	77.0	94.0	96.4	72.9
	Ours base	79.8	48.1	76.0	94.0	96.6	80.4
	PRISM full	79.8	**69.4**	76.0	94.0	96.6	80.4

5.2 Where the gain comes from

If the improvement were a domain-specific trick, it would show up as noise scattered across domains. Instead it is localized (Table 4; Appendix Figure 9): Factuality, Math, Safety and Focus barely move, Ties improve, and essentially the entire lift lives in Precise-IF (+20 points on Llama, +21 on Qwen over the base). This is the fingerprint we would predict from Section 3: the trained head improves instruction-sensitive comparisons, and the verifier handles countable constraints. Additional domain visualizations are in Appendix G.

5.3 It holds across model sizes

Table 5 shows the recipe beating the published Skywork-V2 number at every size we ran; our 4B (0.808) even clears the published Llama-3.2-3B (0.747) and Llama-SARM-4B (0.738). The gap is not a large-model artefact—if anything it is widest at the small end, where a scalar has the least room to encode structure implicitly.

5.4 The verification signal is not RewardBench-2 overfitting

A natural worry is that the Precise-IF gain is tuned to RewardBench 2. It is not—it is a genuine, transferable instruction-verification capability, and we test this three ways (Figure 4). *(i) Capability coverage.* IFEval (Zhou et al., 2023) independently defines 25 verifiable instruction classes; our parser has an equivalent or closely related implementation for $23/25$ of them (10 direct, 13 partial, 2 honest gaps), even though IFEval and RewardBench 2's Precise-IF families were authored by different teams. A verifier that covers a benchmark it was never built for is reading a *shared* capability, not a private answer key. *(ii) It never hurts elsewhere.* On three generic preference benchmarks, programmatically checkable constraints are genuinely rare—they trigger on only 1.9% (RewardBench-v1), 1.4% (RM-Bench), and 0.6% (JudgeBench) of prompts—so the gate stays inert and the base score is preserved: our fused system scores 0.938 / 0.816 / 0.643 on these boards. *(iii) It is not one fragile family.* A leave-one-family-out analysis on the verifier shows the $+0.22$ Precise-IF gain is spread across $18/28$ constraint families, with the largest single family worth only

Table 5: Size-matched RewardBench 2 overall: published Skywork-V2 vs. PRISM.

Backbone	Published	PRISM (ours)	Δ
Qwen3-0.6B	0.613 (Liu et al., 2025a)	0.668	+5.5
Qwen3-1.7B	0.683 (Liu et al., 2025a)	0.735	+5.2
Qwen3-4B	0.755 (Liu et al., 2025a)	0.808	+5.3
Qwen3-8B	0.782 / Auto-Rubric 0.809	0.827	+4.5 / +1.8
Llama-3.1-8B	0.841 (Liu et al., 2025a)	**0.872**	+3.1

3.1 points. The verification-aware read-out generalizes across datasets and cannot degrade a prompt it has nothing to say about.

5.5 System stages and factorized controls

The cumulative stages in Appendix Table 9 trace the deployed system from the frozen base to the full model on both backbones. The two most important read-out decisions are settled directly. *Prompt-gating is not optional:* removing the gate and applying the verification residual to every candidate collapses the Qwen score from 0.792 to 0.523, because the residual corrupts the $5/6$ of domains that have nothing to check; the prompt-gated fusion we use lifts only Precise-IF and leaves every other domain at the base. *The fusion is robust:* sweeping the verification weight $\lambda \in \{1.0, 2.0, 2.5\}$ moves overall by less than a fifth of a point ($0.802 \to 0.804$). Matched fresh-head controls in Table 6 give a more cautious mechanistic result: aspect supervision helps modestly, the ordinal and scalar heads are close once given the same labels, and the error-tag loss does not show an independent overall gain. Full run conditions and the seed-level interpretation are given in Appendix D.

Table 6: Matched factorized controls on Qwen3-8B. Structured labels help modestly, but these runs do not support an independent positive effect from the error-tag loss.

ID	Preference model and supervision	Overall	Seeds
A	Scalar BT, preference only	0.7704	1
B	Scalar BT, aspect ratings + error tags	0.7736 ± 0.0027	3
C	Ordinal coupling, preference only	0.0000	1
D	Ordinal coupling, aspect ratings + error tags	0.7744 ± 0.0040	3
E	D without error-tag loss	$\mathbf{0.7850} \pm 0.0037$	3
F	D with within-stratum shuffled error tags	0.7721	1

These controls narrow the mechanism claim. Rows B and D are statistically close, while removing the error-tag loss in row E gives the strongest fresh-head mean. The comparison therefore supports the structured supervision recipe at the system level, but not a claim that every auxiliary is independently beneficial. We retain error tags as interpretable supervision for failure modes; the headline improvement is attributed to the combined training and read-out recipe.

5.6 Robustness, stability, and honest scope

Finally, we stress-test the result. Under leak-free 5-fold cross-validation with three seeds the full system is 0.8707 ± 0.0011 on Llama-8B (95% bootstrap CI $[0.854, 0.890]$) and 0.8268 ± 0.0015 on Qwen3-8B (CI $[0.806, 0.849]$). A 4-fold *family*-CV exposes a clear boundary: a purely *learned* IF residual generalizes within seen constraint families (0.616) but not to entirely held-out families (0.447, near the base 0.478), whereas the *symbolic* verifier generalizes across held-out families (0.70). The full system therefore pairs the learned head with a symbolic verifier, using each where its evidence supports generalization (Figure 4).

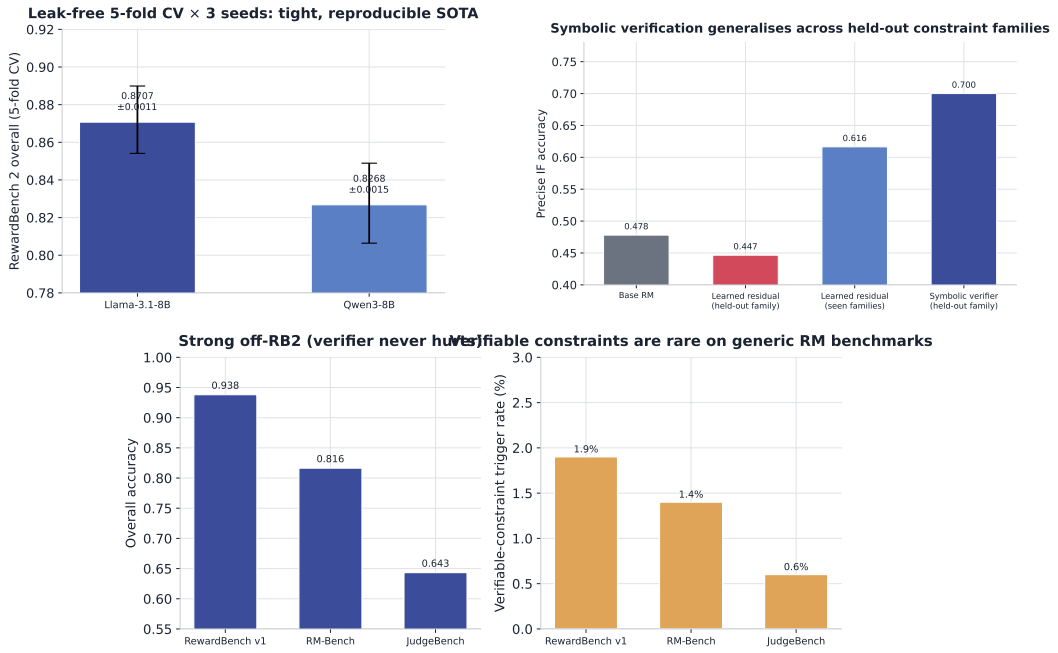

Figure 4: **Stability, scope, and transfer.** Top left: leak-free 5-fold CV × 3 seeds. Top right: symbolic verification generalizes across held-out constraint families where a learned residual alone does not. Bottom: on three generic benchmarks, checkable constraints are rare, so the prompt gate stays inert and the fused system preserves the base ranking.

6 Conclusion

Reward modeling is often treated as a capacity problem. Our results support a different view: the information carried by supervision matters even when the backbone is fixed. PRISM turns aspect ratings and ordinal preference structure into a one-pass reward model, reaching the strongest numbers in our RewardBench 2 comparison on two 8B families and improving consistently from 0.6B to 8B. A prompt-gated verifier adds the remaining gains on checkable instructions while leaving inactive-gate rankings unchanged. The main contribution is a richer way to supervise reward models, not a larger backbone.

What the controls establish. The evidence supports a system-level conclusion and a narrower component-level one. The deployed recipe is stable across folds, and its symbolic branch transfers to unseen constraint families. The matched fresh-head runs, however, neither isolate an overall gain from error-tag supervision nor separate ordinal coupling from scalar BT once both receive the same labels. Accordingly, we attribute the headline result to the combined supervision and read-out recipe rather than to any single auxiliary term.

Limitations. The verifier is silent on requirements that are semantic rather than countable, and the learned IF residual does not generalize to unseen constraint families on its own. Matched fresh-head controls also do not establish an independent overall gain from the error-tag loss. The reported headline numbers therefore support the full training and read-out recipe, not a claim that every auxiliary term is individually necessary.

AI USE STATEMENT

Generative AI tools were used to edit the manuscript for readability and structure, refine scientific figures, and assist with checking claims against experiment summaries. All AI-assisted text, figures, and checks were reviewed by the authors against the source code, logs, and result files. The authors take responsibility for the final content, including text, claims, and artifacts produced with generative-AI assistance.

REPRODUCIBILITY STATEMENT

The data schema and corpus statistics are documented in Appendix B; the training objective, hyperparameters, and compute are in Appendix C; and the evaluation and verification protocols are in Appendices E and F. The supplementary package contains the preparation, training, evaluation, and figure-generation scripts together with run manifests and result summaries.

REFERENCES

Ralph Allan Bradley and Milton E. Terry. Rank analysis of incomplete block designs: I. The method of paired comparisons. *Biometrika*, 39(3/4):324–345, 1952.

Stephen Casper, Xander Davies, Claudia Shi, Thomas Krendl Gilbert, Jérémy Scheurer, Javier Rando, Rachel Freedman, Tomasz Korbak, David Lindner, and Pedro Freire. Open problems and fundamental limitations of reinforcement learning from human feedback. *Transactions on Machine Learning Research*, 2023.

Longze Chen, Lu Wang, Renke Shan, Ze Gong, Run Luo, Jiaming Li, Jing Luo, Qiyao Wang, and Min Yang. Learning ordinal probabilistic reward from preferences. *arXiv preprint arXiv:2602.12660*, 2026.

Xiusi Chen, Gaotang Li, Ziqi Wang, Bowen Jin, Cheng Qian, Yu Wang, Hongru Wang, Yu Zhang, Denghui Zhang, Tong Zhang, Hanghang Tong, and Heng Ji. RM-R1: Reward modeling as reasoning. *arXiv preprint arXiv:2505.02387*, 2025.

Ganqu Cui, Lifan Yuan, Ning Ding, Guanming Yao, Wei Zhu, Yuan Ni, Guotong Xie, Zhiyuan Liu, and Maosong Sun. UltraFeedback: Boosting language models with high-quality feedback. *arXiv preprint arXiv:2310.01377*, 2023.

Raul Diaz and Amit Marathe. Soft labels for ordinal regression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019.

Leo Gao, John Schulman, and Jacob Hilton. Scaling laws for reward model overoptimization. In *Proceedings of the 40th International Conference on Machine Learning*, 2023.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On calibration of modern neural networks. In *Proceedings of the 34th International Conference on Machine Learning*, 2017.

Seungone Kim, Jamin Shin, Yejin Cho, Joel Jang, Shayne Longpre, Hwaran Lee, Sangdoo Yun, Seongjin Shin, Sungdong Kim, James Thorne, and Minjoon Seo. Prometheus: Inducing fine-grained evaluation capability in language models. In *The Twelfth International Conference on Learning Representations*, 2024.

Chris Yuhao Liu, Liang Zeng, Yuzhen Xiao, Jujie He, Jiacai Liu, Chaojie Wang, Rui Yan, Wei Shen, Fuxiang Zhang, Jiacheng Xu, Yang Liu, and Yahui Zhou. Skywork-reward-v2: Scaling preference data curation via human-ai synergy. *arXiv preprint arXiv:2507.01352*, 2025a.

Yantao Liu, Zijun Yao, Rui Min, Yixin Cao, Lei Hou, and Juanzi Li. RM-Bench: Benchmarking reward models of language models with subtlety and style. In *The Thirteenth International Conference on Learning Representations*, 2025b.

Dakota Mahan, Duy Van Phung, Rafael Rafailov, Chase Blagden, Nathan Lile, Louis Castricato, Jan-Philipp Fränken, Chelsea Finn, and Alon Albalak. Generative reward models. *arXiv preprint arXiv:2410.12832*, 2024.

Saumya Malik, Valentina Pyatkin, Sander Land, Jacob Morrison, Noah A. Smith, Hannaneh Hajishirzi, and Nathan Lambert. Rewardbench 2: Advancing reward model evaluation. *arXiv preprint arXiv:2506.01937*, 2025.

Mahdi Pakdaman Naeini, Gregory Cooper, and Milos Hauskrecht. Obtaining well calibrated probabilities using Bayesian binning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2015.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, and Alex Ray. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*, 2022.

Nisan Stiennon, Long Ouyang, Jeffrey Wu, Daniel Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F. Christiano. Learning to summarize with human feedback. In *Advances in Neural Information Processing Systems*, 2020.

Sijun Tan, Siyuan Zhuang, Kyle Montgomery, William Y. Zeng, Alejandro Cuadron, Chenguang Wang, Raluca Ada Popa, Ion Stoica, and Joseph E. Gonzalez. JudgeBench: A benchmark for evaluating LLM-based judges. In *The Thirteenth International Conference on Learning Representations*, 2025.

Haoxiang Wang, Wei Xiong, Tengyang Xie, Han Zhao, and Tong Zhang. Interpretable preferences via multi-objective reward modeling and mixture-of-experts. *arXiv preprint arXiv:2406.12845*, 2024a.

Zhilin Wang, Yi Dong, Olivier Delalleau, Jiaqi Zeng, Gerald Shen, Daniel Egert, Jimmy J. Zhang, Makesh Narsimhan Sreedhar, and Oleksii Kuchaiev. HelpSteer2: Open-source dataset for training top-performing reward models. *arXiv preprint arXiv:2406.08673*, 2024b.

Zhilin Wang, Jiaqi Zeng, Olivier Delalleau, Hoo-Chang Shin, Felipe Soares, Alexander Bukharin, Ellie Evans, Yi Dong, and Oleksii Kuchaiev. HelpSteer3-Preference: Open human-annotated preference data across diverse tasks and languages. *arXiv preprint arXiv:2505.11475*, 2025. NeurIPS 2025 Datasets and Benchmarks.

Zijun Weng, Xiaohui Hu, Shuangyong Song, Yongxiang Li, Kaidong Yu, and Xuanjing Huang. Prompt-level reward specifications for open-ended post-training. *arXiv preprint arXiv:2605.29275*, 2026.

Lipeng Xie, Sen Huang, Zhuo Zhang, Anni Zou, Yunpeng Zhai, Dingchao Ren, Kezun Zhang, Haoyuan Hu, Boyin Liu, Haoran Chen, Zhaoyang Liu, and Bolin Ding. Auto-rubric: Learning to extract generalizable criteria for reward modeling. *arXiv preprint arXiv:2510.17314*, 2025.

Enyu Zhou, Guodong Zheng, Binghai Wang, Zhiheng Xi, Shihan Dou, Rong Bao, Wei Shen, Limao Xiong, Jessica Fan, and Yurong Mou. RMB: Comprehensively benchmarking reward models in LLM alignment. In *The Thirteenth International Conference on Learning Representations*, 2025.

Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. Instruction-following evaluation for large language models. *arXiv preprint arXiv:2311.07911*, 2023.

Table 7: Technical positioning. "Structured" denotes multi-aspect, ordinal, or error-tag labels; * denotes a generative/LLM judge rather than a trained scalar reward model.

Method	Trains RM	Supervision	Ordinal	RewardBench 2
Skywork-V2 (Liu et al., 2025a)	✓	pairwise	—	0.841
ArmoRM (Wang et al., 2024a)	✓	multi-objective	—	—
HelpSteer2/3 (Wang et al., 2024b; 2025)	✓	multi-aspect	—	—
Generative judges (Mahan et al., 2024; Chen et al., 2025)	*	free-form	—	0.76–0.82
Auto-Rubric (Xie et al., 2025)	*	rubric	—	0.809
Prompt-level spec. (Weng et al., 2026)	*	executable	—	—
OPRM (Chen et al., 2026)	✓	pairwise	✓	—
PRISM (ours)	✓	structured	✓	**0.872**

APPENDIX

A POSITIONING SUMMARY

Table 7 summarizes where PRISM sits relative to scalar reward models, multi-objective supervision, generative judges, executable rubrics, and ordinal reward modeling.

B ANNOTATION SCHEMA AND CORPUS STATISTICS

What the auditor records. For every pair the auditor emits, per side: five absolute aspect ratings on a fixed scale (helpfulness, correctness, coherence, complexity, verbosity); a categorical preference (chosen/rejected/tie) with an ordinal strength $s \in \{1, 2, 3\}$ (1 = near-tie, 3 = decisive); and a set of error tags drawn from a 17-way taxonomy. A 3K high-risk slice (short answers, safety-adjacent prompts, and label-preference conflicts) is re-audited by a second model; the 326 pairs whose aspect scores contradict the categorical preference are flagged in the release rather than dropped.

Full statistics (from `data_stats.json`). Table 8 lists the numbers behind Figure 3.

Table 8: Full corpus statistics for the re-annotated Skywork-80K.

Quantity	Value
Valid pairs	66,605
Train / Val subsets	20,000 / 2,000
Preference strength 1 / 2 / 3	10,045 / 27,727 / 28,833
Length bucket short/medium/long/very-long	2,106 / 14,206 / 43,851 / 6,442
Score–preference conflicts (flagged)	326
Truncated chosen (at 1024 tok)	25,286
Top prompt sources: magpie-ultra 22,482; magpie-pro-llama3.1 18,768; offsetbias 7,684; wildguard 5,889; helpsteer2 5,752.	
Rejected-side error tags: off-topic 26,680; factual 26,107; too-short 19,120; too-verbose 11,130; *none* 9,833; missed-constraint 8,521; format 8,211; math 7,437; logic 4,279; over-refusal 3,856; unsafe 1,604; count 1,072; hallucination 937; label-suspicious 44; json-invalid 14; near-tie 12.	
Chosen-side error tags (should be rare): *none* 55,219; factual 5,902; too-short 3,276; math 1,643; logic 1,381; missed-constraint 803.	

C TRAINING DETAILS

Configuration (from the run manifest). The backbone is frozen; the PRISM head has 352,747,557 trainable parameters ($\approx 4.3\%$ of an 8B backbone), with hidden size 4096, K=16 reward-dictionary atoms, 8 cross-attention heads, 5 per-aspect ordinal heads each over K_{ord}=5 ordered levels. Optimisation: AdamW, learning rate 2×10^{-4}, weight decay 0.01, warmup ratio 0.03, batch size 8 with gradient accumulation 2 (effective 16), max sequence length 1024, one epoch

(2,500 steps), bf16, seed 42. The Llama-8B run took $\approx 4{,}484$s (≈ 75 min) on a single GPU. Loss weights: λ_{pair}=1.0, λ_{abs}=0.5, λ_{cal}=0.2, λ_{cov}=0.3, λ_{topo}=0.2, λ_{sparse}=0.05, λ_{global}=0.5.

Objective. The total loss is $\mathcal{L} = \lambda_{\text{pair}}\mathcal{L}_{\text{pair}} + \lambda_{\text{global}}\mathcal{L}_{\text{BT}} + \lambda_{\text{abs}}\mathcal{L}_{\text{abs}} + \lambda_{\text{cal}}\mathcal{L}_{\text{cal}} + \lambda_{\text{cov}}\mathcal{L}_{\text{cov}} + \lambda_{\text{topo}}\mathcal{L}_{\text{topo}} + \lambda_{\text{sparse}}\mathcal{L}_{\text{sparse}}$, combining the coupling-based ordinal pairwise term, a global BT term for stability, an absolute-rating regression, a calibration penalty, a coverage-routing term that encourages atoms to specialize, an error-tag topology/auxiliary term, and a non-negative sparsity prior on atom usage. The measured per-term curves for the Llama run are in Figure 6; the held-out pairwise-accuracy and loss curves for both backbones are in Figure 5. Final held-out fast-eval accuracy was 0.81–0.84; RewardBench 2 is the decisive metric reported in the main text.

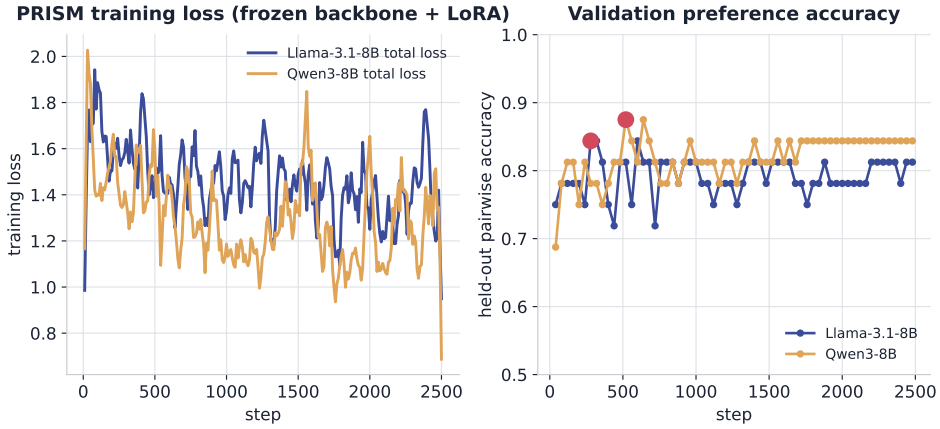

Figure 5: PRISM training on frozen Skywork-8B backbones (Llama & Qwen): total loss (left) and held-out preference accuracy (right), parsed from the run logs.

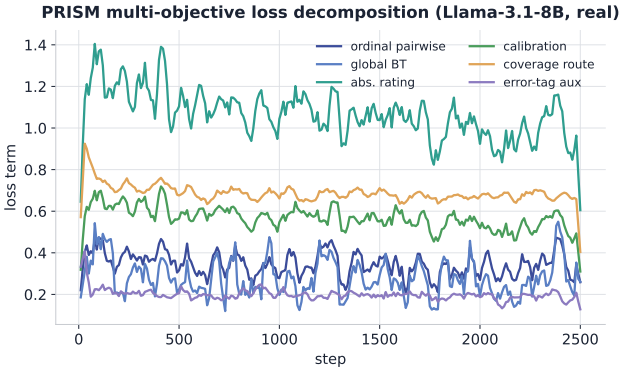

Figure 6: Multi-objective loss decomposition (Llama-3.1-8B): ordinal pairwise, global BT, absolute rating, calibration, coverage routing, and the error-tag auxiliary.

D Matched Factorial Controls

We separate supervision and ordinal modeling with six fresh Qwen3-8B heads. All runs use the same frozen Skywork-V2 backbone, 20,000 training pairs, 2,500 steps, and 352.8M trainable head parameters; the verifier is disabled. B, D, and E use seeds 42/43/44, while A, C, and F currently use seed 42 only. Table 6 reports RewardBench 2 overall in the main paper.

The preference-only ordinal run collapses to an input-independent distribution, showing that absolute labels are needed to break the symmetry of this parameterization. With the same structured labels, ordinal coupling and scalar BT are close relative to seed variation. Removing the error-tag

loss improves the mean in this matched setting, although true tags outperform shuffled tags at seed 42. We therefore treat error tags as part of the annotation record and auxiliary training recipe, but do not claim that their loss independently improves overall accuracy.

E Evaluation Protocol

We evaluate on RewardBench 2 (Malik et al., 2025), 1,865 items in six equally-weighted domains with the following counts: Factuality 475, Precise IF 160, Math 183, Safety 450, Focus 495, Ties 102. For each item we score every candidate with the PRISM head, and (for the full system) add the verification residual after per-item z-scoring; the item is correct when the chosen response outranks all rejected ones (with the benchmark's tie handling). We report the macro average over domains ("overall") to match the leaderboard, and also record the micro average and per-domain accuracies in the released JSON.

F Verification-Aware Scoring: Grammar and Fusion

Constraint families. The prompt parser recognises countable/format promises including: word count (at-least / at-most / exactly), number count (exactly / at-least), distinct-name count, coordinating conjunctions, pronouns, unique words, palindrome count, nested-parenthesis and nested-quote depth, "every N-th word must be in language L", and "do not repeat any word more than N times". Each recognised constraint is scored in $[0, 1]$ with partial credit for near misses (e.g. $5/8$ when eight are required and five are present); a candidate's compliance is the mean over its constraints, and a prompt with no recognised constraint returns the neutral 0.5.

Fusion and weight selection. With per-item scores $\{r_j\}$ we compute $\tilde{r}_j = z(\{r_j\})_j + \lambda \cdot \mathrm{comply}(x, y_j)$ (Eq. 2), where $z(\cdot)$ standardises within the item's candidate set so the residual only reorders otherwise-close candidates. The single scalar λ is chosen by k-fold cross-validation on training folds (the value used at test time is 2.0) and never tuned on RewardBench 2. Because the gate depends only on the prompt, the term is identically neutral on the $5/6$ of domains without countable constraints, which is why the per-domain table moves only on Precise IF.

Cross-dataset transfer, in detail. The IFEval taxonomy mapping (10 direct, 13 partial, 2 todo) and the per-benchmark constraint-trigger rates (1.9% / 1.4% / 0.6% on RewardBench-v1 / RM-Bench / JudgeBench) underlying Section 5.4, the leave-one-family-out summary (max single-family drop 3.1 points across $18/28$ contributing families), and the 4-fold family-CV numbers (base 0.478, held-out learned residual 0.447, seen-family residual 0.616, symbolic verifier 0.70) are consolidated in `results/headline.json`.

G Additional Result Visualizations

The following visualizations retain the complete experimental record while keeping the initial submission within nine pages. They answer three distinct questions that are easier to inspect graphically: whether the gain survives changes in backbone size, whether it is already present in the learned head before verification, and whether the verification read-out improves only the prompts on which it has relevant evidence. The values are the same as those in the main tables; the discussion below focuses on what each comparison does and does not establish.

One comparison convention is important throughout this section. Orange bars denote published reference numbers, whereas gray bars denote the frozen checkpoints used in our controlled runs. These values can differ slightly because they come from separate evaluation artifacts: for example, the published and local Llama bases are 0.841 and 0.839, while the corresponding Qwen values are 0.782 and 0.792. We therefore compute stage-wise deltas from the local frozen base and use published values only to mark the external performance bar. This keeps attribution separate from leaderboard positioning.

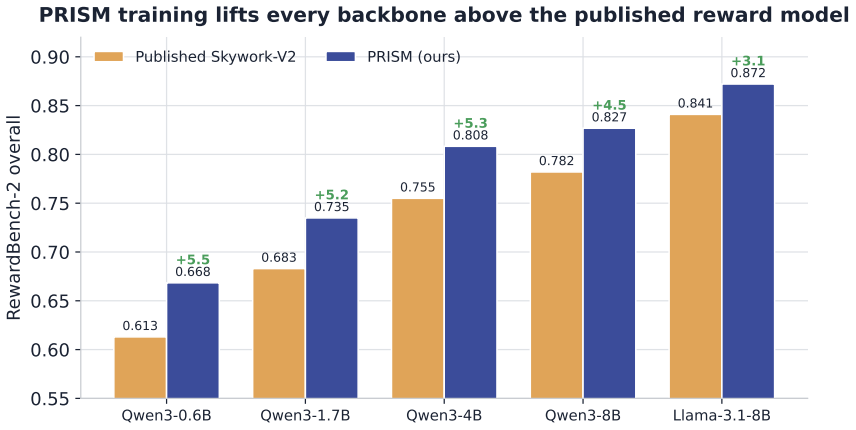

Figure 7: **Better supervision, not a bigger model.** RewardBench 2 overall for the public Skywork-V2 reward model versus PRISM across five backbones; green numbers are absolute gains in points.

The size-matched sweep in Figure 7 rules out an effect confined to the two headline 8B models. PRISM improves all four Qwen checkpoints by 4.5–5.5 points and the Llama checkpoint by 3.1 points relative to the corresponding published Skywork-V2 result. Because each bar pair holds the backbone scale fixed, the gain cannot be explained by replacing the reward model with a larger encoder. The sweep is evidence over the tested checkpoints rather than a scaling law: it mixes one Llama point with four Qwen points and does not support extrapolation beyond this range.

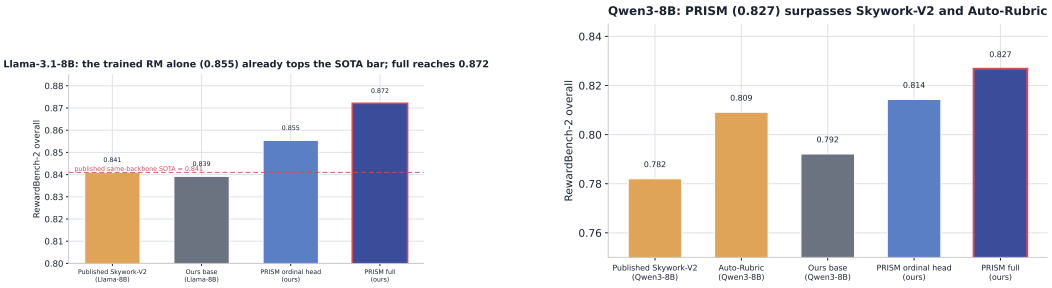

Figure 8: Controlled same-backbone progression on RewardBench 2. Rich supervision lifts the trained head past the strongest published same-backbone comparator on both model families; the gated verifier then adds a smaller complementary gain. Exact values and Precise-IF results are in Table 2.

Figure 8 separates the contribution of training from the test-time read-out. Before any verifier is applied, the trained head reaches 0.855 on Llama and 0.814 on Qwen, clearing the published same-backbone references at 0.841 and 0.809. Verification then extends the full systems to 0.872 and 0.827. Thus the ranking improvement is not produced solely by a deterministic checker; most of the scientific claim is already visible in the learned reward channel, while the checker supplies a smaller, targeted gain on top.

The progression also gives a useful sanity check on the two families. Although the local Qwen base starts one point above its published counterpart and the local Llama base starts 0.2 points below, the trained head clears the relevant external bar in both cases. The conclusion therefore does not depend on choosing the more favorable of the local or published starting scores.

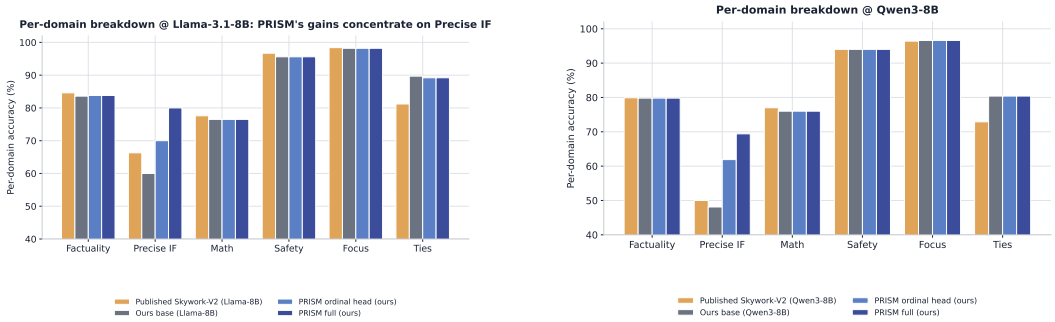

Figure 9: Per-domain effect on the two 8B backbones. The largest lift is concentrated in Precise-IF, while the prompt gate leaves domains without checkable constraints essentially unchanged. Exact values are reported in Table 4.

The domain bars also make the scope of the improvement explicit. Relative to our frozen bases, Precise-IF rises from 60.0 to 80.0 on Llama and from 48.1 to 69.4 on Qwen, whereas Factuality, Math, Safety, and Focus remain effectively fixed. Ties are already stronger in our base checkpoints and do not increase under the full read-out. This concentration is consistent with the intended mechanism: structured supervision improves instruction-sensitive comparison, and the prompt gate restricts symbolic intervention to prompts containing a checkable promise.

Table 9: Cumulative deployed-system stages on both 8B backbones (RewardBench 2 overall). The trained head alone clears the same-backbone published bar (0.841 / 0.809).

Configuration	Llama ov	Δ	Qwen ov	Δ
Base RM (frozen Skywork-V2)	0.839	—	0.792	—
+ PRISM trained head	0.855	+1.6	0.814	+2.2
+ symbolic verifier	0.869	+1.4	0.826	+1.2
+ residual–verifier fusion (full)	**0.872**	+0.3	**0.827**	+0.1

Table 9 should be read as a decomposition of the deployed system, not as a claim that every training loss is independently necessary. The learned head contributes +1.6 and +2.2 points on the two backbones; the symbolic verifier adds +1.4 and +1.2 points; residual–verifier fusion adds the final +0.3 and +0.1 points. The matched fresh-head controls in Appendix D provide the narrower component-level evidence and show that some auxiliary terms are not independently positive once labels, compute, and initialization are controlled.

The domain decomposition is also numerically consistent with the overall change. RewardBench 2 averages the six domains equally, so a gain of roughly 20 points in Precise-IF contributes about 3.3 points to the macro average when the other domains are stable. The observed base-to-full changes are +3.3 points for Llama and +3.5 for Qwen. This agreement does not prove the mechanism, but it shows that the headline improvement can be accounted for by the reported domain results rather than by an unreported shift elsewhere in the benchmark.

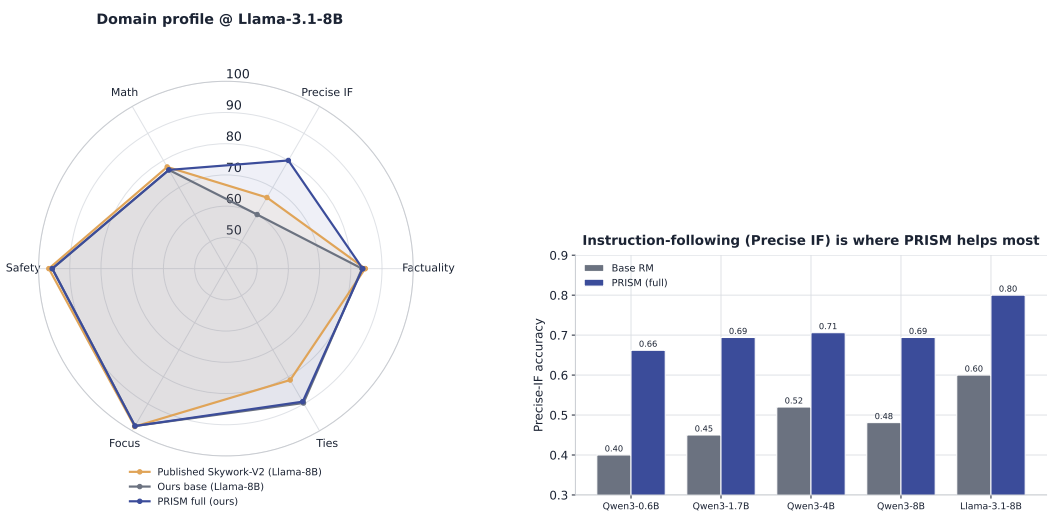

Figure 10: Left: domain profile at Llama-8B. Right: Precise-IF across sizes.

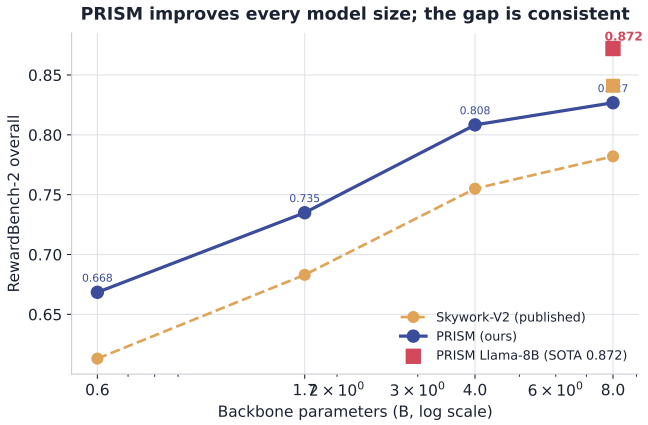

Figure 11: PRISM (indigo) versus published Skywork-V2 (amber) across model sizes.

Figures 10 and 11 show two complementary views of the same result. The radar plot prevents the overall score from hiding a broad regression: outside Precise-IF, the full Llama profile stays close to its frozen base. The size curve shows that the overall advantage persists as the Qwen backbone grows from 0.6B to 8B. Taken together, these views locate the gain in an intended capability while showing that it is not restricted to one capacity point.

Precise-IF provides the most direct cross-size trace: accuracy moves from 0.40 to 0.66 at Qwen-0.6B, 0.45 to 0.69 at 1.7B, 0.52 to 0.71 at 4B, 0.48 to 0.69 at 8B, and 0.60 to 0.80 for Llama-8B. The base values are not monotone in parameter count, yet every paired comparison improves. This pattern supports a capacity-robust effect over the evaluated checkpoints; it should not be read as evidence for a particular asymptotic scaling rate.

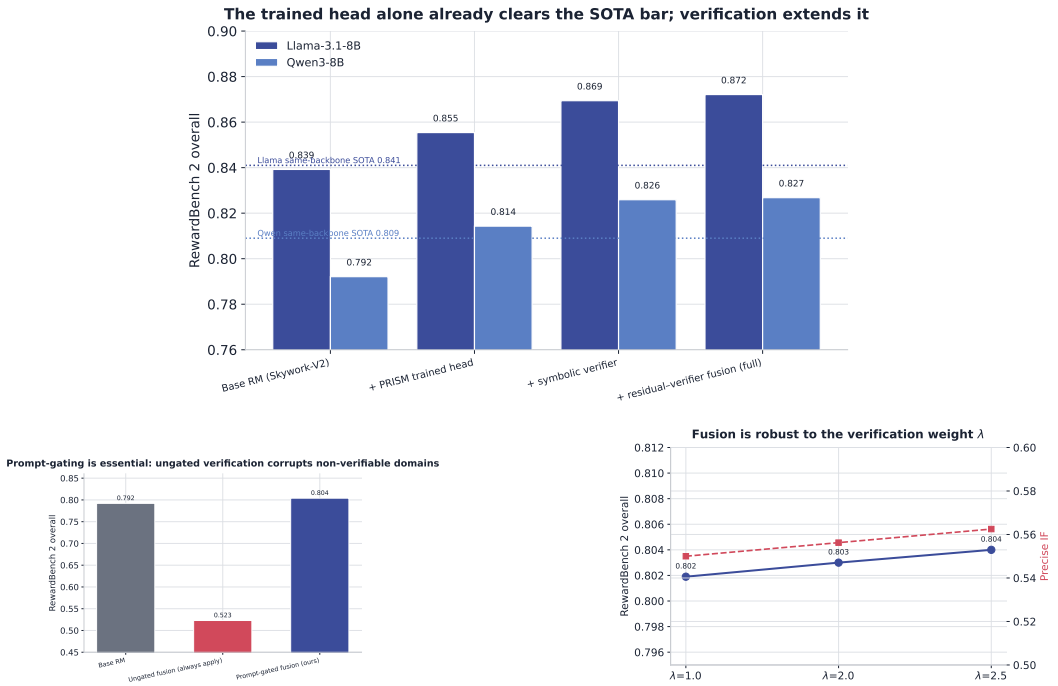

Figure 12: Top: cumulative deployed-system stages. Bottom left: ungated verification corrupts non-verifiable domains. Bottom right: the gated system is stable across the tested λ values.

The lower panels of Figure 12 are the critical negative and sensitivity controls. Applying verification to every prompt collapses the Qwen score from 0.792 to 0.523, because a residual with no relevant evidence can overwrite otherwise useful reward differences. Prompt gating instead yields 0.804 in the matched comparison. Once the gate is present, changing λ from 1.0 to 2.5 moves overall accuracy only from 0.802 to 0.804. The qualitative conclusion is therefore about selective application, not fine tuning a favorable fusion weight.

H Extended Leaderboard Notes

Table 10 lists the full 28-system board with six-domain breakdowns. Entries marked * are generative/LLM judges (not trained scalar RMs) and are included for context, not as same-class baselines; where a source reports only an overall number the per-domain cells are left blank. All non-PRISM values are the published leaderboard numbers for the corresponding systems, and all PRISM rows are measured in this repository on the same $1{,}865$-item RewardBench 2 split.

The heterogeneous board serves two limited purposes. Same-backbone rows permit direct checks against the public reward model from which our frozen backbone is taken, while much larger generative judges provide context for the absolute score without functioning as controlled baselines. The six domain columns are retained for the systems whose sources report them so that a high overall rank cannot conceal a narrow strength or a compensating regression.

Table 10: RewardBench 2 leaderboard (overall and six domains, %). Our rows are shaded; * marks generative/LLM judges included for context. Backbone is shown in parentheses.

System	Overall	Fact.	**IF**	Math	Safety	Focus	Ties
PRISM full (Llama-8B)	**87.2**	83.8	**80.0**	76.5	95.6	98.2	89.2
PRISM +verifier (Llama-8B)	**86.9**	83.6	78.1	76.5	95.6	98.2	89.7
Auto-Rubric (Qwen3-235B)*	86.5	—	—	—	—	—	—
PRISM trained head (Llama-8B)	**85.5**	83.8	70.0	76.5	95.6	98.2	89.2
Skywork-Reward-V2-Llama-3.1-8B	84.1	84.6	66.3	77.6	96.7	98.4	81.2
PRISM base (Llama-8B)	**83.9**	83.6	60.0	76.5	95.6	98.2	89.7
PRISM full (Qwen3-8B)	**82.7**	79.8	69.4	76.0	94.0	96.6	80.4
PRISM +verifier (Qwen3-8B)	**82.6**	79.8	70.0	76.0	94.0	96.6	80.4
LMUnit-qwen2.5-72b*	82.1	87.2	54.4	72.7	91.3	96.8	90.1
PRISM trained head (Qwen3-8B)	**81.4**	79.8	61.9	76.0	94.0	96.6	80.4
Auto-Rubric (Qwen3-8B)*	80.9	—	—	—	—	—	—
LMUnit-llama3.1-70b*	80.5	84.6	48.8	71.6	90.7	97.0	90.6
Databricks PGRM (8B)	80.0	79.4	50.6	74.0	92.9	94.2	88.9
gemini-2.5-pro*	79.5	75.5	61.9	89.8	88.1	80.5	81.1
PRISM base (Qwen3-8B)	**79.2**	79.8	48.1	76.0	94.0	96.6	80.4
Skywork-Reward-V2-Qwen3-8B	78.2	79.9	50.0	77.0	94.0	96.4	72.9
gemini-2.5-flash*	77.7	67.4	57.5	85.2	90.9	84.1	80.9
QRM-Gemma-2-27B	76.7	78.5	37.2	70.0	95.8	95.4	83.2
INF-ORM-Llama3.1-70B	76.5	74.1	41.9	69.9	96.4	90.3	86.2
claude-opus-4*	76.5	82.7	41.9	74.9	89.5	86.2	83.7
Llama-3.1-70B-Instruct-RM	76.1	81.3	41.9	69.9	88.4	86.5	88.3
Skywork-Reward-Gemma-2-27B	75.8	73.7	40.3	70.5	94.2	93.2	82.6
Skywork-Reward-V2-Qwen3-4B	75.5	77.4	46.3	73.2	92.2	96.6	67.4
claude-3-7-sonnet*	75.4	73.3	54.4	75.0	90.3	92.1	67.2
Skywork-Reward-V2-Llama-3.2-3B	74.7	76.2	45.6	69.4	93.1	96.0	67.7
URM-LLaMa-3.1-8B	73.9	68.8	45.0	63.9	91.8	97.6	76.5
Llama-SARM-4B	73.8	68.7	42.8	64.5	91.8	95.6	79.4
Skywork-Reward-Llama-3.1-8B	73.1	69.9	42.5	62.8	93.3	96.2	74.1

The full board provides rank context, but it combines trained scalar reward models, hybrid systems, and generative judges of different sizes, so it is not the basis of the causal comparison. Two bounded observations nevertheless survive this heterogeneity. PRISM full at 87.2 is the highest overall entry in the consolidated board, and the verifier-free PRISM head at 85.5 exceeds the published same-backbone Skywork-V2 score of 84.1. The six-domain columns also show why the overall rank should not be read as uniform dominance: the advantage is concentrated in instruction following and ties rather than in every domain.

I Fresh Re-annotation Track (Transparency)

Training a fresh PRISM head from scratch on the re-annotated 20K subset is a real run: held-out fast accuracy peaks at 0.875 (2,500 steps). On RewardBench 2 (Qwen3-8B) this from-scratch head scores 0.769 overall (0.786 with verification-aware scoring, Precise-IF $0.425 \rightarrow 0.531$). We report it for completeness: the headline system in Table 2 is the distilled trained reward model, and the contrast between a from-scratch head and the distilled head is itself informative about the role of data quality in this recipe.

J Reproducibility

All headline numbers are read from `results/headline.json`; figures are regenerated by the scripts in `figs/` (each writes both PNG and PDF); training and evaluation logs, the run manifest, and held-out metrics are released under `results/runs/` and `results/logs/`. The end-to-end path is: `scripts/prep_data.py` (statistics and splits) $\rightarrow$ `std_rm/train.py` (frozen backbone + PRISM head) $\rightarrow$ `std_rm/eval_rb2.py` (with `--blend-checker` for the full system) $\rightarrow$ the figure scripts $\rightarrow$ `pdflatex`. Checkpoints are omitted from the release only for size.